

lindeb

mau\Lu Link Database

Tietokantasovellus-harjoitustyö / Tulir Asokan

Sisällys

1 Johdanto	4
1.1 Tavoitteet ja toiminnallisuus	4
1.2 Teknologia	4
1.3 Toteutus- ja toimintaympäristö	5
2 Yleiskuva järjestelmästä	6
2.1 Käyttötapauskaavio	6
2.2 Käyttäjäryhmät	6
2.3 Käyttötapauskuvaukset	7
2.3.1 Jokaihmissen käyttötapaukset	7
2.3.2 Käyttäjän käyttötapaukset	7
3 Järjestelmän tietosisältö	8
3.1 Tietokohdekuvaukset	8
3.2.1 User	8
3.2.2 AuthToken	9
3.2.3 Link	9
3.2.4 Tag	9
4 Relaatiotietokantakaavio	10
5 Järjestelmän yleisrakenne	11
6 Käyttöliittymä ja sen komponentit	12
6.1 Linkkilista	12
6.2 Tagilista	12
6.3 Linkin tallennuskirjanmerkki	12
7 Asennustiedot	13
7.1 Kääntäminen	13

7.2 Konfiguraatio	13
8 Käynnistys- ja käyttöohje	14

1 Johdanto

1.1 Tavoitteet ja toiminnallisuus

Järjestelmän tavoitteena on sallia helppo linkkien tallentaminen ja tallennettujen linkkien selaaminen ja etsiminen.

Järjestelmän perustoiminnallisuuteen kuuluu linkkien tallennus ja tallennettujen linkkien hallinta, linkkien tagaaminen ja niiden selaaminen. Linkkien tallennuksen on onnistuttava tallennettavalta sivulta yhdellä napinpainalluksella, esimerkiksi kirjanmerkin tai selainlisäosan avulla.

Järjestelmä vaatii kirjautumisen. Käyttäjien lisäämä tieto pysyy yksityisenä vain käyttäjän nähtävänä.

Hakutoiminnon on kyettävä hakemaan tietoja myös sivun sisällön perusteella, eikä vain perustietojen kuten otsikon, sivuston ja tagien perusteella. Tämän lisäksi linkkejä on voitava järjestää ja suodattaa perustietojen perusteella.

Järjestelmään ei tule lisätä mitään sisäisiä sosiaalisia ominaisuuksia. Kaikki ihmisten välinen interaktio tapahtuu muissa palveluissa. Jos aikaa riittää, järjestelmään voidaan lisätä mahdollisuus jakaa linkkejä muihin palveluihin.

Järjestelmä on tarkoitus tulevaisuudessa integroida [mau\Lu](#)-palveluun.

1.2 Teknologia

Projektin palvelinpuoli toteutetaan [Go](#)-ohjelmointikielellä. Palvelimessa on REST-tyylinen HTTP/JSON-ohjelmointirajapinta. Itse ohjelman palvelimen lisäksi järjestelmä hyödyntää myös [Elasticsearch](#)-palvelinta monimutkaisempien hakujen sallimiseksi. Tietokantana käytetään [MariaDB](#)-tietokannanhallintajärjestelmää.

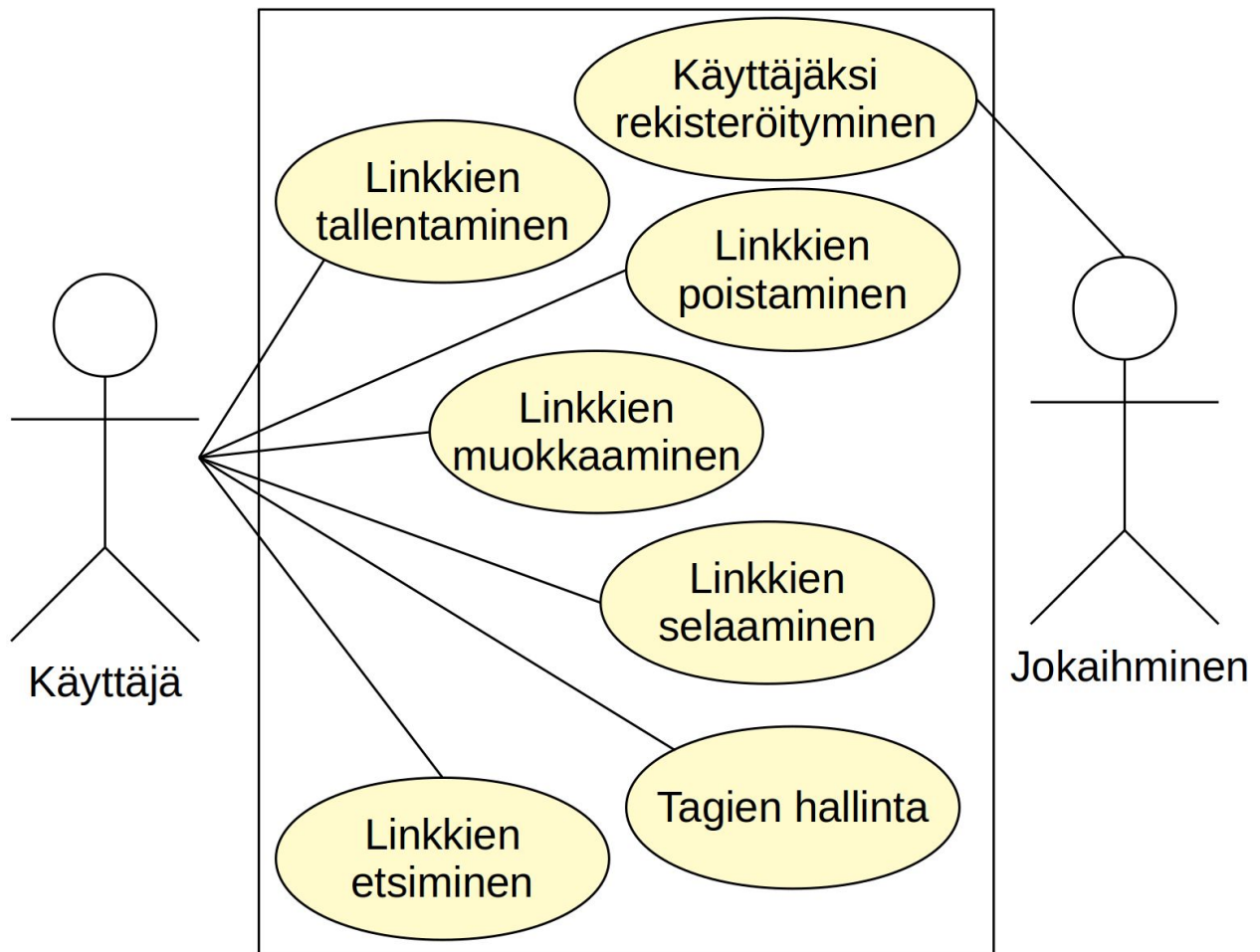
Käyttöliittymä toteutetaan web-ohjelmana [React](#)illa ja modernilla JavaScriptillä. Tyylitiedostot tehdään itse [Sass](#)-esikäsittelijällä. Käyttöliittymän toimivutta vanhoilla selaimilla ei taata. Firefoxin uusin versio on suositeltu, mutta järjestelmä testataan myös Chromen uusimmalla versiolla.

1.3 Toteutus- ja toimintaympäristö

Projektia kehitetään JetBrainsin [IDEA](#)-kehitysympäristöllä, johon on lisätty [Go-plugin](#). Sovelluksen voi teoriassa asentaa mille tahansa laitteelle, mille on Go-kääntäjä. Virallinen asennus löytyy osoitteesta lindeb.mau.lu.

2 Yleiskuva järjestelmästä

2.1 Käyttötapauskaavio



2.2 Käyttäjärühmät

Jokaihmisellä tarkoitetaan ketä tahansa, joka internetin välityksellä tulee katsomaan lindebin www-sivuja.

Käyttäjällä tarkoitetaan järjestelmään rekisteröityneitä käyttäjiä.

2.3 Käyttötapauskuvaukset

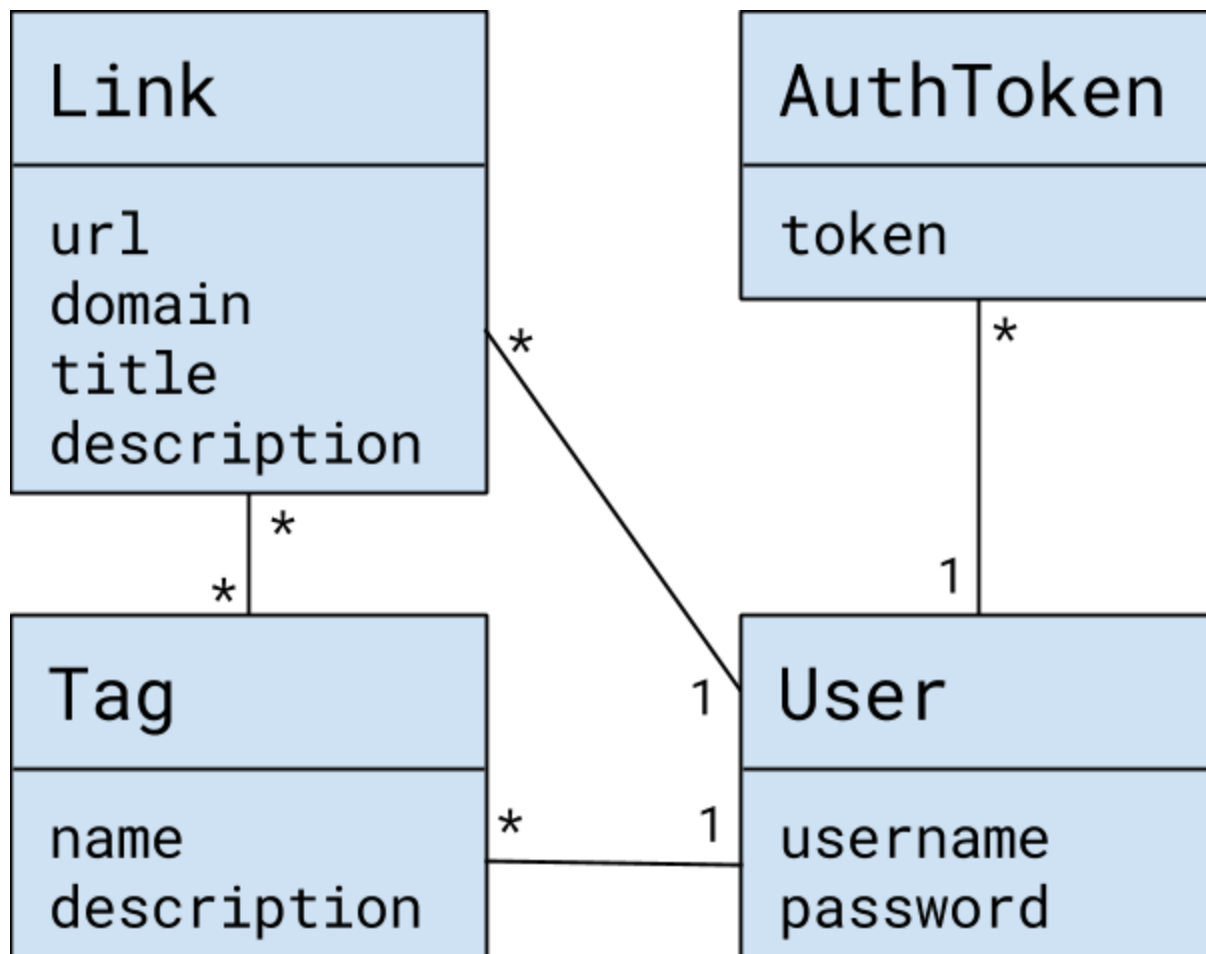
2.3.1 Jokaihmissen käyttötapauskset

- Rekisteröityminen: kuka tahansa voi rekisteröityä palvelun käyttäjäksi

2.3.2 Käyttäjän käyttötapauskset

- Linkkien tallentaminen: käyttäjä voi tallentaa mielenkiintoisia linkkejä myöhemmin luettavaksi
- Linkkien muokkaaminen: käyttäjä voi muokata linkkiä, esimerkiksi tallennushetkellä vahingossa jääneiden ylimääräisten kyselyparametrien poistamiseksi tai tagien lisäämiseksi
- Linkkien etsiminen: käyttäjä voi etsiä linkkejä sivun sisällön tai metadatan perusteella
- Linkkien selaaminen: käyttäjä voi selata, järjestellä ja suodattaa tallennettuja linkkejä perustietojen, kuten tagien ja verkkotunnuksen perusteella
- Tagien lisääminen: käyttäjä voi lisätä tageja manuaalisesti tai läpinäkyvästi linkkien hallinnan yhteydessä
- Tagien muokkaaminen: käyttäjä voi vaihtaa tagin nimeä ja kuvausta
- Muita käyttötapauskia: sisään- ja uloskirjautuminen, linkkien poistaminen, tagien listaaminen, tagien poistaminen

3 Järjestelmän tietosisältö



3.1 Tietokohdekuvaukset

3.2.1 User

Rekisteröitynyt käyttäjä.

Attribuutti	Tyyppi	Kuvaus
username	Merkkijono	Käyttäjän asettama käyttäjänimi. Enintään 32 merkkiä.
password	Merkkijono	bcrypt-hajautusfunktiolla suojattu salasana.

3.2.2 AuthToken

Käyttäjän kirjautumiskohtainen todennusmerkki.

Attribuutti	Tyyppi	Kuvaus
token	Merkkijono	SHA256-hajautusfunktiolla suojattu 64-merkkinen satunnaisgeneroitu merkkijono.

3.2.3 Link

Käyttäjän tallentama linkki, ja siihen liittyviä tietoja. Linkillä voi olla useita tageja, mutta vain yksi omistaja. Yhden omistajan rajoitus johtuu siitä, että omistaja voi vapaasti muokata esimerkiksi linkille tallennettua otsikkoa ja sen tageja.

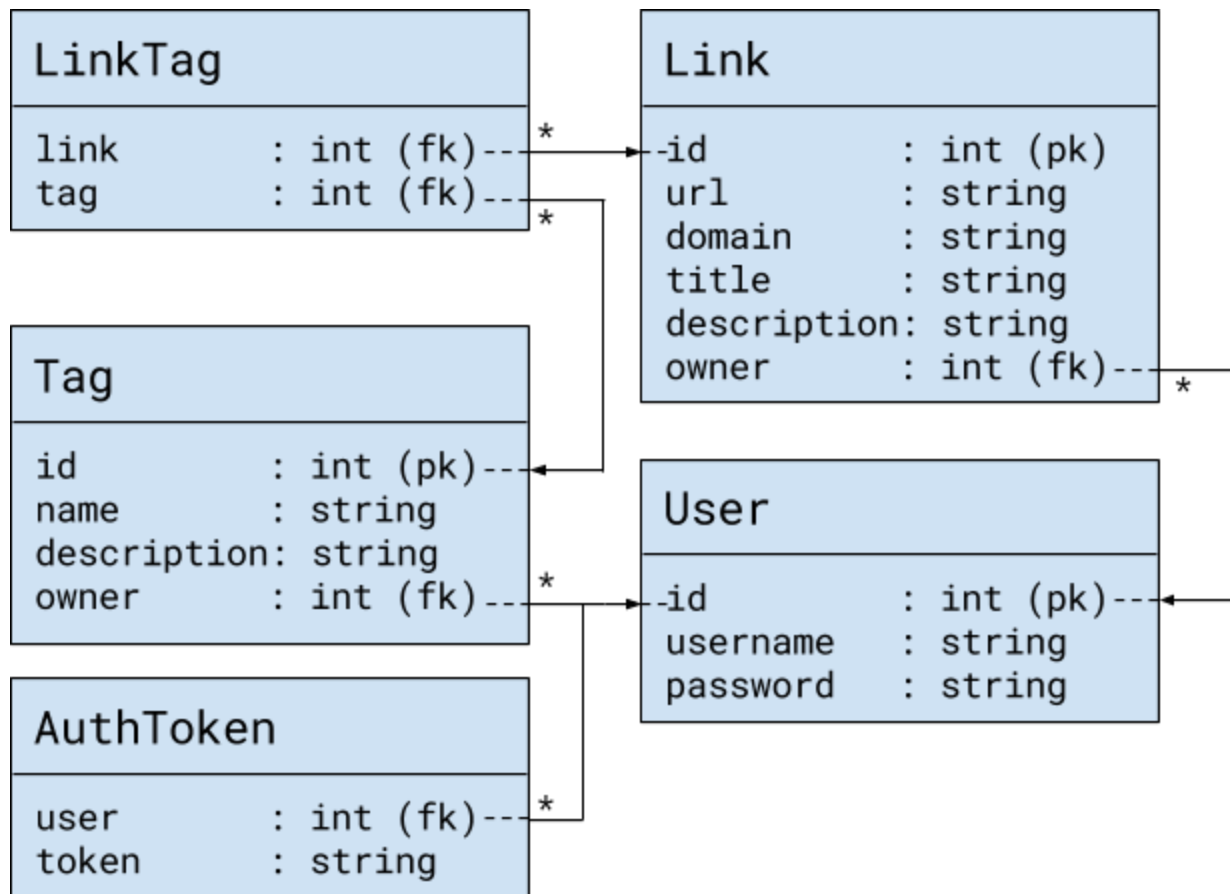
Attribuutti	Tyyppi	Kuvaus
url	Merkkijono	Koko linkki mukaanlukien protokolla ja kaikki parametrit.
domain	Merkkijono	Linkin domain-osa, joka näytetään käyttäjälle listauksissa.
title	Merkkijono	Linkin osoittaman sivun otsikko.
description	Merkkijono	Lyhyt kuvaus linkin osoittamasta sivusta.

3.2.4 Tag

Käyttäjäkohtainen tagi ja lyhyt kuvaus sille. Useampi linkki voidaan tagata samalla tagilla. Tagit ovat käyttäjäkohtaisia, sillä käyttäjä voi vapaasti määritellä tagin kuvauksen.

Attribuutti	Tyyppi	Kuvaus
name	Merkkijono	Tagin nimi, vain pieniä kirjaimia ja enintään 32 merkkiä.
description	Merkkijono	Tagin kuvaus.

4 Relaatiotietokantakaavio



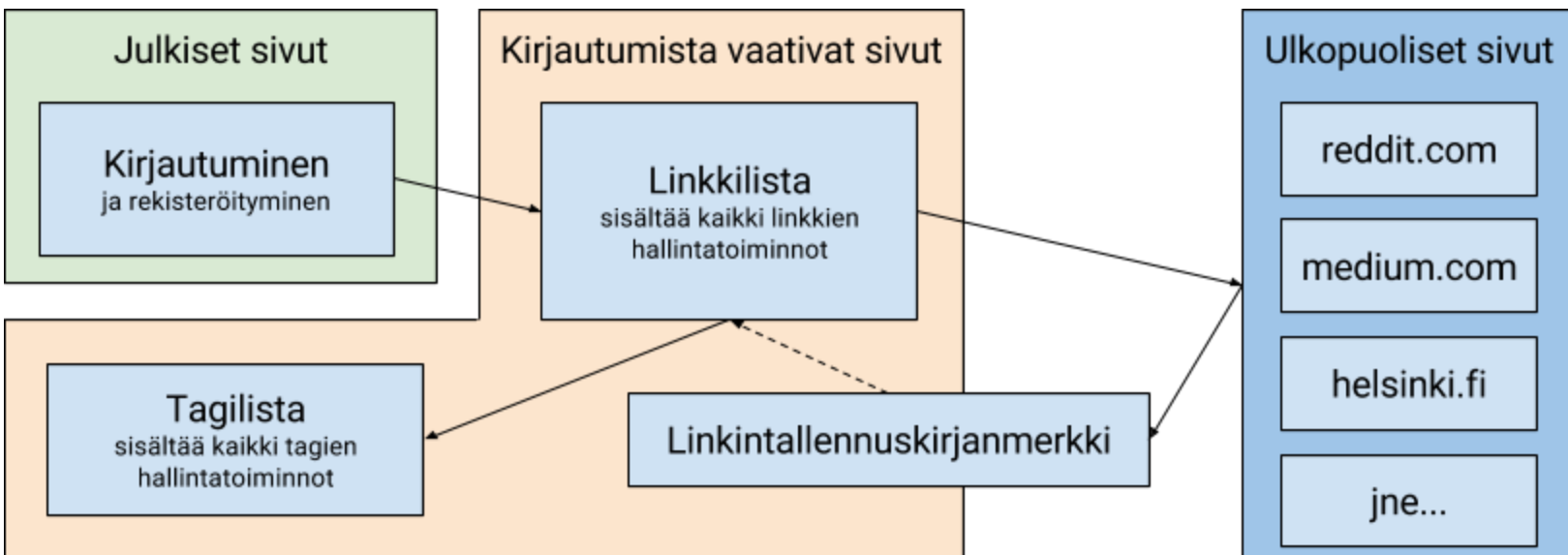
5 Järjestelmän yleisrakenne

Sovellus on jaettu Reactilla toteutettuun käyttöliittymään ja Go:lla toteutettuun palvelimeen. Sovelluksen rakenne muistuttaa hieman MVC-mallia, mutta täydelliseen MVC-mallin noudattamiseen ei ole pyritty, sillä samat hyödyt on saavutettu erottamalla käyttöliittymä ja palvelin toisistaan JSON-rajapinnan avulla.

Palvelin on jaettu kolmeen osaan: tietokantaa hallitsevaan "db"-pakettiin, HTTP/JSON-ohjelmointirajapinnan toteuttavaan "api"-pakettiin ja pääpakettiin, jossa on palvelimen konfiguraation lukeminen ja käynnistuksen hoitaminen. Palvelimessa on myös "util"-paketti, jossa on muiden pakettien vaatimia yksinkertaisia funktioita. Kirjastoja hallinnoidaan Go:n virallisella, mutta kokeellisella [dep](#)-työkalulla, joka sijoittaa kirjastot vendor-hakemistoon. Konfiguraatio laitetaan tiedostoon nimeltä `config.yaml`. Konfiguraatiotiedosto kannattaa luoda kopioimalla `example-config.yaml`-tiedosto.

Käyttöliittymä koostuu lähinnä React-komponenteissa kansiossa `src/components` ja Sass-tyylitiedostoista kansiossa `src/style`. Näiden lisäksi kansiossa `public` on sivuston logo ja metatietoja ja kansiot `config` ja `scripts` sisältävät Reactin konfiguraatio- ja skriptitiedostoja. Käyttöliittymällä itsellään ei ole varsinaista konfiguraatiota. Kirjastoja hallinnoidaan [npm](#)-pakettimanagerilla, joka sijoittaa kirjastot `node_modules`-hakemistoon.

6 Käyttöliittymä ja sen komponentit



6.1 Linkkilista

Linkkilista sisältää kaikki käyttäjän linkin sivujaolla. Linkkilistan ylä laidassa on hakupalkki, jonka avulla voi etsiä tiettyjä linkkejä. Hakupalkin avulla voi myös rajata haun esimerkiksi tiettyyn tagiin tai sivustoon.

Kun löytää haluamansa linkin, otsikosta painamalla voi avata linkin ja linkin tietoja voi muokata laatikon oikeassa yläkulmassa olevalla muokkausnapilla. Linkin poistaminen tapahtuu muokkausnäköymästä.

6.2 Tagilista

Tagilista sisältää kaikki käyttäjän tagit. Tagilistassa voi lisätä, muokata ja poistaa tageja.

6.3 Linkin tallennuskirjanmerkki

Linkin tallennuskirjanmerkki ei ole varsinainen sivu, vaan selaimessa oleva kirjanmerkki tai lisäosa, jota painamalla senhetkinen sivu lisätään käyttäjän linkkikokoelmaan.

7 Asennustiedot

7.1 Kääntäminen

Tämän vaiheen voi ohittaa, jos lataa Githubista valmiiksi käännetyt paketin [Releases](#)-osioista.

Palvelimen itse kääntämiseen on oltava Go-työkalupakki asennettuna ja oikein konfiguroituna. Tämän jälkeen projektin voi ladata ja kääntää komennolla

```
go get maunium.net/go/lindeb
```

jonka jälkeen projekti löytyy kansiota `$GOPATH/src/maunium.net/go/lindeb` ja käännetty palvelinohjelma löytyy tiedostosta `$GOPATH/bin/lindeb`.

Käyttöliittymän kääntäminen tapahtuu kansiossa `frontend` komennolla

```
npm run build
```

Käännön jälkeen staattiset tiedostot löytyvät käyttöliittymän kansiota `dist`.

7.2 Konfiguraatio

Konfiguraatiopohja löytyy tiedostosta `example-config.yaml`, josta sen voi kopioida vakiokonfiguraatitiedostoon `config.yaml`. Konfiguraatitiedoston polkua voi halutessaan muuttaa palvelinohjelman `--config-parametrillä`. Konfiguraatiopohjan pitäisi toimia tietokantakonfiguraatiota lukuunottamatta suoraan ainakin kehitysympäristössä. Konfiguraatiopohjassa on kommentit, jotka kertovat, mitä kenttiin pitäisi laittaa.

Jos et aktivoi palvelimen frontend-ominaisuutta, joudut käyttämään esimerkiksi Nginx-palvelinta lindeb-palvelimen ja käyttäjän välissä käyttöliittymän jakamiseen.

8 Käynnistys- ja käyttöohje

Kun palvelin on asennettu ja konfiguroitu oikein, sen voi käynnistää yksinkertaisesti ajamalla palvelinohjelman. Yleensä ajaminen tapahtuu komennolla `lindeb` tai `./lindeb`.

Palvelimen käynnistyttyä voit avata sen selaimella. Virallinen instanssi löytyy osoitteesta lindeb.mau.lu. Mitään vakiotunnuksia ei ole: luo omat tunnukset kirjautumissivulla.