

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	9
3.2.1 User	9
3.2.2 AuthToken	9
3.2.3 Link	9
3.2.4 Tag	10
3.2.5 Setting	10
4 Relaatiotietokantakaavio	11
5 Järjestelmän yleisrakenne	12
5.1 Palvelin	12
5.2 Käyttöliittymä	12
5.3 Selainlisäosa	13
6 Käyttöliittymä ja sen komponentit	14
6.1 Linkkilista	14

6.2 Tagilista	14
6.3 Asetukset	15
6.4 Linkin lisäys	15
6.4.1 Selainlisäosa	15
7 Asennustiedot	16
7.1 Kääntäminen	16
7.2 Konfiguraatio	16
8 Käynnistys- ja käyttöohje	18
9 Testaus, puutteet ja jatkokehitysideat	19
9.1 Puutteet ja bugit	19
9.2 Jatkokehitysideat	19
10 Omat kokemukset	21

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 jollain tavalla [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. MariaDB:n kanssa

yhteensopimattomia tietokannanhallintajärjestelmiä ei tueta, sillä SQL-kyselyt ovat kirjoitettu käsin.

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

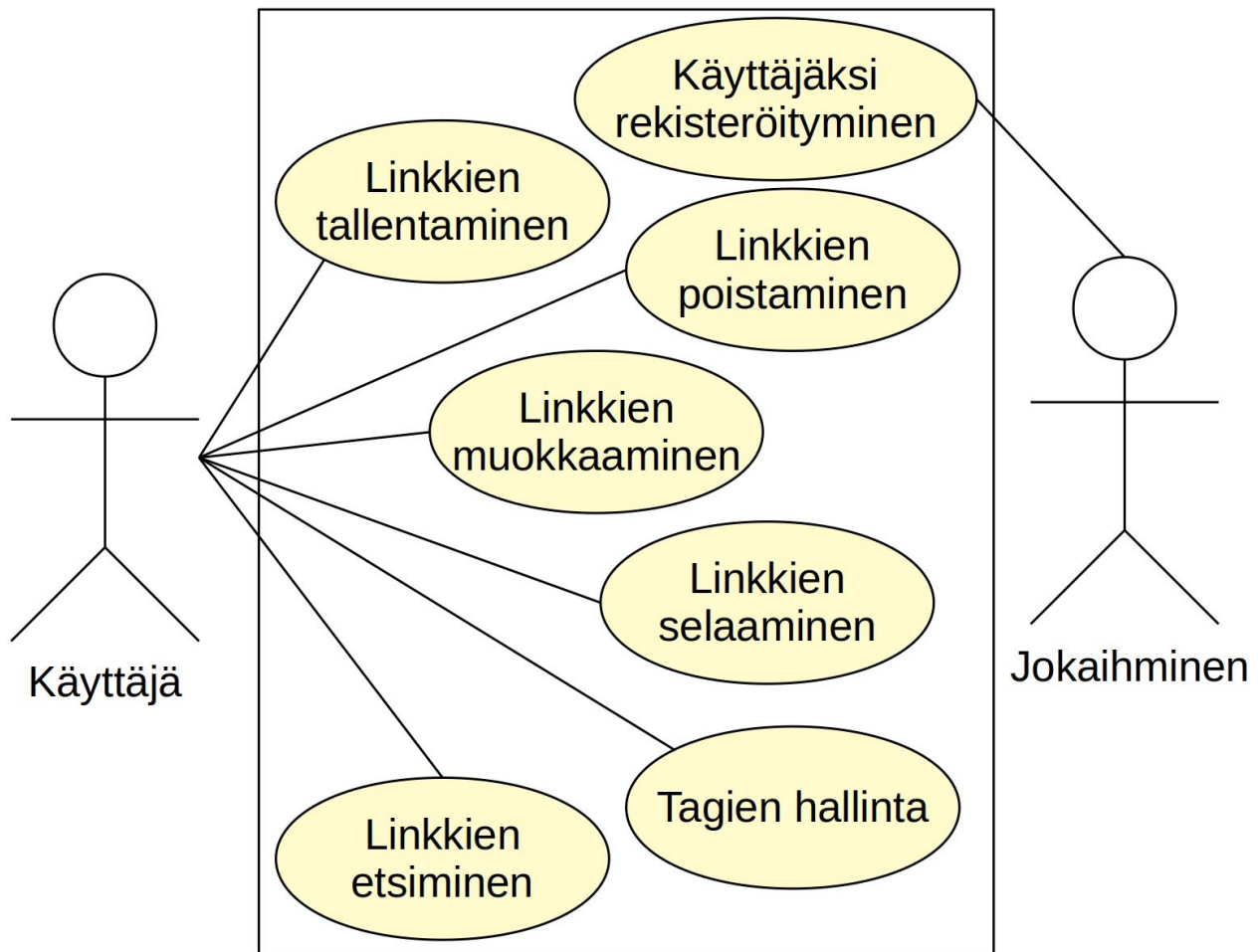
1.3 Toteutus- ja toimintaympäristö

Projektia kehitetään JetBrainsin [IDEA](#)-kehitysympäristöllä, johon on lisätty [Go-liitännäinen](#).

Palvelimen voi teoriassa asentaa mille tahansa laitteelle mille voi kääntää Go-koodia. 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

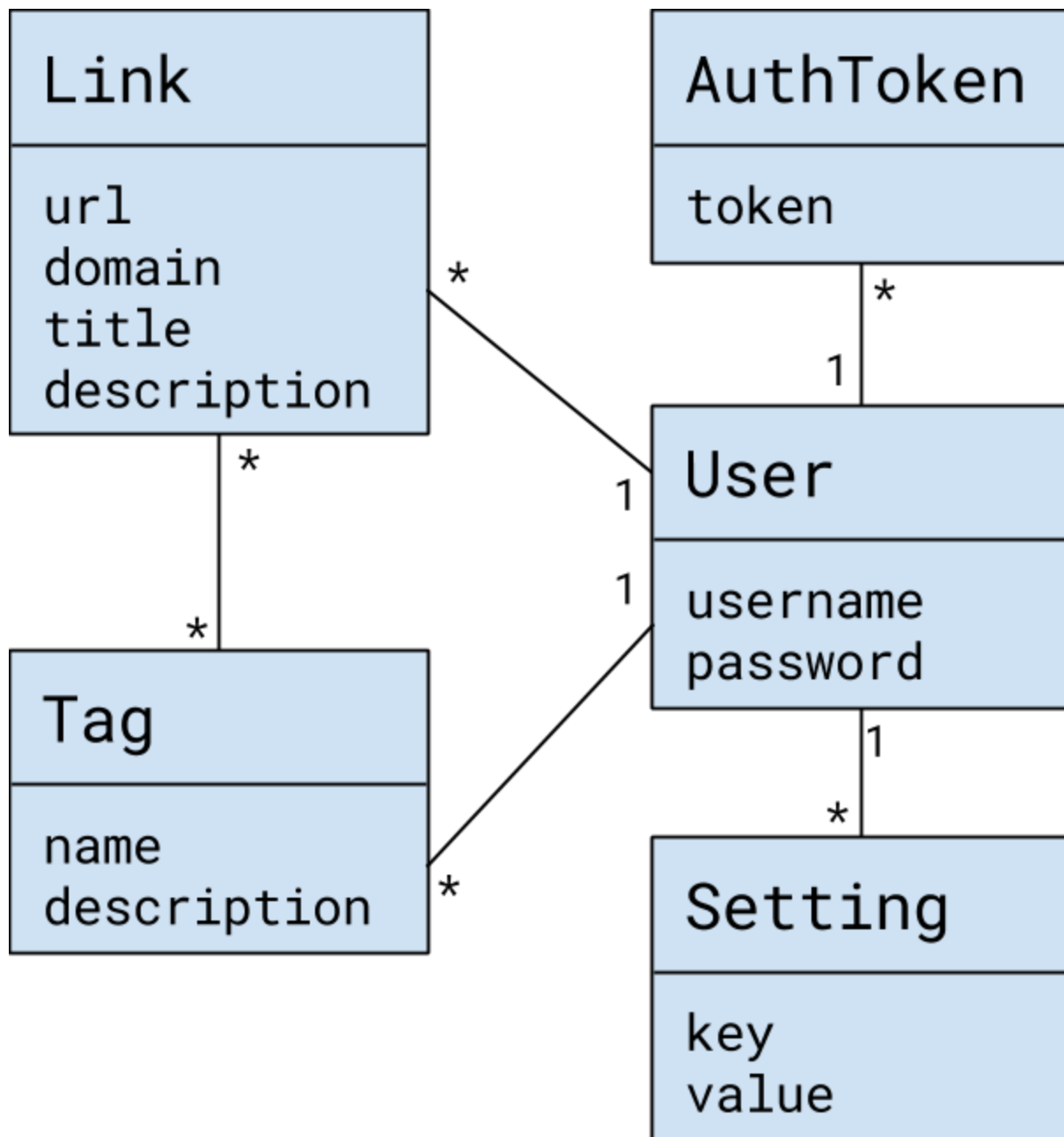
2.3.1 Jokaihmissen käyttötapauskset

- 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
- **Asetusten asettaminen:** käyttäjä voi säätää sovelluksen joidenkin osien toimintaa itselleen sopivammaksi. Asetukset synkronoituvat palvelimen kautta kaikille käyttäjän laitteille.
- 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. Aina tasan 60 merkkiä.

3.2.2 AuthToken

Käyttäjän kirjautumiskohtainen todennusmerkki.

Attribuutti	Tyyppi	Kuvaus
token	Merkkijono	SHA256-hajautusfunktiolla suojattu 64-merkinen 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.

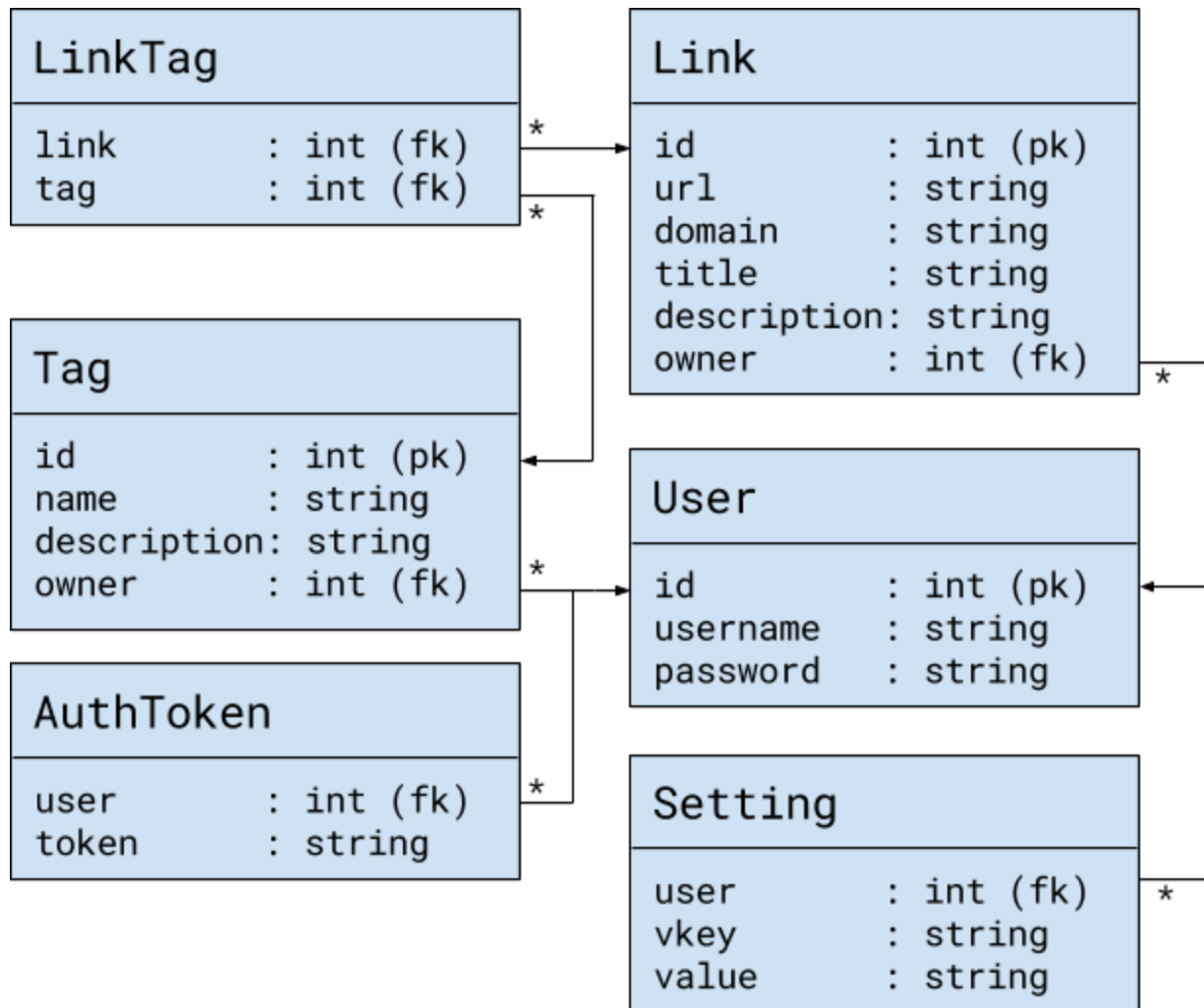
Attribuutti	Tyyppi	Kuvaus
name	Merkkijono	Tagin nimi, vain pieniä kirjaimia ja enintään 32 merkkiä.
description	Merkkijono	Tagin kuvaus.

3.2.5 Setting

Käyttäjän tallentama asetus.

Attribuutti	Tyyppi	Kuvaus
vkey	Merkkijono	Asetuksen avain. Varsinaisessa tietokannassa nimellä "vkey" SQL-avainsanojen välttämiseksi.
value	Merkkijono	Asetuksen arvo JSON-muodossa.

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.

5.1 Palvelin

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.

5.2 Käyttöliittymä

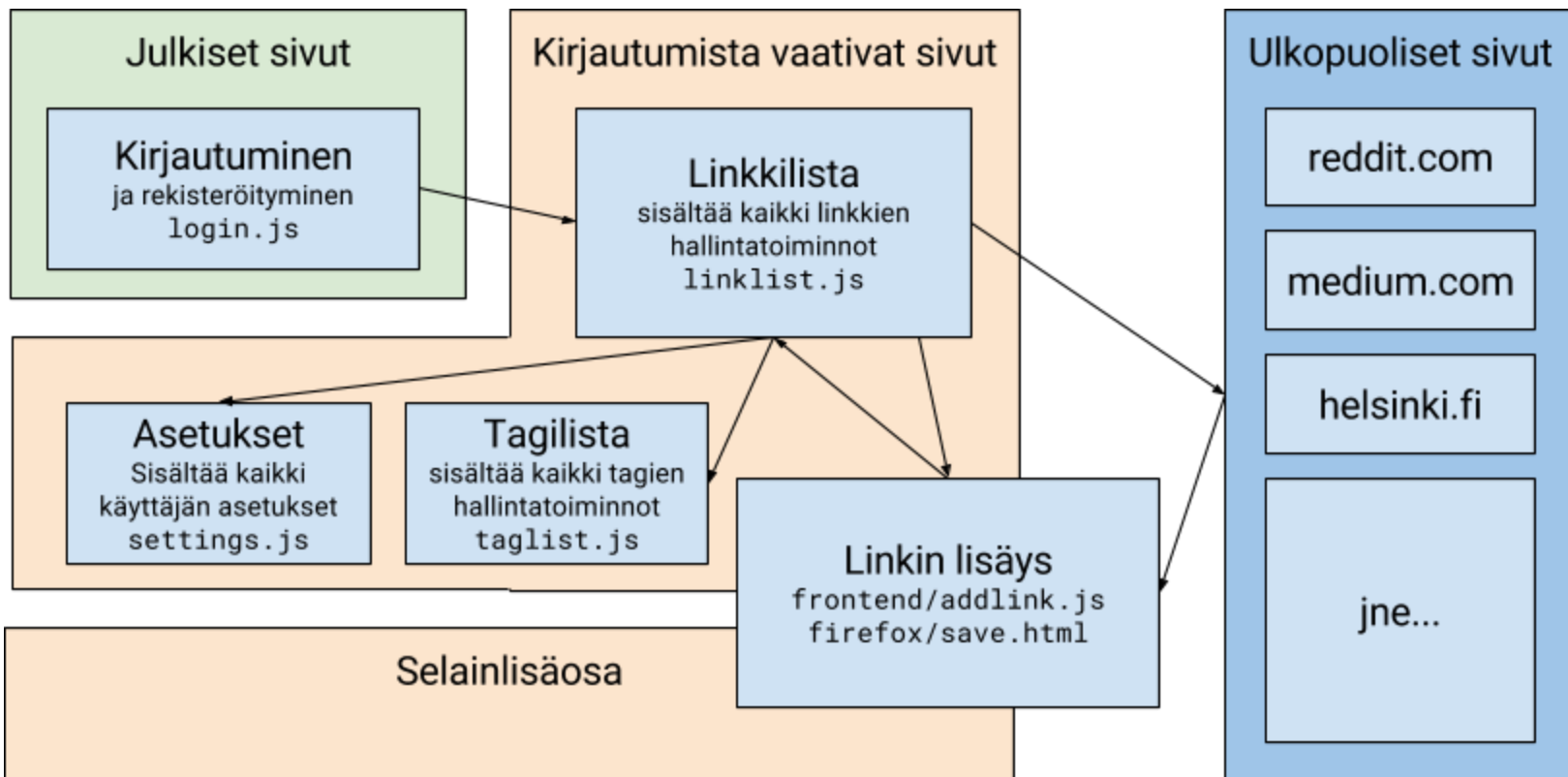
Käyttöliittymä koostuu lähinnä React-komponenteissa kansiossa `src/components` ja Sass-tyylitiedostoista kansiossa `src/style`. Näiden lisäksi kansiossa `public` on sivuston logo ja metatietoja ja kansiot `config` ja `scripts` sisältävät Reactin konfiguraatio- ja skriptitiedostoja. Käyttöliittymällä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

5.3 Selainlisäosa

Käyttöliittymän ja palvelimen lisäksi järjestelmään kuuluu myös selainlisäosa, jonka avulla voi tallentaa linkkejä nopeasti. Tällä hetkellä lisäosa tukee ainoastaan linkkien tallentamista, mutta tulevaisuudessa siihen voidaan lisätä muita ominaisuuksia.

Lisäosa on tällä hetkellä saatavilla vain Firefoxille, mutta sen saa helposti toimimaan myös Chromella, sillä myös Chrome tukee WebExtensionsin kaltaista lisäosarajapintaa.

6 Käyttöliittymä ja sen komponentit



6.1 Linkkilista

Linkkilista sisältää kaikki käyttäjän linkin sivujaolla. Linkkilistan yläaidassa 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 muokkausnäkyästä.

6.2 Tagilista

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

6.3 Asetukset

Käyttäjän määriteltävissä olevien asetusten ja käyttäjätietojen muuttaminen. Myös selainlisäosan asetukset löytyvät tästä näkymästä.

6.4 Linkin lisäys

Linkin lisäyssivulla lisätään uusia linkkejä. Sivulle pääsee linkkिलistasta. Nopeaa tallennusta varten on myös kirjanmerkki, jota painamalla linkin lisäyssivu aukeaa valmiiksi täytettynä.

6.4.1 Selainlisäosa

Linkin voi lisätä siirtymättä pois nykyiseltä sivulta käyttämällä selainlisäosaa, joka sisältää minikokoisen linkinlisäyssivun popup-tyylisenä ikkunana.

7 Asennustiedot

7.1 Kääntäminen

Tämän vaiheen voi ohittaa, jos lataa Githubista valmiiksi käännetyt paketin [Releases](#)-osiosta. Huomaa kuitenkin, että paketeissa tuleva palvelin toimii vain Linuxilla ja amd64-suoritinarkkitehtuurilla.

Palvelimen itse kääntämiseen on oltava Go-työkalupakki asennettuna ja oikein konfiguroituna. Tämän jälkeen projektin voi ladata ja kääntää suoraan 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 vaatimat kirjastot eivät ole valmiina repossa, sillä ne vievät huomattavasti enemmän tilaa. Saat asennettua vaaditut kirjastot kansiossa `frontend` komennolla

```
npm install
```

jonka jälkeen voit kääntää käyttöliittymän samassa kansiossa komennolla

```
npm run build
```

Kääntämisen jälkeen staattiset tiedostot löytyvät käyttöliittymän kansiota `dist`.

7.2 Konfiguraatio

Palvelimen konfiguraatiopohja löytyy tiedostosta `example-config.yaml`, josta sen voi kopioida vakiokonfiguraatitiedostoon `config.yaml`. Konfiguraatitiedoston polkua voi halutessaan muuttaa palvelimen `--config-parametrillä`. Konfiguraatiopohjan pitäisi toimia tietokantakonfiguraatiota lukuunottamatta suoraan

ainakin kehitysympäristössä. Pohjassa on kommentteja, jotka kertovat, mitä kenttiin pitäisi laittaa.

Valmiissa Debian-paketissa konfiguraatiotiedosto löytyy polusta `/etc/lindeb/config.yaml` ja valmiissa `.tar.xz`-paketissa konfiguraatiotiedosto on suoraan `config.yaml`-nimellä.

Jos poistat käytöstä palvelimen frontend-ominaisuuden, joudut käyttämään esimerkiksi Nginx-palvelinta lindeb-palvelimen ja käyttäjän välissä käyttöliittymän jakeluun.

8 Käynnistys- ja käyttöohje

Kun palvelin on asennettu ja konfiguroitu oikein, sen voi käynnistää yksinkertaisesti ajamalla palvelinohjelman. Yleensä ajaminen tapahtuu komennolla `lindeb` tai `./lindeb`. Jos asensit palvelimen Debian-paketin avulla, voit käynnistää palvelimen myös `systemd`:n avulla: `sudo systemctl start lindeb`.

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

9 Testaus, puutteet ja jatkokehitysideat

Käyttöliittymän kaikki toiminnallisuudet on testattu Firefoxin ja Chromen viimeisimmillä versioilla.

9.1 Puutteet ja bugit

Sovellus ei ole kunnolla optimoitu mobiililaitteille.

Selainlisäosan popupin sulkeminen ennen linkin tallennuksen onnistumista saattaa aiheuttaa tallennuksen keskeytymisen.

9.2 Jatkokehitysideat

Rekisteröityessä käyttäjältä voisi kysyä sähköpostin. Tallentamalla käyttäjän sähköposti voidaan toteuttaa salasanan resetointi sen unohtuessa. Lisäksi voidaan toteuttaa helpompi kirjautuminen mobiililaitteilla lähettämällä sähköpostiin linkki, jota painamalla pääsee kirjautumaan ilman salasanaa.

Hakua voisi parantaa käyttämällä Elasticsearchia paremmin. Haun lisäksi paremmat suodatus- ja järjestelyvaihtoehdot olisivat hyödyllisiä. Järjestelyyn voisi myös auttaa linkkien käyntikertojen seuraaminen.

Kirjautumisen turvallisuuden lisäämiseksi voidaan lisätä toinen tekijä kirjautumiseen (2-factor authentication), kuten [TOTP](#) (Time-based One-time Password) tai tuki [U2F](#) (Universal 2nd Factor) USB-laitteille. Kirjautumista vahvistaessa olisi myös hyvä antaa käyttäjälle mahdollisuus kirjautua ulos muilta laitteilta.

Sovellukseen voi lisätä joitain sosiaalisia toiminnallisuuksia, esimerkiksi linkkien jakamisen sosiaaliseen mediaan tai chat-ohjelmiin.

Selainlisäosaan voi lisätä toiminnallisuuksia kuten linkin muokkaamisen, kun on sivulla jonka linkin on lisännyt aiemmin. Lisäosan vaatimaa erillistä koodia voisi vähentää

käyttämällä Reactia myös siinä, sillä silloin koodin voisi jakaa nettisivun kanssa.
Lisäosasta voisi myös tehdä Chromelle version.

10 Omat kokemukset

Sovellusta tehdessä opin käyttämään Reactia paremmin ja tutustuin Elasticsearch-hakujärjestelmään.