

EveryConvo

DOKUMENTAATIO

1. Johdanto

EveryConvo on uusi sosiaalisen median palvelu, jonka tarkoituksena on tarjota käyttäjille mahdollisimman vaivaton tapa pitää yhteyttä tuttaviiin ja jakaa ajatuksiaan halutulle yleisölle.

EveryConvo eroaa muista sosiaalisen median palveluista ryhmiin perustuvalla viestintäsystemillä, joka tarjoaa mahdollisimman pienen kynnyksen jakaa kertomuksiaan tarkoin rajatulle ryhmälle. Harva kehtaa laittaa esimerkiksi Facebook profiiliinsa kuvia villeimmiltä viikonlopuilta suuren kaveriryhmän nähtäville. EveryConvossa yksityisimmät tarinat voi jakaa pienen ryhmän kesken.

Vaikka EveryConvon käyttö perustuu ryhmiin, niin tarinoita voi jakaa myös koko maailman nähtäville omaan profiiliinsa. Kaikkien käyttäjien, joita käyttäjä seuraa, tarinat ja kommentit listataan yhteen paikkaan, sovelluksen päänäkökenttään.

EveryConvo tarjoaa myös yksinkertaisen pikaviestintämahdollisuuden niin ryhmien sisällä kuin käyttäjien kesken.

Sovellus toteutetaan palvelinpuolella Javalla, Tomcat toimintaympäristössä. Tietokantana toimii MySQL tietokanta, mutta tietokanta on helppo vaihtaa muutamilla muutoksilla tukemaan mitä tahansa SQL tietokantaa.

Käyttäjältä vaadittu laitteisto on mikä tahansa laite, missä on tarpeeksi moderni web-selain JavaScript tuella. Pääominaisuudet toimivat myös hieman vanhemmilla selaimilla, mutta tällöin ulkoasu saattaa kärsiä ja lieviä bugeja ilmetä.

Yleiskuvaus järjestelmästä

Järjestelmä koostuu kolmesta pääosasta: tietokannasta, sovellusrajapinnasta sekä käyttöliittymästä.

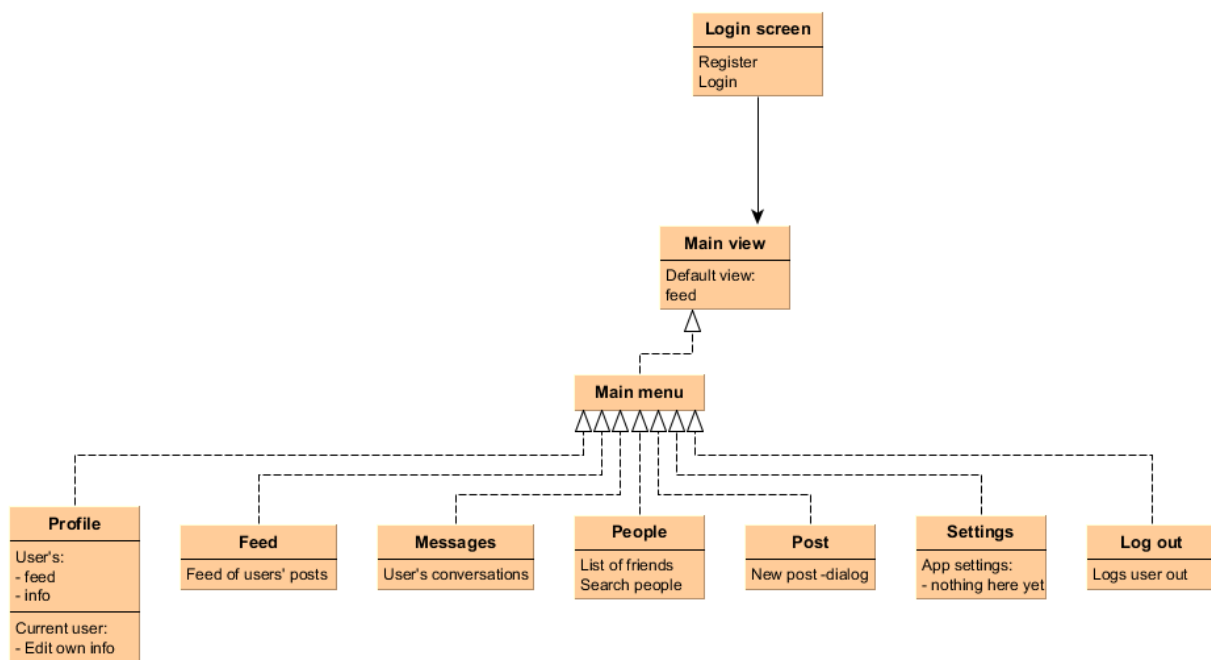
Sovellusrajapinta tarjoaa valmiiksi määriteltäviä kutsuja, jotka tekevät erilaisia toimenpiteitä tietokantaan. Itse sovellusrajapinta ei tarjoa datan käsittelyn lisäksi muuta logiikkaa.

Käyttöliittymä kutsuu rajapintaa, joka palauttaa kysytyt tiedot JSON-tietomuodossa. JSON-tiedostoista kootaan halutut näkymät käyttöliittymään. Käyttöliittymä on vahvasti JavaScript riippuvainen. Rajapintakutsut suoritetaan Ajax-kutsuilla ja näkymät luodaan dynaamisesti JavaScriptillä.

Nykyinen selainteknologia ja kehittyneet JavaScript-moottorit mahdollistavan selaimessa tapahtuvan näkymärakennuksen tehokkaasti. Tällaisella menettelyllä saadaan pidettyä sovelluksen jokaisella kolmella pääosalla tarkoin rajatut tehtävät.

2. Yleiskuva järjestelmästä

Käyttöliittymäkaavio



Oheinen kaavio on hyvin yksinkertainen kuvaus käyttöliittymästä. Päänäkymään (Main view) on sidottu oikeastaan kaikki muut näkymät. Näkymät vaihdetaan dynaamisesti päänäkömään sisällä. Oheisen kaavion tarkoitus on enemmänkin hahmottaa päävalikon elementit ja niihin kytketyt näkymät.

Käyttötapauskaavio



Käyttäjärühmät

Rekisteröimätön käyttäjä

Käyttäjä, jolla ei ole vielä tiliä sovellukseen.

Rekisteröitynyt käyttäjä

Sovelluksen normaalit roolit omaava käyttäjä. Kaikki sovellukseen rekisteröityneet käyttäjät kuuluvat tähän ryhmään

Ryhmän vastuuhenkilö

Rekisteröitynyt käyttäjä, jolla on normaaliin käyttäjien lisäksi mahdollista poistaa muiden lähettämiä tietokohteita hallinnoimastaan ryhmästä. Tämä käyttäjä voi myös päättää ketkä käyttäjät kuuluvat ryhmään.

Käyttötapauskuvaukset

Rekisteröityminen

Käyttäjä luo itselleen tunnuksen ja salasanan, sekä täyttää halutessaan lisätietokentät. Salasana tallennetaan turvallisesti hashattuna.

Käyttäjän todennus

Jotta käyttäjä pääsee käyttämään sovellusta täytyy hänen rekisteröidä itselleen tunnus. Luodun tunnuksen avulla käyttäjä voidaan todentaa kyseiseksi henkilöksi. Todennus tapahtuu syöttämällä käyttäjätunnus ja siihen liitetty salasana.

Käyttäjän poisto

Käyttäjät voivat poistaa itsensä palvelusta. Tällöin käyttäjän profiilin kaikki tiedot ja sinne jaetut tarinat tuhotaan. Lähetetyt viestit ja muualle jaetut tarinat jäävät kuitenkin palveluun.

Haku

Käyttäjät voivat haluta hakea palvelusta ryhmiä, toisia käyttäjiä, keskusteluja, tarinoita tai omia yksityisiä viestejä. Haut suoritetaan vapaamuotoisella tekstihaulla, jota voi täsmentää erilaisilla attribuuteilla (henkilöhaku, omat viestit jne).

Käyttäjien/ryhmien seuraaminen

Käyttäjät voivat haluta seurata mielenkiintoisia ryhmiä tai käyttäjiä. Mikäli ryhmä/käyttäjä on asettanut näkyvyydekseen jukisen näkyvyyden, voi kuka tahansa palvelun käyttäjä lisätä heidät seurattavien listaan. Yksityisiä ryhmiä ja käyttäjiä voi seurata vasta kun he antavat siihen luvan.

Käyttäjien/ryhmien seuraamisen lopettaminen

Käyttäjiä ja ryhmiä voi lakata seuraamasta vapaasti. Yksityisten käyttäjien ja ryhmien kohdalla joudutaan tehdä samat vaiheet kuin ensimmäisellä kerralla, mikäli ryhmää/käyttäjää halutaan alkaa seuraamaan uudestaan.

Ryhmään liittyminen

Julkisiin ryhmiin voi liittyä kuka vain. Jos ryhmä on asetettu yksityiseksi, tarvitaan ryhmään liittyäkseen hyväksytyn liittymispyynnön tai kutsun.

Ryhmästä eroaminen

Ryhmistä voi erota vapaasti. Yksityisistä ryhmästä eronnut, joka haluaa palata myöhemmin takaisin ryhmään, joutuu menetellä samalla tapaa kuin ensimmäisellä kerralla.

Ryhmän luonti

Jokainen käyttäjä voi luoda ryhmiä. Ryhmien näkyvyyden voi ryhmän luonut käyttäjä itse määritellä. Ryhmän luoja voi kutsua ja hyväksyä käyttäjiä ryhmään.

Pikaviestintä

Julkisiin ryhmiin ja julkisille käyttäjille voivat kaikki sovelluksen käyttäjät lähettää pikaviestejä. Yksityisille henkilöille voi lähettää viestejä vasta silloin, kun käyttäjä on hyväksynyt seuraamispyynnön. Yksityisiin ryhmiin voi lähettää viestejä vasta kuin käyttäjä on liittynyt ryhmään. Pikaviestit näkevät vain ryhmän jäsenet ja keskustelun osapuolet.

Tarinoiden jakaminen

Pikaviestejä pidempia kertomuksia, tarinoita, voi jakaa omaan profiiliin ja julkisiin ryhmiin. Yksityisiin ryhmiin tarinoita jaettaessa täytyy olla liittynyt ryhmään. Julkisiin profiileihin ja ryhmiin jaetut tarinat näkevät kaikki käyttäjät. Yksityisiin profiileihin ja ryhmiin jaetut tarinat näkevät vain ko. profiilia/ryhmää seuraavat käyttäjät.

Tarinoiden kommentointi

Kaikki käyttäjät voivat kommentoida niitä tarinoita, joita heillä on oikeus nähdä.

Tarinoiden poisto

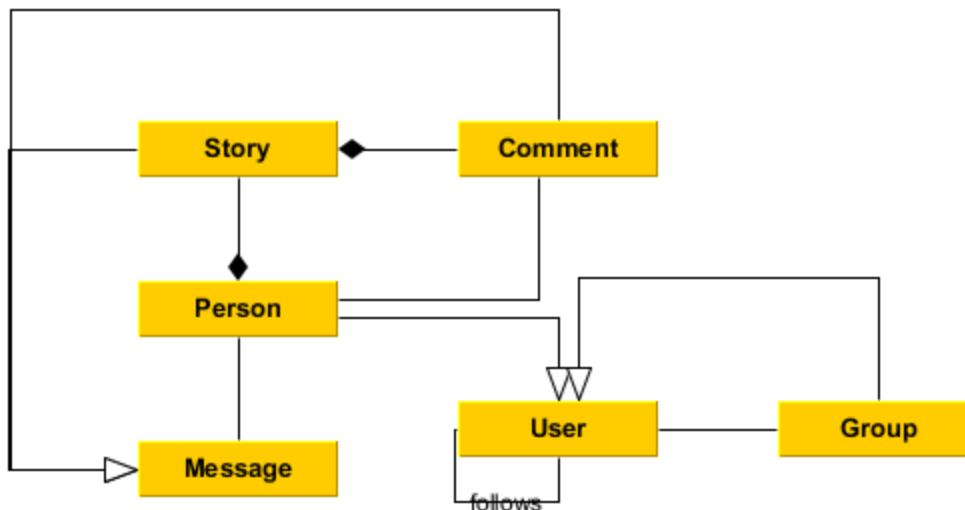
Vain omaan profiiliin ja käyttäjän hallinnoimiin ryhmiin jaetut tarinat voi käyttäjä poistaa.

Tapahtumalistaus (päänäkymä)

Jokaiselle käyttäjälle näkyy heille itselleen räätälöity tapahtumalistaus, jossa näkyy kaikki heidän seuraamien ryhmien ja käyttäjien toiminnot, joita käyttäjällä on oikeus nähdä.

3. Järjestelmän tietosisältö

Tietokohteet



User

Käyttäjä. Voi olla joko oikea henkilö tai ryhmä. Käyttäjät voivat seurata toisia käyttäjiä.

Attribuutti	Arvojoukko	Kuvailu
username	merkkijono, max. 20	
description	merkkijono, ei ylärajaa	

websiteurl	merkkijono, ei ylärajaa	Linkki käyttäjän omille www sivuille, vapaamuotoinen
location	merkkijono, max. 255	Käyttäjän sijainti vapaamuotoisena tekstinä
visibility	merkkijono, max. 1	Käyttäjän näkyvyys sovelluksessa: w: julkinen a: kaverin kaverit f: kaverit i: yksityinen
imageurl	merkkijono, ei ylärajaa	Käyttäjän profiilikuvan osoite

Person

Oikeaksi henkilöksi ilmoittautunut käyttäjä.

Attribuutti	Arvojoukko	Kuvailu
firstname	merkkijono, max. 60	
lastname	merkkijono, max. 60	

Group

Ryhmäksi ilmoittautunut käyttäjä. Ryhmään voi liittyä käyttäjiä.

Attribuutti	Arvojoukko	Kuvailu
-------------	------------	---------

Message

Pikaviesti. Voidaan liittää ryhmään tai kahteen käyttäjään.

Attribuutti	Arvojoukko	Kuvailu
content	merkkijono, ei ylärajaa	
timestamp	timestamp	Viestin lähetysajankohta

Story

Tarina. Usein pidempi kuin pikaviesti ja tarkoitettu suuremmalle yleisölle. Perii Message tietokohteen.

Attribuutti	Arvojoukko	Kuvailu
mediaurl	merkkijono, ei ylärajaa	Tarinaa liitetty medialinkki

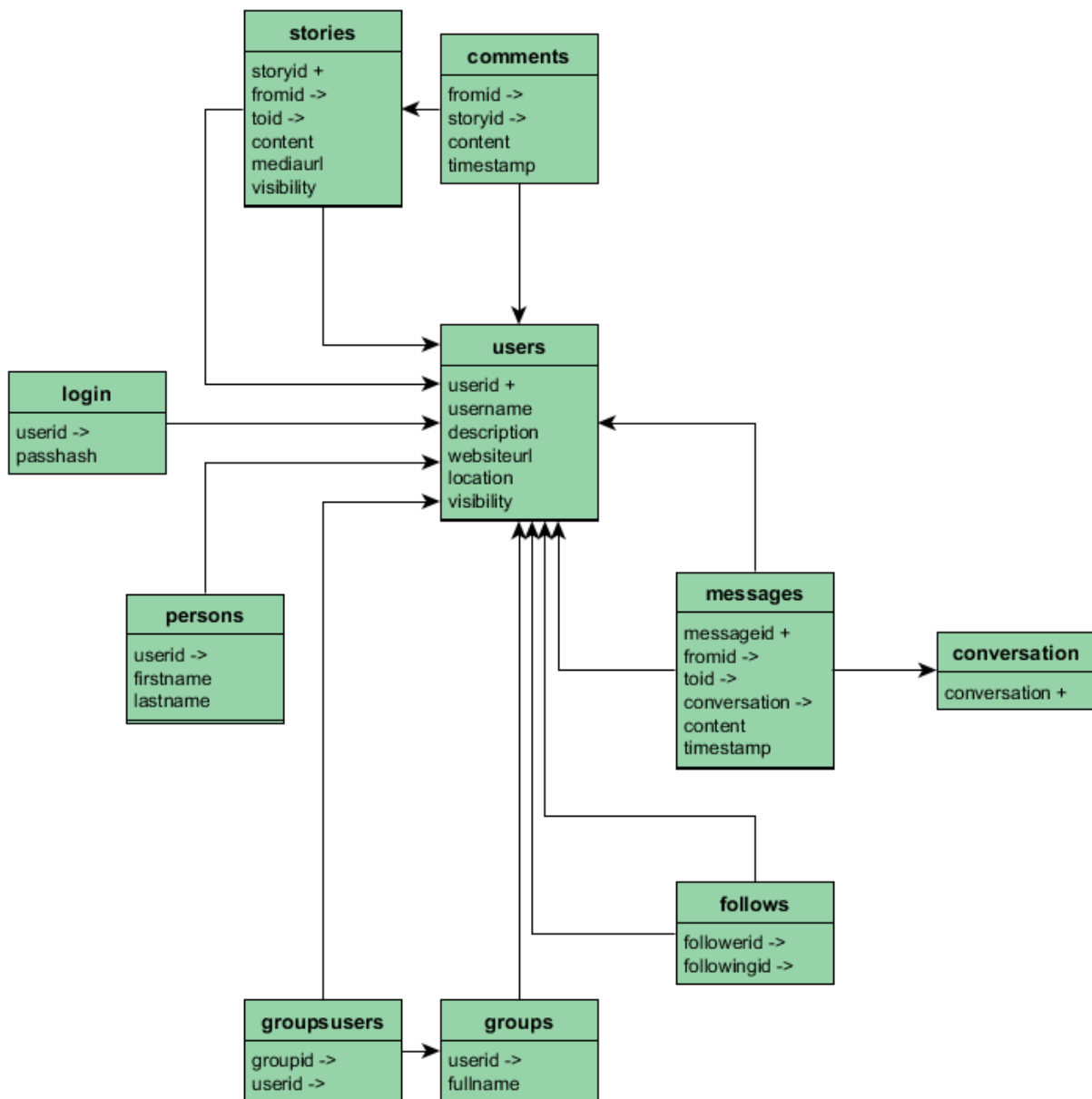
Comment

Kommentti. Voidaan liittää tarinaan. Perii Message tietokohteen

Attribuutti	Arvojoukko	Kuvailu
-------------	------------	---------

4. Relaatiotietokantakaavio

Oheisessa kaaviossa attribuuttiin liitetty "+" merkki kuvaa pääavainta (primary key) ja nuoli (->) kuvaa viiteavainta (foreign key).



5. Järjestelmän yleisrakenne

Järjestelmä koostuu kahdesta pääosasta; rajapinnan tarjoavasta sovelluksesta ja rajapintaa käyttävästä käyttöliittymäsovelluksesta. Rajapintasovellus antaa kysytyt tiedot JSON-tiedostomuodossa.

Rajapintasovellus

Lähdekooditiedostot ovat paketissa *fi.raka.everyconvo.api*. Kyseinen paketti pitää sisällään muita paketteja, joihin on kasattu sovelluksen eri osat:

- *fi.raka.everyconvo.api.entities*
 - Tietokantatietueita mallintavat luokat
- *fi.raka.everyconvo.api.json*
 - Yleiset JSON-apuluokat
- *fi.raka.everyconvo.api.servelets*
 - Serveletit
- *fi.raka.everyconvo.api.sql*
 - Yleiset SQL-apuluokat
- *fi.raka.everyconvo.api.utils*
 - Muut yleiset apuluokat

Rajapintasovellus palauttaa tiedot seuraavanlaisessa JSON-muodossa:

```
{
  "data": [
    {
      "--attribuutti--": "--arvo--"
    }
  ],
  "prev": "--edelliseen sivuun vievä linkki--",
  "next": "--seuraavaan sivuun vievä linkki--"
}
```

Lista *data* pitää sisällään haetut elementit. Mikäli elementtejä on enemmän kuin kerrallaan voidaan enimmillään näyttää, pitää JSON-olio sisällään seuraavalle sivulle vievän linkin *next*. Mikäli elementtejä on myös edellisellä sivulla, on JSON-oliossa mukana attribuutti *prev*.

Asetustiedosto löytyy polusta *src/main/webapp/WEB-INF/db_config.xml*. Asetustiedostossa määritellään tietokantaa käyttävän käyttäjän käyttäjänimi ja salasana.

Sovellus tallentaa käyttäjien lataamat tiedostot kansioon *src/main/webapp/u*.

Käyttöliittymäsovellus

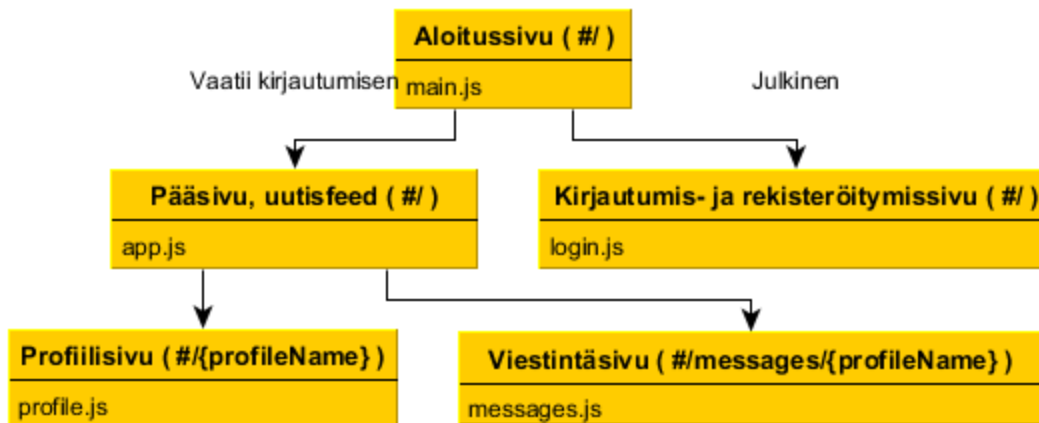
Käyttöliittymäsovellus on rakennettu HTML5 rajapintoja ja JavaScriptiä käyttäen. Sivun ulkoasu tyyleistä vastaa CSS tyylitiedostot. Käyttöliittymäsovellus käyttää *require.js* kirjastoa, mikä luo sovellukselle kyseisen hakemistopuun:

- img // Staattiset kuvatiedostot
- scripts // JavaScript tiedostot
 - \ - lib // Yleiset apuskriptit
 - | - widgets // Widgetit

Scripts-kansion juuressa on pääluokat, jotka vastaavat sovelluksen päänäkymien muodostamisesta. Koko käyttöliittymäsovelluksen juuressa on *index.html*-tiedosto ja tyylitiedostot.

Käyttöliittymä rakennetaan kokonaan dynaamisesti käyttäen JavaScriptiä. Sovellus tekee kutsuja rajapinnalle, jolta se saa JSON-tiedostomuodossa vastauksen ja piirtää näkymät saadun tiedon pohjalta.

6. Käyttöliittymä ja järjestelmän komponentit



Käyttöliittymän näkymäkaavio on hyvin simppeli. Aloitussivu hoitaa käyttäjän ohjauksen joko sovelluksen pääsivulle tai kirjautumissivulle, mikäli käyttäjä ei ole kirjautunut sisään. Pääsivu näyttää käyttäjällä uutissyötteen, jossa on koottuna muiden käyttäjien julkaisemat tarinat. Profiilisivu näyttää valitun käyttäjän tiedot ja ko. käyttäjän julkaisemat tarinat. Viestintäsivu näyttää kyseisen käyttäjän välisen keskustelun.

Kaikille sovelluksen kirjautumista edellyttävillä sivuilla pääsee päävalikosta. Päävalikosta aukeaa myös henkilö- ja ryhmälistaus. Nämä listaukset näyttävät sovelluksen käyttäjät, ja käyttäjää klikkaamalla pääsee ko. käyttäjän profiilisivulle. Päävalikosta aukeaa myös ryhmien ja tarinoiden lisäykseen tarvittava dialogi.

Profiilisivulla kukin käyttäjä voi muokata omia tietojaan.

7. Asennustiedot

Ennen sovelluksen asennusta, kopioi pakattu .war-tiedosto palvelimelle. Varmista että MySQL palvelin on toiminnassa ja palvelimellasi on tuki Java-servlettien ajoon. Jos palvelimen asetukset on kunnossa, voit aloittaa automaattisen asennuksen menevällä seuraavaan osoitteeseen:

```
palvelimesiosoite/EveryConvoAPI/install?username={mysql_käyttäjä}  
&password={mysql_salasana}
```

Jos kaikki meni niin kuin piti, selaimeen tulla viesti onnistuneesta asennuksesta. Asennustiedosto luo tietokantapalvelimelle *everyconvo* nimisen tietokannan kaikkine tauluineen. Samalla myös sovelluksen asetukset asetetaan käyttämään juuri luotua tietokantaa.

TODO: Graafinen käyttöliittymä

8. Käynnistys- / käyttöohje

Sovelluksen käytön kanssa ei pitäisi tulla ongelmia. Toiminnot ovat hyvin samankaltaiset muihinkin sosiaalisen median palveluihin verrattuna. Testausta varten voit luoda uuden käyttäjätunnuksen tai kirjautua muutamalla testitunnuksella sisään.

Testitunnukset:

viivi : viivi

pekka : pekka

10. Omat kokemukset

Tietokantasovelluksia olen tehnyt jo aiemmin ja niitä lähinnä työnkin puolesta tulee tehtyä. Nyt kuitenkin valitsin Javan PHP:n sijaan, mikä osoittautui mukavaksi valinnaksi.

Hankalinta sovelluksen teossa oli saada relaatiot toimimaan keskenään. Myös pientä päänvaivaa aiheutti koodin optimointi ja turhan kooditoiston karsiminen.

11. Muu dokumentaatio