

Dokumentaatio: Algoritmitietokanta

Ville Saarinen
Kesä 2017, Tietokantasovellus-harjoitustyö
Helsingin yliopisto

Contents

1. Johdanto	2
Järjestelmän kuvaus	2
Toteutus-/toimintaympäristö	3
2. Yleiskuva järjestelmästä	3
Käyttäjryhmät	3
Käyttötapauskuvaukset	4
3. Järjestelmän tietosisältö	6
Käsitekaavio	6
4. Relaatiotietokantakaavio	8
5. Käyttöohje	9

1. Johdanto

Järjestelmän kuvaus

Harjoitustyön aiheena on **algoritmitietokanta**, jonka on tarkoitus auttaa algoritmeista ja tietokoneella suoritettavasta ongelmanratkaisusta kiinnostuneita löytämään helposti ja nopeasti tietoa erilaisista algoritmeista, niiden ominaisuuksista ja konkreettisista toteutustavoista.

Tavoitteena on, että järjestelmään kirjautunut käyttäjä voi vaivattomasti hakea järjestelmään tallennettuja algoritmeja, niiden toteutuksia eri ohjelmointikielillä ja tärkeimpiä ominaisuuksia. Lisäksi tarkoituksena on tarjota käyttäjille mahdollisuus lisätä tietokantaan uusia algoritmeja, algoritmien aika-analyysejä ja toteutustapoja, ja tällä tavalla kerryttää yhteistä joukkoistettua tietopankkia, josta on hyötyä kaikille käyttäjille.

Algoritmeja voi hakea algoritmityypin, tagin, aikavaatimuksen tai toteutuskielen perusteella. Jokaiseen algoritmiin voi myös linkittää siihen läheisesti liittyviä muita algoritmeja.

Järjestelmän ylläpitäjällä on oma liittymä, jonka kautta hän ylläpitää järjestön käyttäjien jäsentietoja, siivoaa kirjoituskantaa ja määrittelee tarvittaessa uusia algoritmityyppejä, joiden perusteella kirjoituksia voi ryhmitellä.

Toteutus-/toimintaympäristö

Järjestelmä toteutetaan Helsingin yliopiston tietojenkäsittelylaitoksen **users**-palvelimella, tietokantana on **PostgreSQL** ja webpalvelinympäristössä ohjelmointikielenä on PHP. Ohjelman tietokanta ei ole vaihdettavissa, ja ohjelma toimii useimmissa selaimissa (ei vaatimusta JavaScript-tuelle).

2. Yleiskuva järjestelmästä

Käyttäjäryhmät

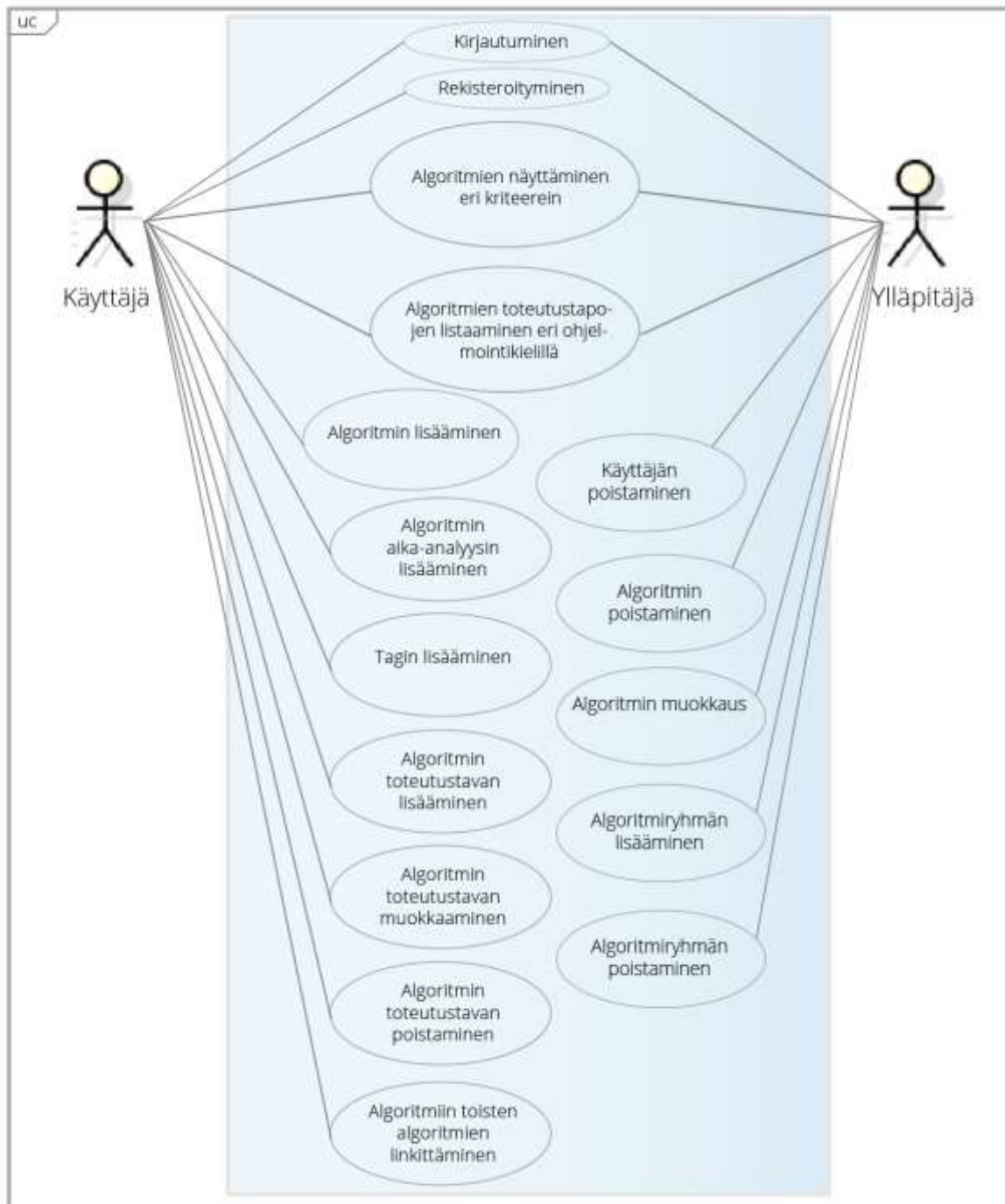
Käyttäjä

Käyttäjällä tarkoitetaan algoritmitietokannan käyttäjää.

Ylläpitäjä

Ylläpitäjän rooli sisältää tavallisen käyttäjän roolin lisäksi ylläpitoon liittyviä tehtäviä ja toiminnallisuuksia; algoritmien ja käyttäjien poistoa sekä muokkaamista

Algoritmitietokanta Käyttötapauskaavio



Käyttötapauskuvaukset

Käyttäjän käyttötapaukset

- **Algoritmien näyttäminen eri kriteerein:**

Käyttäjä valitsee kriteerin jonka mukaan hän haluaa algoritmit järjestettävän, jonka jälkeen hänelle näytetään listaus kriteerit täyttävistä algoritmeista
Esimerkki: Minkälaisia järjestysalgoritmeja on olemassa?

- **Algoritmin toteutustapojen listaaminen eri ohjelmointikielillä:**
Käyttäjä valitsee algoritmin, jonka jälkeen hänelle näytetään kyseisen algoritmin toteutustavat eri ohjelmointikielillä
Esimerkki: Miten Ford-Fulkerson algoritmi voidaan toteuttaa Ruby:lla?
- **Algoritmin lisääminen:**
Käyttäjä valitsee toiminnon "lisää uusi algoritmi", jonka jälkeen hän voi lisätä uuden algoritmin lyhyen kuvauksen kera ja oheistietoineen tietokantaan
- **Algoritmin aika-analyysin lisääminen:**
Käyttäjä valitsee toiminnon "lisää uusi aika-analyysi", jonka jälkeen hän voi lisätä haluamalleen algoritmille uuden aika-analyysin
- **Tagin lisääminen:**
Käyttäjä lisää uuden tagin algoritmille
Esimerkki: Lisää Mergesort algoritmille tagi "Divide and Conquer"
- **Algoritmin toteutustavan lisääminen:**
Käyttäjä valitsee toiminnon "lisää uusi toteutustapa", jonka jälkeen hän voi lisätä tietylle algoritmille uuden ohjelmointikielikohtaisen toteutustavan
- **Algoritmin toteutustavan muokkaaminen:**
Käyttäjä muokkaa aiemmin lisäämänsä ohjelmointikielikohtaista toteutusta
- **Algoritmin toteutustavan poistaminen:**
Käyttäjä poistaa aiemmin tekemänsä ohjelmointikielikohtaisen toteutustavan
- **Algoritmiin toisten algoritmien linkittäminen:**
Käyttäjä lisää algoritmille linkkejä toisiin relevantteihin algoritmeihin

Muita käyttötapauksia: rekisteröityminen, kirjautuminen

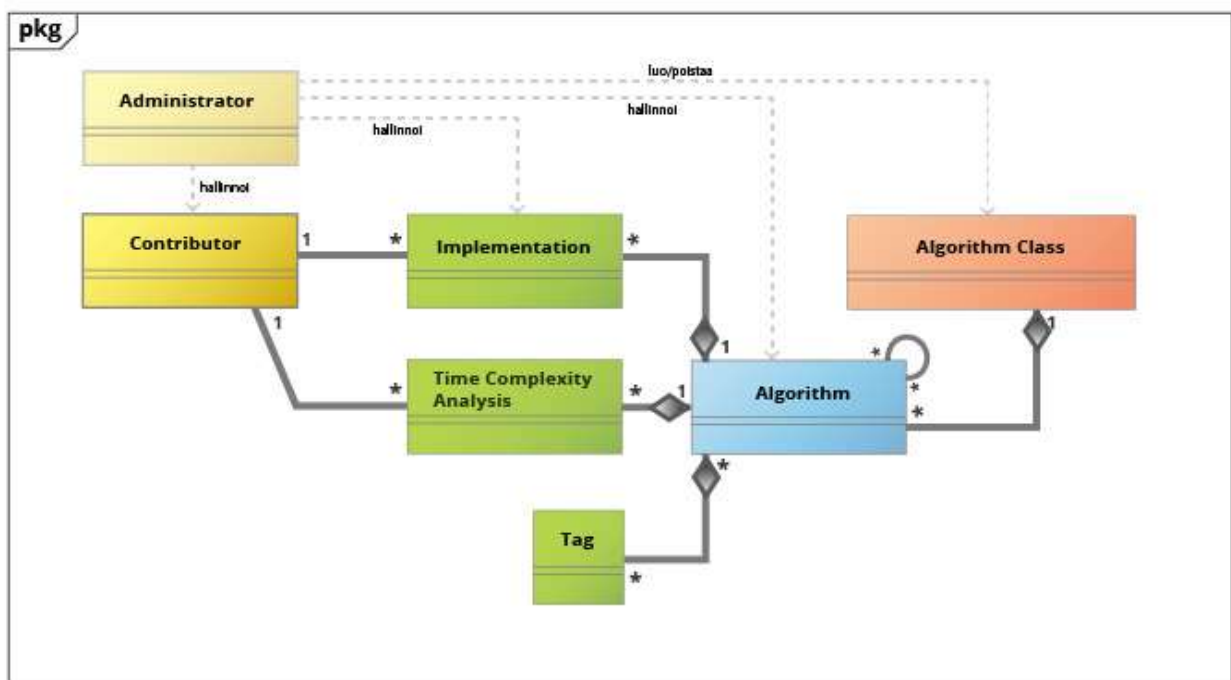
Ylläpitäjän käyttötapaukset

- **Käyttäjän poistaminen:**
Ylläpitäjä poistaa käyttäjän algoritmitietokannasta
- **Algoritmin poistaminen:**
Ylläpitäjä poistaa valitsemaansa algoritmin tietokannasta
- **Algoritmin muokkaus:**
Ylläpitäjä muokkaa valitsemaansa algoritmia

- **Algoritmiryhmän lisääminen:**
Ylläpitäjä lisää uuden algoritmiryhmän/tyypin tietokantaan
- **Muita käyttötapauksia:** kirjautuminen, algoritmien näyttäminen eri kriteerein, Algoritmien toteutustapojen listaaminen eri ohjelmointikielillä, algoritmiryhmän poisto

3. Järjestelmän tietosisältö

Käsitekaavio



Tietokohde: Algorithm

Attribuutti	Arvojoukko	Kuvailu
Name	Merkkijono, max 120 merkkiä	Algoritmin nimi, esim "Ford Fulkerson algoritmi"
Time Complexity	Merkkijono, max 30 merkkiä	Algoritmin aikavaatimus muodossa: "O(nlogn)"
Year	Merkkijono, 4 merkkiä	Algoritmin keksintävuosi, esim "2001"
Author	Merkkijono, max 120 merkkiä	Algoritmin alkuperäinen keksijä

Jokaiseen Algorithm-luokan ilmentymään liittyy yksi Algorithm Class-luokan ilmentymä, mutta jälkimmäiseen voi kuulua mielivaltainen määrä Algorithm-luokan ilmentymiä. Algoritmiluokkaan on mahdollisuus liittää mielivaltainen määrä sekä Time Complexity Analysis- että Implementation-

luokkien ilmentymiä. Nämä ilmentymät ovat kuitenkin riippuvaisia nimenomaisesta Algorithms-luokan ilmentymästä. Jokaiseen Algorithm-luokan ilmentymään voi liittyä myös mielivaltainen määrä Tag-luokan ilmentymiä. Lisäksi Algorithm luokan ilmentymiin voi liittyä mielivaltainen määrä muita saman luokan ilmentymiä. Administrator luokan ilmentymä, eli järjestelmän ylläpitäjä voi myös poistaa algoritmeja tietokannasta.

Tietokohde: Class

Attribuutti	Arvojoukko	Kuvailu
Name	Merkkijono, max 120 merkkiä	Algoritmiluokan nimi, esim "Sorting algorithm"

Jokaiseen Class-luokan ilmentymään voi kuulua mielivaltainen määrä Algorithm-luokan ilmentymiä

Tietokohde: Contributor

Attribuutti	Arvojoukko	Kuvailu
Username	Merkkijono, max 15 merkkiä	Käyttäjänimi, esim "Kalle Oittinen"
Password	Merkkijono, max 30 merkkiä	Salasana
Administrator	Totuusarvo/Boolean	Totta, jos ylläpitäjän oikeudet

Contributor-luokan ilmentymät voivat luoda, päivittää ja poistaa sekä Implementation- että Analysis-luokkien ilmentymiä. Jokaiseen Contributor-luokkaan voi liittyä siis mielivaltainen määrä molempia. Jokaiselle Implementation- ja Analysis-luokkien ilmentymistä on kuitenkin olemassa liitos tasan yhteen Contributor-luokan ilmentymään.

Tietokohde: Implementation

Attribuutti	Arvojoukko	Kuvailu
Programming Language	Merkkijono, max 30 merkkiä	Ohjelmointikielen nimi, esim "Python"
Date	Aikaleima	Aikaleima, joka kertoo milloin kohde on tehty muodossa: pp.kk.vuosi
Description	Merkkijono, max 2000 merkkiä	Varsinainen algoritmin kuvaus tekstinä

Jokainen Implementation-luokan ilmentymä on liitetty sekä yhteen Algorithms- että Contributor-luokan ilmentymään.

Tietokohde: Analysis

Attribuutti	Arvojoukko	Kuvailu
Time Complexity	Merkkijono, max 30 merkkiä	Analyysin tulos iso O-notaatio-muodossa
Date	Aikaleima	Aikaleima, joka kertoo milloin kohde on tehty muodossa: pp.kk.vuosi

Description	Merkkijono, max 4000 merkkiä	Varsinainen analyysin kuvaus tekstinä
-------------	------------------------------	---------------------------------------

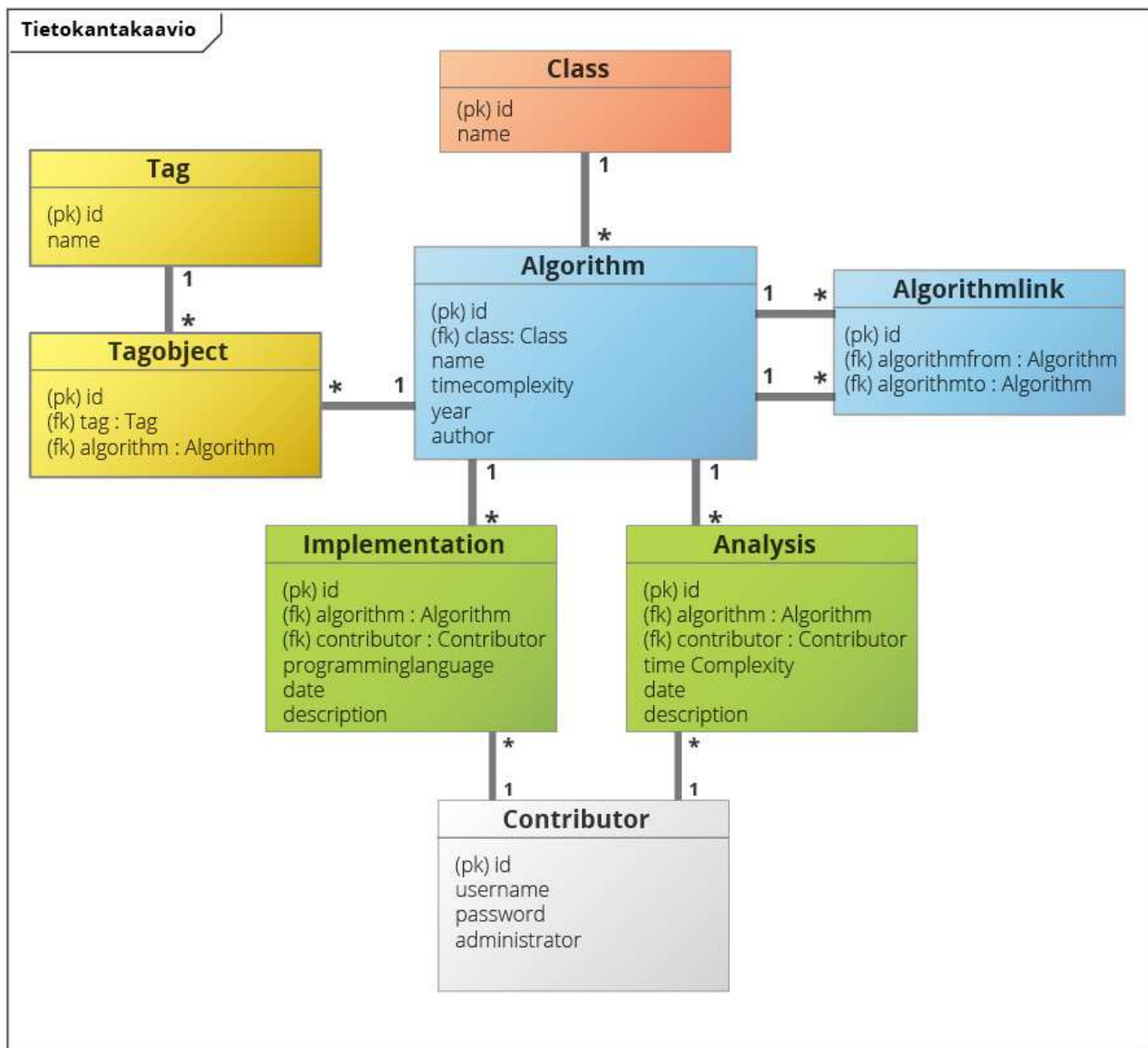
Jokainen Analysis-luokan ilmentymä on liitetty sekä yhteen Algorithms- että Contributor-luokan ilmentymään.

Tietokohde: Tag

Attribuutti	Arvojoukko	Kuvailu
Name	Merkkijono, max 30 merkkiä	Tagin nimi

Jokaiseen Tag-luokan ilmentymään voi liittyä mielivaltainen määrä Algorithms-luokan ilmentymiä, ja jokaiseen Algorithm-luokan ilmentymään voi liittyä myös mielivaltainen määrä Tag-luokan ilmentymiä.

4. Relaatiotietokantakaavio



*Contributor sisältää nyt myös administrator-attribuutin, joka on tyypiltään **boolean**. Tämän avulla käyttäjät voidaan eritellä ylläpitäjiin ja tavallisiin käyttäjiin, jolloin erillistä **Administrator-taulua ei tarvita**.*

5. Käyttöohje

Algoritmitietokannan aloitussivu sijaitsee osoitteessa: <http://saaville.users.cs.helsinki.fi/tsoha/>

Algoritmitietokannan yläreunassa on navigointiin tarkoitettu paneeli, josta pääsee sekä algoritmien listaukseen että etusivulle. Kelle tahansa satunnaiselle käyttäjälle on näkyvissä koko tietokannan sisältö. Analyysien ja esimerkki-implementaatioiden luomista ja muokkaamista varten tarvitaan kuitenkin sisäänkirjautuminen omilla käyttäjätunnuksilla. Jos näitä ei ole luotu, täytyy uuden käyttäjän rekisteröityä järjestelmään omalla tunnuksella ja salasanalla.

Rekisteröityminen käyttäjäksi

Uuden tunnuksen luomisen voi aloittaa painamalla etusivulla **Register**-painiketta. Tämän jälkeen aukeaa rekisteröitymissivu, jossa voi suorittaa rekisteröinnin. Rekisteröinti kirjaa samalla automaattisesti myös uuden käyttäjän sisään järjestelmään.

Kirjautuminen

Kirjautuminen onnistuu painamalla aloitussivulla **Login**-painiketta ja antamalla omat käyttäjätunnuksensa esiin ilmestyvälle kirjautumis-sivulle.

Ylläpitäjän tunnukset:

Uusien algoritmien lisäämiseen ja vanhojen poistamiseen on oikeudet vain tietokannan ylläpitäjällä. Ylläpitäjän käyttöoikeudet sovellukseen:

- **Käyttäjänimi:** Yyni Yllapitaja
- **Salasana:** pwsedwrs223**

Uloskirjautuminen:

Sisäänkirjautunut käyttäjä voi suorittaa uloskirjautumisen navigointipaneelin **"Log out"**-painikkeesta. Sessio myös vanhentuu myös aikarajoituksen täytyttyä, mikäli käyttäjä on pitänyt taukoa palvelun käytössä yli 20 minuutin ajan.

6. Järjestelmän yleisrakenne

Tietokantasovellus on suunniteltu MVC-malliin yhteensopivaksi. Juurihakemistosta löytyy kansio **app**, joka sisältää eritellysti sekä kolme alakansiota:

- **model:** sisältää sovelluksen toimintalogiikan, joka on toteutettu PHP:llä. Model-luokat ovat vastuussa tietokantayhteyksien ottamisesta ja olioiden hakemisesta. Nimentä **olionimi.php**-tyylillä

- **view**: sisältää sovelluksen näkymät, toteutettu html-sivuina. Alakansiot eritelty olioiden nimien mukaan. Tiedostojen nimentä **index.html**, **show_olionimi.html**-tyyliin.
- **controller**: sisältää sovelluksen kontrolleriluokat, nimentä **olionimi_controller.php**-tyyliin

Lisäksi ohjelman käyttämät apukirjastot on sijoitettu hakemistoon **lib**, ja asetukset löytyvät kansioista **config**. Dokumentaatio löytyy kansioista **doc**.

7. Käyttöliittymä ja järjestelmän komponentit

Sovelluksen ylälaudassa on navigointipalkki, josta pääsee oikaisemaan algoritmitietokantaan, ja etusivulle. Lisäksi kirjautunut käyttäjä pääsee kirjautumaan navigointipalkin avulla ulos. Ohessa kaavio sovelluksen näkymistä ja niiden kontrollereista:

