

Dokumentaatio: Algoritmitietokanta

Ville Saarinen
Kesä 2017, Tietokantasovellus-harjoitustyö
Helsingin yliopisto

Contents

1. Johdanto	2
2. Käyttötapaukset.....	2
Käyttäjäryhmät.....	2
Käyttötapauskuvaukset.....	3

1. Johdanto

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.

2. Käyttötapaukset

Käyttäjäryhmät

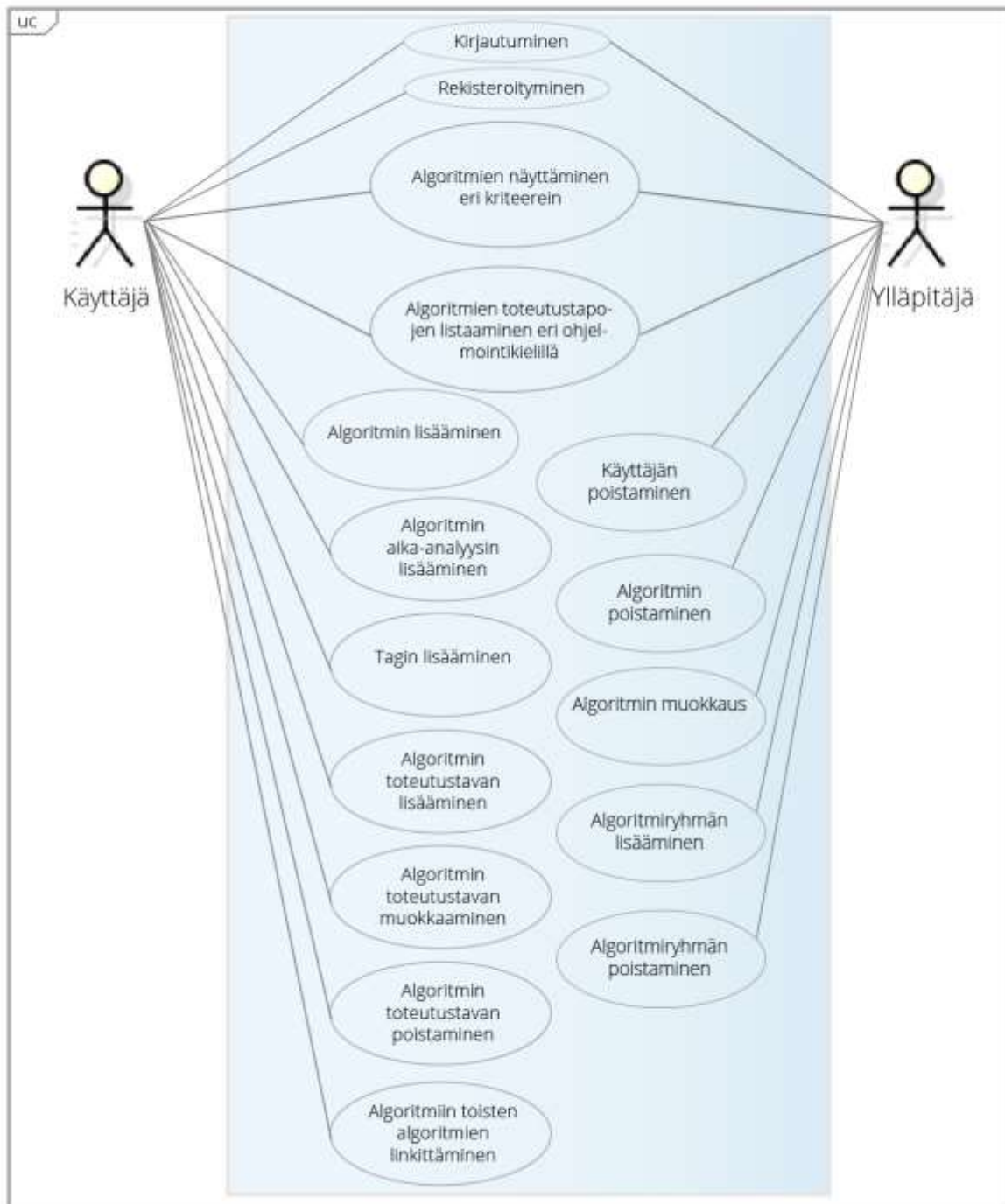
Käyttäjä

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 poistaa 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