

Niklas Luukkala

Aineopintojen harjoitustyö: Tietokantasovellus
(kevät 2015 periodi 4)

Blogipalvelu

- [1. Johdanto](#)
- [2. Yleiskuva järjestelmästä](#)
 - [Käyttötapauskaavio](#)
 - [Käyttäjäryhmät](#)
 - [Ei-sisäänkirjautuneet käyttäjät](#)
 - [Sisäänkirjautuneet käyttäjät](#)
 - [Käyttötapauskuvaukset](#)
 - [Ei-sisäänkirjautuneiden käyttäjien käyttötapaukset](#)
 - [Sisäänkirjautuneiden käyttäjien käyttötapaukset](#)
- [3. Järjestelmän tietosisältö](#)
- [4. Relaatiotietokantakaavio](#)
- [5. Järjestelmän yleisrakenne](#)
- [6. Käyttöliittymä ja järjestelmän komponentit](#)
- [7. Asennustiedot](#)
- [8. Käynnistys- ja käyttöohje](#)
- [9. Tunnetut bugit ja puutteet, jatkokehitysideat](#)
- [10. Omat kokemukset](#)

1. Johdanto

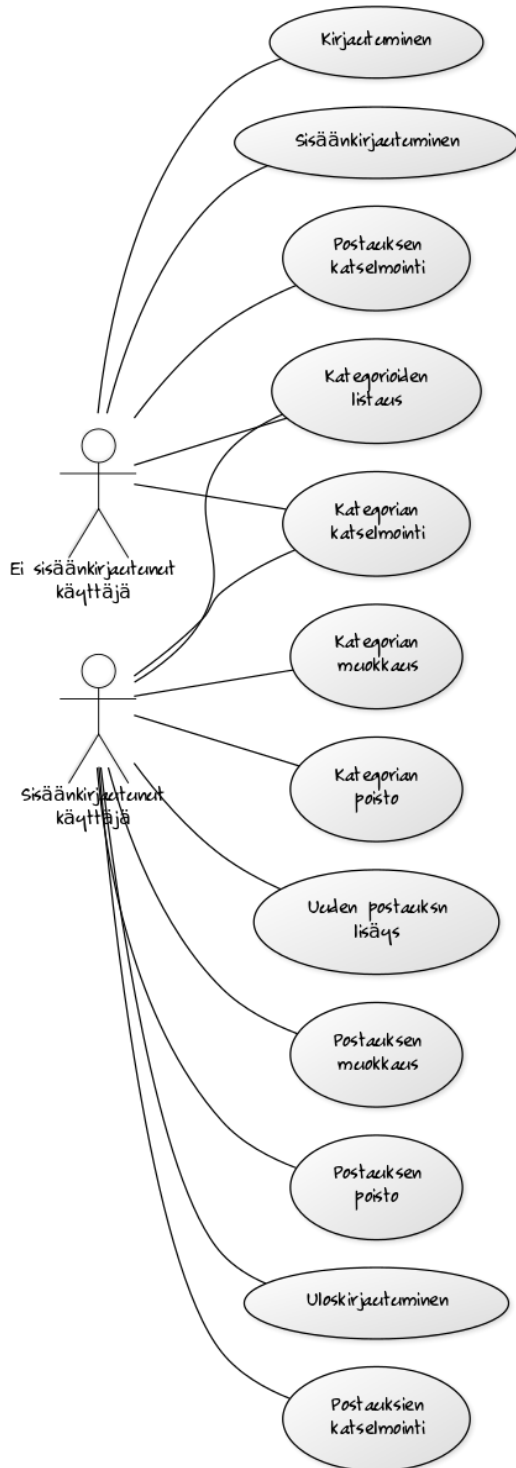
Blogi on blogimoottori yhden blogin pyörittämiseen. Käyttäjän pitää rekisteröityä tai kirjautua kirjoittaakseen uuden merkinnän. Postauksiin eli päiväkirjamerkintöihin voi liittää kategorioita.

Blogi on PHP:llä (frameworkina SLIM) ja PostgreSQL:llä toteutettu palvelu. Käyttää TSOHA-bootstrapia.

Toimintaympäristö on Helsingin Yliopiston users.cs-palvelin, Apache-palvelimen alla.

2. Yleiskuva järjestelmästä

Käyttötapauskaavio



Käyttäjärühmät

Ei-sisäänkirjautuneet käyttäjät

Käyttäjä, joka saattaa olla rekisteröitynyt mutta ei ole vielä kirjautunut blogiin.

Sisäänkirjautuneet käyttäjät

Käyttäjä, joka on rekisteröitynyt ja kirjautunut sisään.

Käyttötapauskuvaukset

Ei-sisäänkirjautuneiden käyttäjien käyttötapauskset

Rekisteröinti

Blogin luonnin jälkeen voi tehdä vain yhden käyttäjätunnuksen (voi itse muuttaa, ohjeet selviää dokumentaation myöhemmältä osiolta)

Sisäänkirjautuminen

Blogin postauksien katselmointi

Etusivulta katsoa postauksia tai klikata yksittäistä postausta ja katsoa se kokonaisuudessaan

Kategorioiden katselmointi

Kategoriat liittyvät blogipostauksiin, postauksella voi olla monia eri kategorioita. Katselmointi tarkoittaa että katsotaan kategoriaa ja siinä lukee sen kuvaus ja sitten on linkit blogipostauksiin jotka kuuluvat tähän kategoriaan.

Sisäänkirjautuneiden käyttäjien käyttötapauskset

Blogin postauksien katselmointi

Kuten edellä.

Kategorioiden katselmointi

Kuten edellä.

Kategorian lisäys

Samalla kun kirjoittaa uuden postauksen; kategorioita ei voi lisätä "erikseen"

Kategorian muokkaus

Kategorian poisto

Uuden postauksen lisäys

Uuden päiväkirjamerkinnän lisääminen

Postauksen muokkaus

Postauksen poisto

Uloskirjautuminen

3. Järjestelmän tietosisältö



Tietokohde: Käyttäjä

Attribuutti	Arvojoukko	Kuvailu
nimi	Merkkijono, max. 50 merkkiä	käyttäjän nimi
salasana	Merkkijono, max. 50 merkkiä	salattuna

Tietokohde: Postaus

Attribuutti	Arvojoukko	Kuvailu
id	Kokonaisluku	postauksen uniikki id
kayttaja	Merkkijono, max. 50 merkkiä	postauksen lähettäjä
pvm	Päivämäärä	päivämäärä milloin postaus on kirjoitettu
otsikko	Merkkijono, max. 200 merkkiä	200-merkinen otsikko
leipateksti	Tekstiä	TEXT muoto ettei tila lopu kesken
julkaistu	Merkki	yksi merkki (y tai n), näkykö postaus muille kuin blogin omistajalle

julkaistu-attribuutti on char, eikä boolean, koska olen lukenut kuinka rasittavaa PHP:llä ja PostgreSQL:llä on ymmärtää toistensa boolean arvoja.

Tietokohde: PostauksenKategoria

Attribuutti	Arvojoukko	Kuvailu
postausID	Kokonaisluku	postauksen uniikki id
kategoria	Tekstiä	kategorian nimi

Tietokohde: Kategoria

Attribuutti	Arvojoukko	Kuvailu
nimi	Tekstiä	kategorian nimi
kuvaus	Tekstiä	kategorian kuvaus

Kategorian nimi on TEXT-muodossa, sillä se oli yksinkertaisempi toteuttaa kuin varchar(50). Tämä teki validoinnista helpompaa, sillä en käyttänyt javascriptiä kategorioiden lisäykseen, vaan yksinkertaista input-laatikkoa mihin kirjoitetaan kategoriat pilkulla erotettuna. Kun käytin varcharia ja validoin, että nimi on varmasti oikean mittainen, niin error-teksti tuli liian myöhään ja virheellistä kategoriaa edeltävät kategoriat oli jo liitetty liitostaulussa.

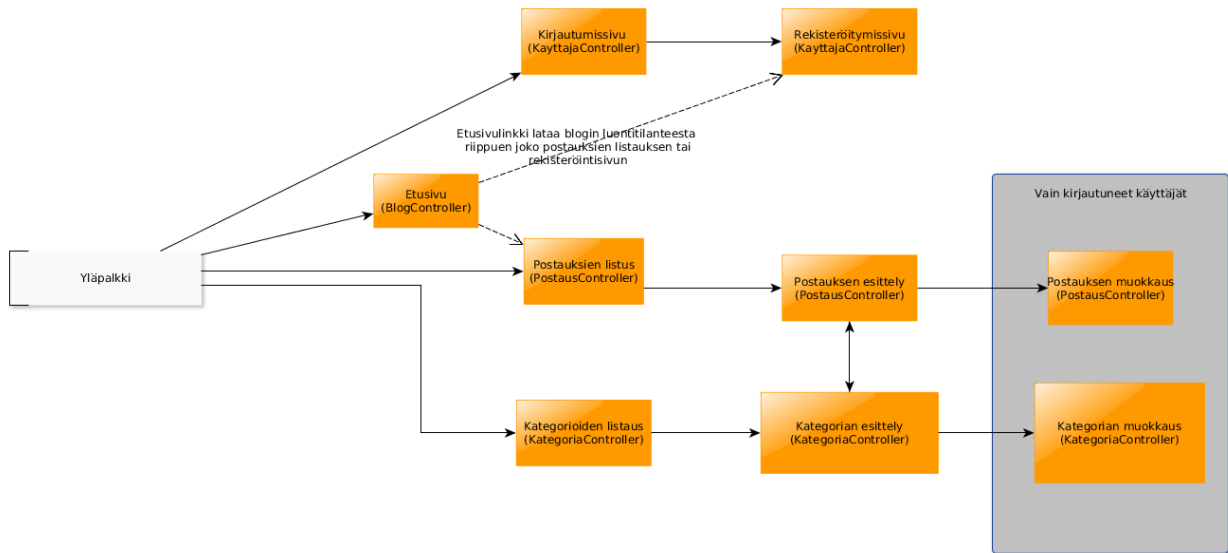
4. Relaatiotietokantakaavio



5. Järjestelmän yleisrakenne

Tietokantasovellusta tehdessä on noudatettu MVC-mallia. Kontrollerit, näkymät ja mallit sijaitsevat hakemistoissa controllers, views- ja models. Käytetyt apukirjastot on sijoitettu hakemistoon lib. Kaikki tiedostonimet on kirjoitettu pienellä.

6. Käyttöliittymä ja järjestelmän komponentit



7. Asennustiedot

Asenna sovellus kopioimalla sen tiedostot palvelimen nettiin näkyvään hakemistoon. Aseta sen jälkeen USERNAME ja PROJECT_FOLDER oikeiksi tiedostoon config/environment.sh.

Sitten pitää ajaa tietokannan rakennus koodi create_tables.sh

Sitten kirjautua nettisivulle ja rekisteröityä.

Jos haluaa että voi luoda useita käyttätunnuksia (enemmän kuin 1), niin kooditiedostoston kayttaja_controller.php rivin 19 muuttujan rekisterointi_auki arvo tulee muuttaa falsesta trueksi.

8. Käynnistys- ja käyttöohje

<http://xnix.users.cs.helsinki.fi/prujusto/>

käyttäjätunnuksen

nimi: niku

salasana: joo

9. Tunnetut bugit ja puutteet, jatkokehitysideat

Ei tunnettuja bugeja.

Seuraavaksi voisi toteuttaa etsi-ominaisuuden, postauksien sivutuksen ja käyttäjän salasanan kryptauksen.

10. Omat kokemukset

Tsoha on paras kurssi millä olen käynyt Helsingin Yliopistolla. Toteutin blogin 10 vuotta sitten PHP:llä ja PostgreSQL:llä, mutta puutteellisesti. Silloin sovelluksessani SQL-injektiot olivat olivat mahdollisia, kuten myös XSS-hyökkäykset. En myöskään osannut kunnolla relaatiotietokantojen perusteita, en esimerkiksi käyttänyt Katogoria ja Postaus taulujen välillä liitostaulua, enkä käyttänyt FOREIGN Key-juttuja ollenkaan, enkä tiennyt että PostgreSQL:ssä on automaattinen uniikki id -järjestelmä. Ennen en myöskään ymmärtänyt MVC-mallia vaikka yritin ymmärtää. Nyt pystyin vihdoinkin tekemään sovelluksen paremmalla tavalla. Kiitos kurssista