

Ystävänvälityspalvelu

Tietokantasovelluksen harjoitustyö, Periodi II

Arttu Siren

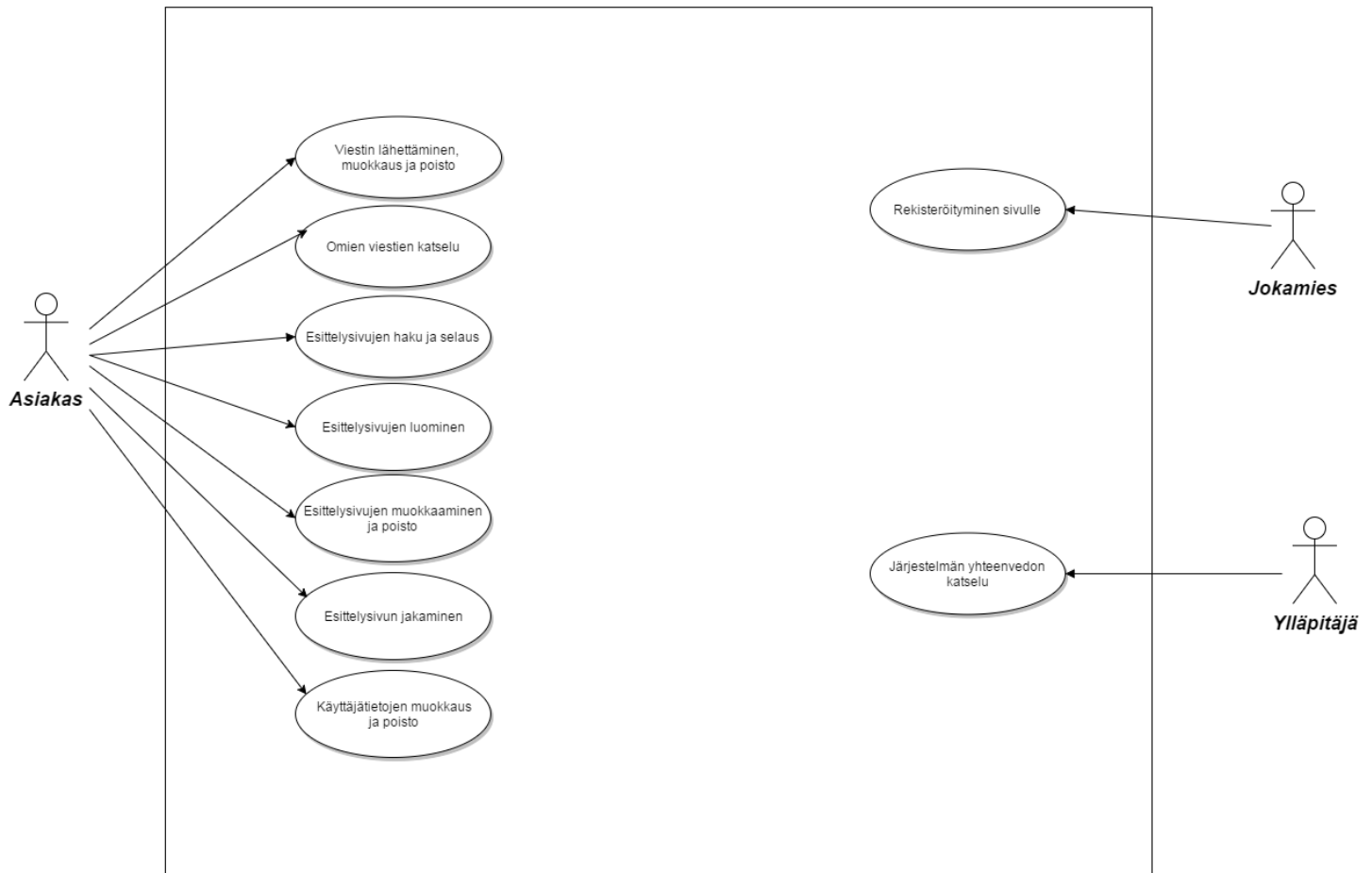
Johdanto

Työn aihe on ystävänvälityspalvelu web-sovellus jonka tarkoitus on auttaa asiakkaita löytämään sopivia ystäviä. Asiakkaat luovat esittelysivuja, jotka ovat joko julkisia tai yksityisiä. Yksityiset esittelysivut voidaan jakaa toisille. Jos asiakas näkee miellyttävän esittelysivun, voi hän lähettää sen kirjoittajalle viestin. Sovellusta ylläpitää ylläpitäjä, joka voi selailla järjestelmän yhteenvetoja. Järjestelmä pyrkii auttamaan yksinäisiä henkilöitä löytämään ystäviä.

Sovellus toteutetaan käyttäen Java-ohjelmointikieltä. Sovelluksen pohjana tulee toimimaan Tietokantojen perusteet -kurssilla käytetty tikape-runko, eli järjestelmä tulee käyttämään Spark ja Thymeleaf ohjelmistokehyksiä. Sovellus tulee käyttämään PostgreSQL tietokantaa ja se tullaan toteuttamaan Heroku-palvelimelle. Täten ohjelmisto edellyttää PostgreSQL tietokantaa eikä tietokannan vaihtaminen ole kovin helppoa. Käyttäjän selaimen tulee tukea javascript ohjelmointikieltä, jotta käyttöliittymä toimii ja näkyy halutulla tavalla.

Yleiskuva järjestelmästä

Käyttötapauskaavio



Käyttäjärühmät

Jokamies

Jokamies on kirjautumaton ja rekisteröitymätön käyttäjä, joka on tullut nettisivulle esimerkiksi mainoksien kautta.

Asiakas

Asiakas on rekisteröitynyt käyttäjä. Asiakas toimii palvelun peruskäyttäjänä, Se on samalla ystäväehdokas ja ystävän hakija.

Ylläpitäjä

Ylläpitäjä on PariTELE Oy:n työntekijä, joka voi tarkastella palvelun tietoja.

Käyttötapauskuvaukset

Jokamiehen käyttötapauskset

Rekisteröityminen, kirjautuminen.

Asiakkaan käyttötapauskset

Viestin lähettäminen, muokkaus ja poisto

Asiakas voi esittelysivujen selauksen yhteydessä lähettää viestin esittelysivun omistajalle. Hän voi päättää onko viesti anonyymi vai ei. Jos viesti ei ole anonyymi, voi vastaanottaja siihen vastata viestillä ja jakaa hänelle yksityisiä sivuja. Lähettämiään viestejä voi myös muokata ja poistaa.

Omien viestien katselu

Asiakas voi lukea omia saamiaan ja lähettämiään viestejä. Jos viesti ei ole anonyymi, voi hän vastata viestiin ja jakaa toisilleen yksityisiä sivuja.

Esittelysivujen haku ja selaus

Asiakas voi seuloa ja hakea haluamiansa ystäväehdokkaita heiden julkisten esittelysivujen perusteella. Seulominen tapahtuu profiilissa määriteltyjen hakutarkoitusten perusteella.

Esittelysivun luominen

Asiakas voi tehdä esittelysivuja, jotka ovat joko yksityisiä tai julkisia. Kaikki esittelysivut ovat asiakkaan itsensä-näköisiä ja muokattavia ja yksityisiä esittelysivuja voi jakaa tietyille henkilöille.

Esittelysivun jakaminen

Asiakas voi jakaa yksityisiä esittelysivujaan muille asiakkaille. Asiakas voi joko jakaa vain yhden tai vaikkapa kaikki yksityiset sivut valitsemalleen henkilölle, jos niin haluaa.

Muita käyttötapauksia ovat: kirjautuminen, omien esittelysivujen muokkaaminen ja poisto, käyttäjätietojen muokkaus ja poisto.

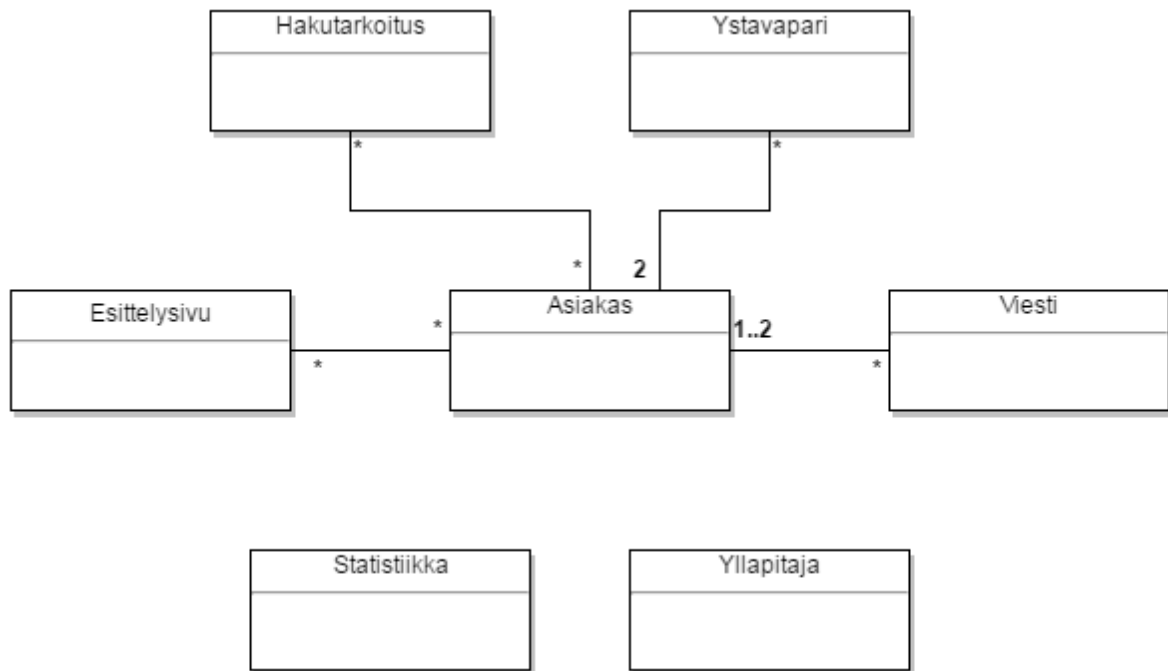
Ylläpitäjän käyttötapauskset

Järjestelmän yhteenvedon katselu

Ylläpitäjä pääsee katsomaan palvelun tietoja: kuinka monta kävijää palvelussa on ? kuinka monta ystäväparia on järjestelmässä ? Kuinka monta viestiä tai esittelysivua palvelussa on? Yhteenvedo vastaa muun muassa näihin kysymyksiin.

Muita käyttötapauksia ovat: kirjautuminen

Järjestelmän tietosisältö



Käsitekaavio

Tietokohde: Asiakas

| Attribuutti | Arvojoukko | Kuvailu |
|---------------|-----------------------------|---|
| Käyttäjänimi | Merkkijono, max 20 merkkiä | Käyttäjän nimimerkki |
| Salasana | Merkkijono, max 100 merkkiä | Käyttäjän salasana |
| Syntymäaika | Päiväys (Date) | Käyttäjän syntymäpäivä |
| Sukupuoli | Kokonaisluku, | 1 = Mies, 2= Nainen, 3= Jokin muu / En halua kertoa |
| Liittymisaika | Päiväys (Date) | Käyttäjän liittymispäivä |
| Osoite | Merkkijono, max 120 merkkiä | Asiakkaan osoite, esim: Katu 15B, Espoo, Suomi |

Asiakkaalla voi olla monia hakutarkoituksia ja monia esittelysivuja. Yksi esittelysivu kuuluu vain yhdelle asiakkaalle, mutta yksityiset esittelysivut voi olla jaettuna monille asiakkaille. Viestillä on vastaanottaja-asiakas ja sillä voi olla myös lähettäjä-asiakas. Ystäväparilla on kaksi asiakasta ja asiakkaalle voi kuulua monta ystäväparia.

Tietokohde: Ystäväpari (Ystavapari)

| Attribuutti | Arvojoukko | Kuvailu |
|-------------|----------------|------------------------------------|
| Aika | Päiväys (Date) | Aika milloin asiakkaista tuli pari |

Ystäväparilla on kaksi Asiakasta ja yksi asiakas voi kuulua moniin ystäväpareihin.

Tietokohde: Hakutarkoitus

| Attribuutti | Arvojoukko | Kuvailu |
|-------------|------------|---|
| Tarkoitus | Merkkijono | Ystävienhaku kriteeri (Miksi asiakas hakee ystäviä) |

Hakutarkoituksia voi asiakkaalla olla monia ja eri asiakkailla voi olla samoja hakutarkoituksia.

Tietokohde: Esittelysivu

| Attribuutti | Arvojoukko | Kuvailu |
|-------------|-------------------------------|---|
| Otsikko | Merkkijono, max. 100 merkkiä | Esittelysivun otsikko |
| Leipäteksti | Merkkijono, max. 1000 merkkiä | Esittelysivun leipäteksti, itse esittelysivun sisältö |
| Luotu | Aikaleima (Timestamp) | Sivun luontiaika |
| Muokattu | Aikaleima (Timestamp) | Sivun viimeisin muokkausaika |
| Julkinen | Totuusarvo | Onko sivu julkinen vai ei |

Asiakkaalla on monia esittelysivuja, mutta yksi esittelysivu kuuluu vain yhdelle asiakkaalle. Asiakas voi kuitenkin jakaa esittelysivut monelle henkilölle.

Tietokohde: Viesti

| Attribuutti | Arvojoukko | Kuvailu |
|---------------|------------------------------|--|
| Lähettiläjä | Kokonaisnumero | Lähettiläjän asiakkaan id, -1, jos lähettiläjä on anonyymi |
| Vastaanottaja | Kokonaisnumero | Vastaanottajan asiakkaan id |
| Sisältö | Merkkijono, max. 500 merkkiä | Viestin sisältö |
| Lähetetty | Aikaleima(Timestamp) | Milloin viesti on lähetetty |

Viestillä on vastaanottaja-asiakas ja mahdollisesti myös lähettiläjä-asiakas. Yhdellä asiakkaalla voi olla monta viestiä.

Tietokohde: Statistiikka (Ei tallenneta tietokantaan tälläisenään)

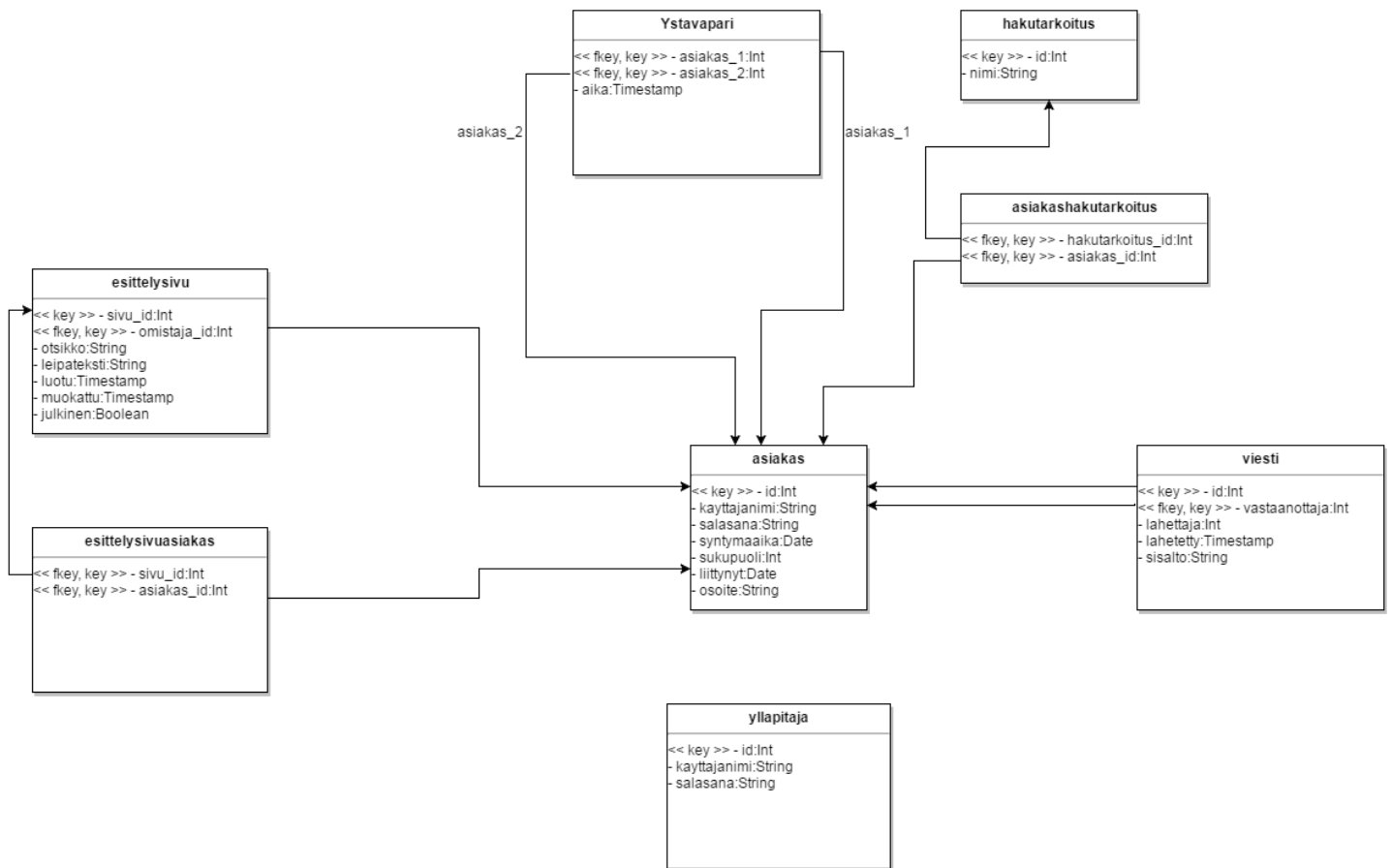
| Attribuutti | Arvojoukko | Kuvailu |
|--------------|----------------|--|
| Kayttajia | Kokonaisnumero | Kuinka monta käyttäjää järjestelmässä on |
| Viesteja | Kokonaisnumero | Kuinka monta viestiä järjestelmässä on |
| Sivuja | Kokonaisnumero | Kuinka monta esittelysivua järjestelmässä on |
| Ystavapareja | Kokonaisnumero | Kuinka monta ystäväparia järjestelmässä on |

Statistiikka mallintaa järjestelmän tieto-yhteenvetoa. Se näytetään ylläpitäjille.

Tietokohde: Ylläpitäjä

| Attribuutti | Arvojoukko | Kuvailu |
|--------------|-----------------------------|------------------------|
| Käyttäjänimi | Merkkijono, max 20 merkkiä | Ylläpitäjän nimimerkki |
| Salasana | Merkkijono, max 100 merkkiä | Ylläpitäjän salasana |

Relaatiotietokantakaavio



Relaatiotietokantakaavio

Järjestelmän yleisrakenne

Tietokantasovellusta tehdessä on noudatettu MVC-mallia. Näkymät sijaitsevat polussa **resources/templates**. Kaikki java-luokat on polussa **java/tsoha/ystavapalvelu**. Kontrollerit sijaitsevat kansiossa **controller**, joka sijaitsee java-polussa. Mallit sijaitsevat kansiossa **models**, joka sijaitsee java-polussa.

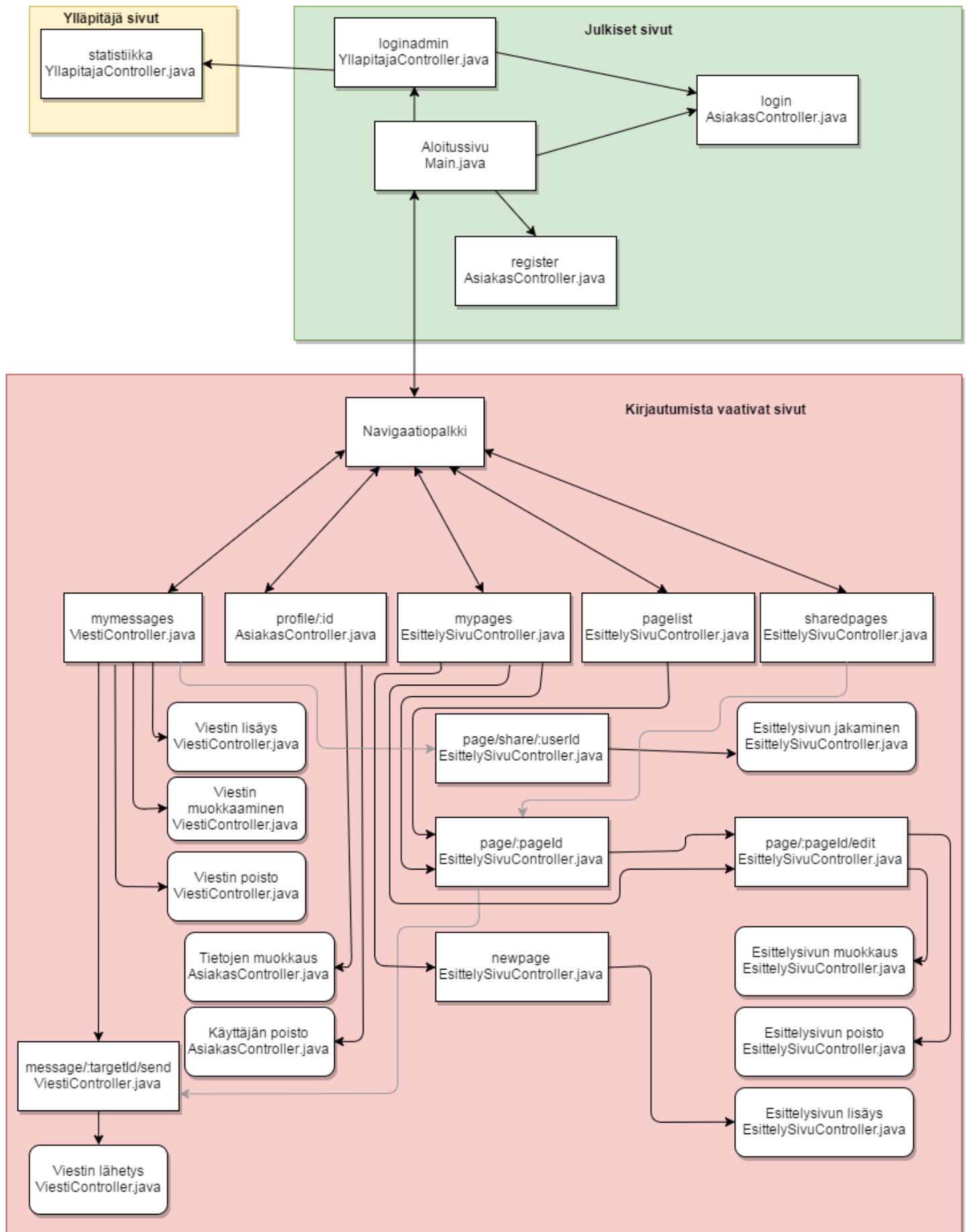
Apuluokkia on kansiossa **database**, joka sijaitsee java-polussa. SQL-lauseet löytyvät polusta **resources/sql** ja käyttöliittymän apukirjastot ovat polussa **resources/public/static**

Asetuksia ei ole.

Kaikkien näkymien tiedostonimet ovat kirjoitettu pienellä, Java-tiedostojen ensimmäinen kirjain on isolla.

Istuntoa on käytetty Sparkin `request.session()`:ia käyttämällä. Kun käyttäjä kirjautuu, laitetaan sessioon attribuutiksi "kayttaja" *Käyttaja*-olio. Jokaisessa kontrollerin polussa tarkistetaan onko session attribuuttia laitettu ja jos on niin se voidaan antaa näkymälle tai sillä voidaan hakea tietoa. Sama on tehty Ylläpitäjän kirjautumisessa *Yllapitaja*-oliolla sessioattribuuttiin "admin".

Käyttöliittymä ja järjestelmän komponentit



Asennustiedot

Lataa sovellus mille tahansa palvelimelle missä on git versionhallinta. Sovelluksen asentaminen Heroku palvelimelle tapahtuu käyttämällä heroku-toolbelt ohjelmaa (<https://devcenter.heroku.com/articles/heroku-cli>). Kun heroku-toolbelt ohjelman on asentanut ja siihen on kirjautunut käyttämällä komentoa **heroku login**, voi sovelluksen ladata nettiin. Sovelluksen juurikansiossa tulee suorittaa komento heroku create. Tämän jälkeen kuuluu suorittaa komento **git push heroku master**. Tämän jälkeen sovellus on ladattu Herokuun, ja sovellus antaa osoitteen mistä siihen voi päästä käsiksi selaimella. Tietokantakäyttäjiä ei tarvitse asettaa, sillä Heroku tekee ne puolestasi.

Käynnistys- / käyttöohje

Sovellus löytyy osoitteesta <https://protected-reef-49430.herokuapp.com/>

Sovellus tarvitsee kirjautumista toimiakseen täysin,
Testatessaan voi kirjautua seuraavalla tunnuksella sivulle

| Käyttäjänimi | Salasana |
|--------------|----------|
| toto | test |

Ylläpitäjä tunnukset (Yhteenvedon katsomiseen) ovat seuraavat:

| Käyttäjänimi | Salasana |
|--------------|----------|
| admin | admin2 |

Testaus, tunnetut bugit ja puutteet & jatkokehitysideat

Testaus

Kaikki testaus on tehty manuaalisesti.

Jatkokehitysideat

Sovellukseen voisi kirjoittaa testejä. Samoin tietoturvassa on hieman parannettavaa, Salasanat tallennetaan suoraan tekstinä tietokantaan joka ei ole hyvä käytäntö. Salasanojen hash:aaminen ja salt:taus olisi hyvä lisätä sovellukseen.

Omat kokemukset

Tietokantasovelluksen tekeminen oli yllättävän aikaa vievää, mutta ihan mukavaa. Helpointa tekemisessä oli itse sovelluksen koodaaminen. Eniten aikaa meni Thymeleaf templateiden tekemiseen, sillä Thymeleaf on super tarkka syntaksista. Opin paljon siitä, miten (ja miten ei) suunnitella web-sovellus.

Muu dokumentaatio

SQL-lauseet löytyvät polusta **resources/sql**.