**Rapport 1**

**Jonathan Meyer, Sergio Fenoll,**

**Dawid Miroyan, Cedric De Schepper**

**14 maart 2018**

## 1. Inleiding

### Geïmplementeerde vereisten

* Gebruiker kan een account aanmaken
* Gebruiker kan inloggen en haar gegevens raadplegen en wijzigen
* Administrator kan alle gebruikers inspecteren
* Administrator kan gebruikers op (non-)actief zetten en verwijderen
* Gebruiker kan ruwe data opladen vanuit een CSV-bestand
* Gebruiker kan meerdere CSV-bestanden opladen a.d.h.v. een ZIP-bestand
* Gebruiker kan ruwe data opladen vanuit een database dump in ‘standaard’ SQL
* Gebruiker kan metadata toevoegen
* Gebruiker kan enkel haar/zijn eigen data opvragen

### 

### 

### 

### 

### Software Stack

* *VCS*
  + *Git*
  + *Github (publieke repository)*
  + *Travis*
* *Hosting*
  + *‘Server’ bij Sergio thuis*
    - *‘Production’ branch*
    - *‘Development’ branch*
  + *NGINX*
  + *Gunicorn*
* *Frameworks/libraries*
  + *Flask*
  + *Flask-login*
  + *Jinja2*
  + *psycopg2(-binary)*

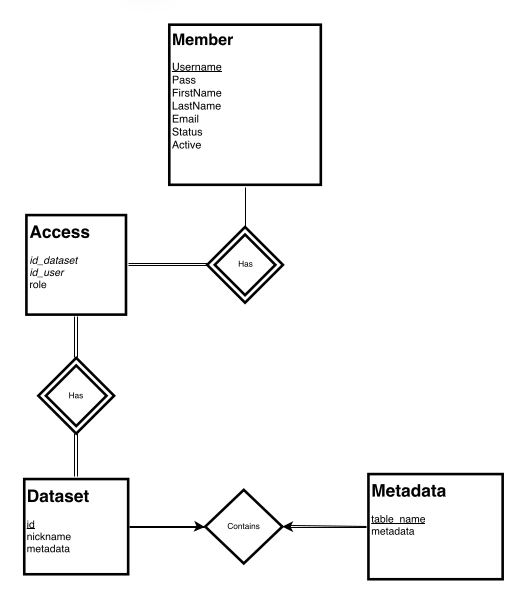
### Specialisatie

Aangezien we als team weinig individuele ervaring hebben met de technologieën, hebben we besloten om geen zeer duidelijk onderscheid te maken in wie wat voor taken onderneemt.   
Momenteel hebben we die werkwijze dus aangenomen, waardoor elk teamlid zowel aan de front- als backend heeft gewerkt.   
Sergio heeft echter voornamelijk gezorgd voor de frontend (of toch, de uiteindelijke), terwijl Jonathan en Cedric zich meer bezig hebben gehouden met backend. Dawid heeft voornamelijk unit tests voorzien en aan debugging voor de bestaande code gedaan.

De taakverdeling regelen we via Github issues, zodat we niet in elkaars vaarwater komen.  
Exacte details zijn dan ook daar beschikbaar (zie link onderaan).

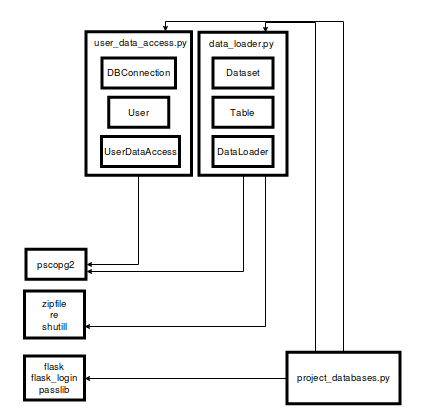
## 2. Design

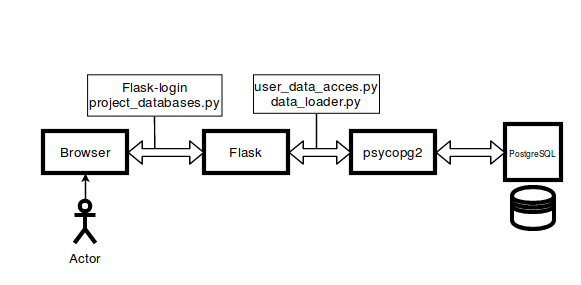
### Databank ontwerp

*  
Figuur: ER-diagram van de huidige database*

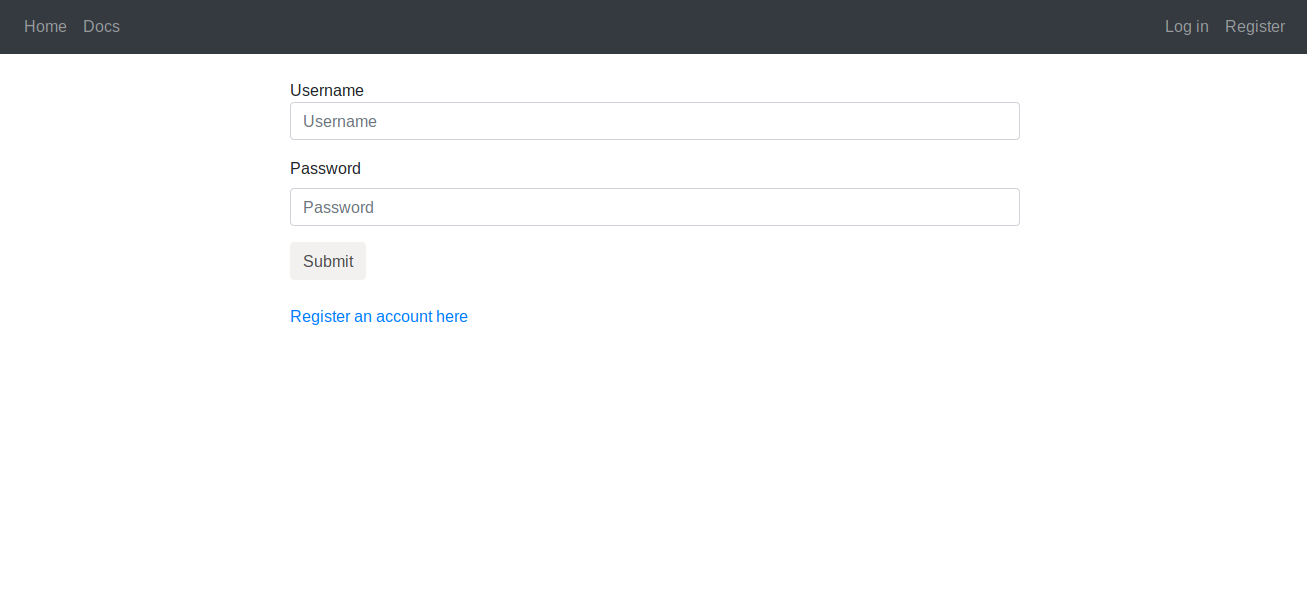
Ons huidig database model laat toe gebruikers toe te voegen, datasets bij te houden, metadata over datasets op te slaan en te beheren welke gebruikers toegang hebben tot welke datasets. De bijbehorende entiteiten en relaties zijn hierboven weergegeven.   
Verder kunnen gebruikers data uploaden en opslaan in de database. De datasets komen overeen met schema’s in de database, en data van gebruikers wordt als nieuwe tabellen aan de bijbehorende schema’s toegevoegd. Per schema wordt ook een tabel bijgehouden waarin de metadata voor de data van de gebruiker wordt opgeslagen.

### Programma ontwerp

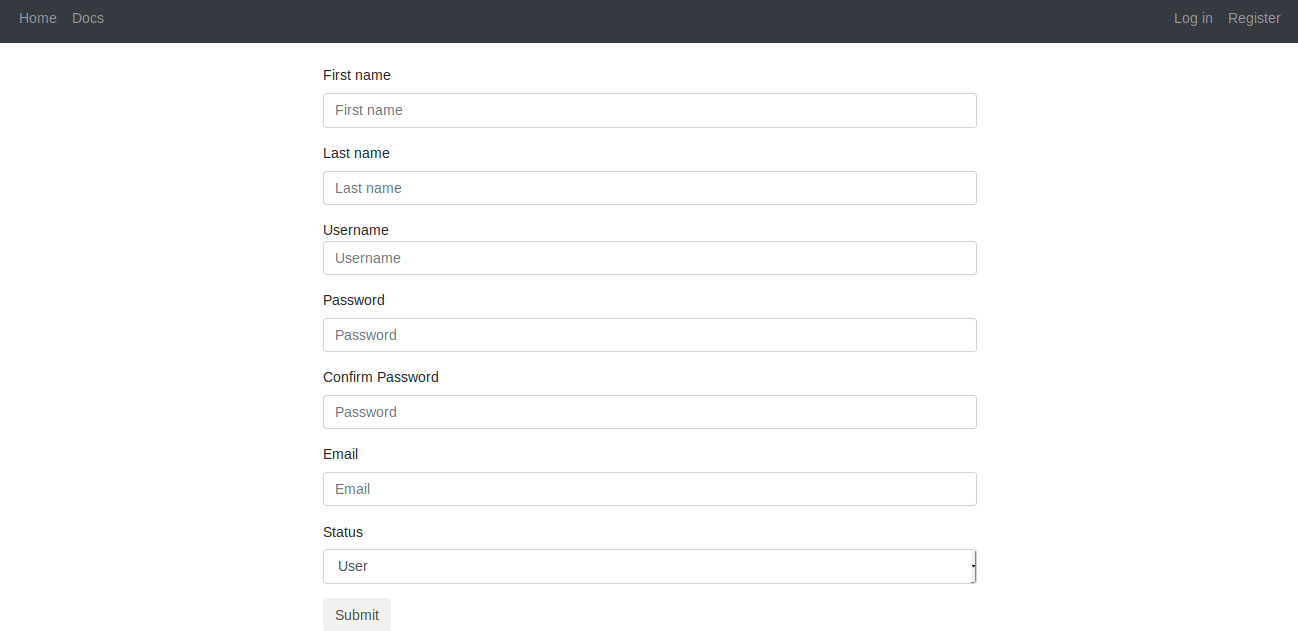
**

**

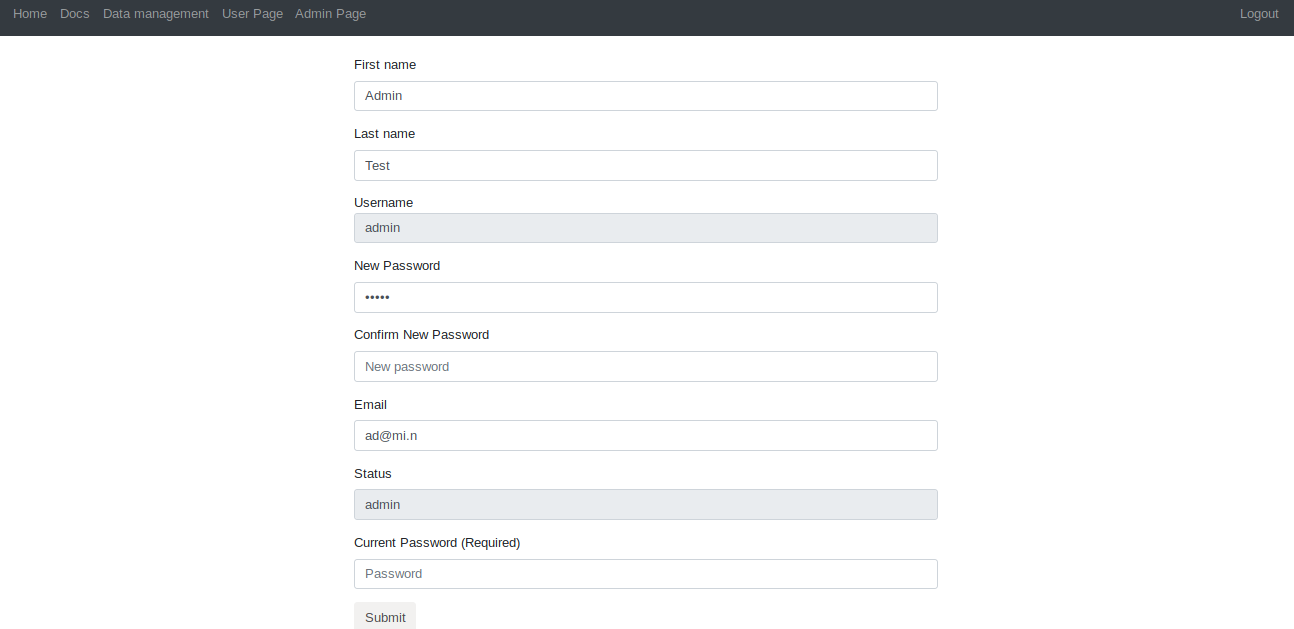
## 3. Manual

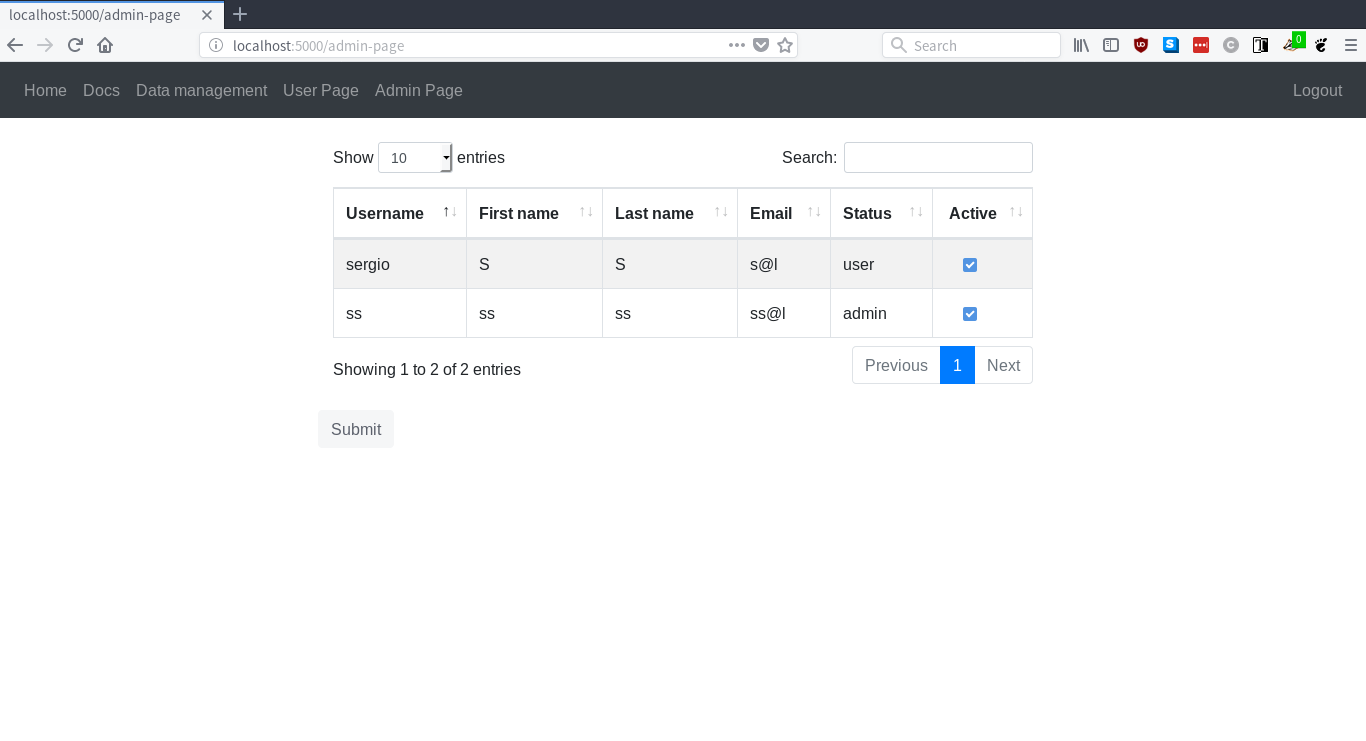
**

Op deze pagina kunnen gebruikers zich aanmelden. Indien het account gedeactiveerd is, of de logingegevens fout zijn, wordt de gebruiker op de hoogte gesteld.   
Er is ook een doorverwijzing naar de pagina waar nieuwe gebruikers zich kunnen registreren.

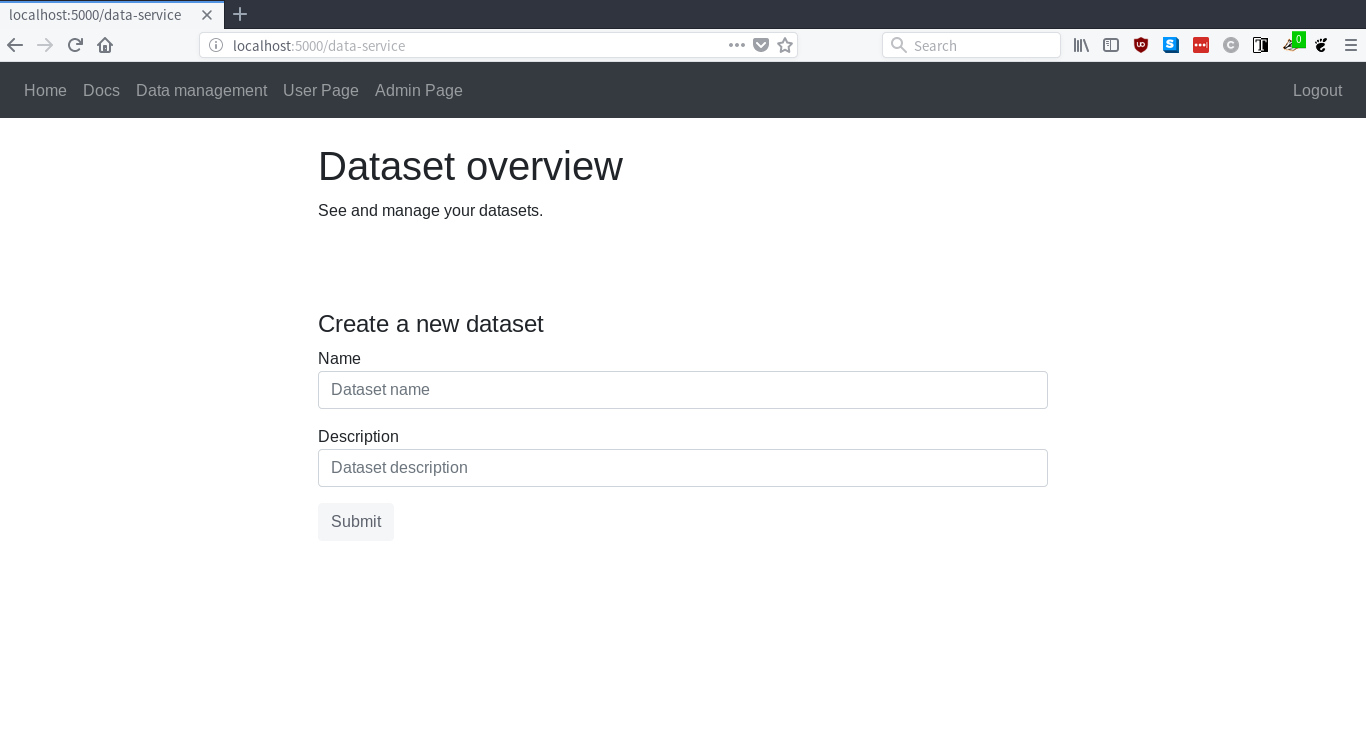


Op deze pagina kunnen gebruikers een nieuw account op de service aanmaken. Momenteel is het bij registratie mogelijk om de status van het account (admin of user) te kiezen.

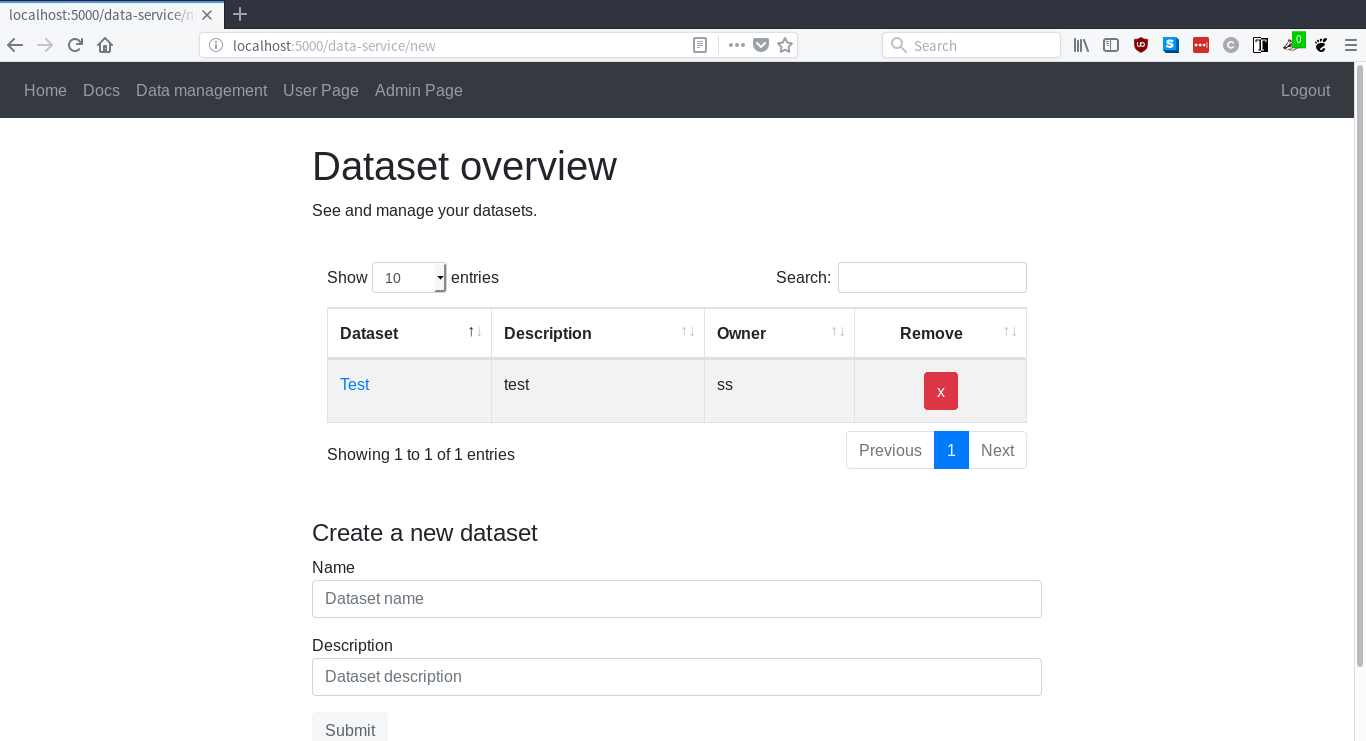
Hier kunnen gebruikers hun gegevens aanpassen.



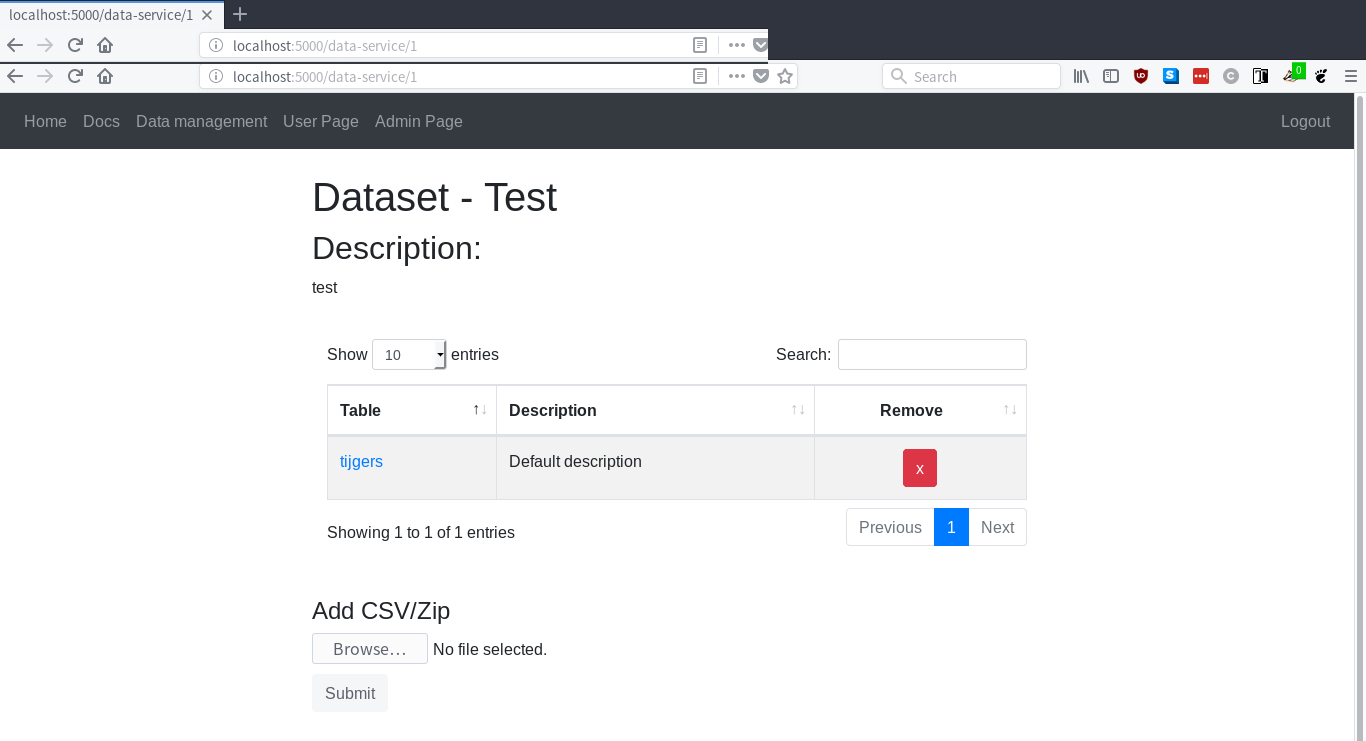
Hier kunnen admin gebruikers de active state van alle users aanpassen,



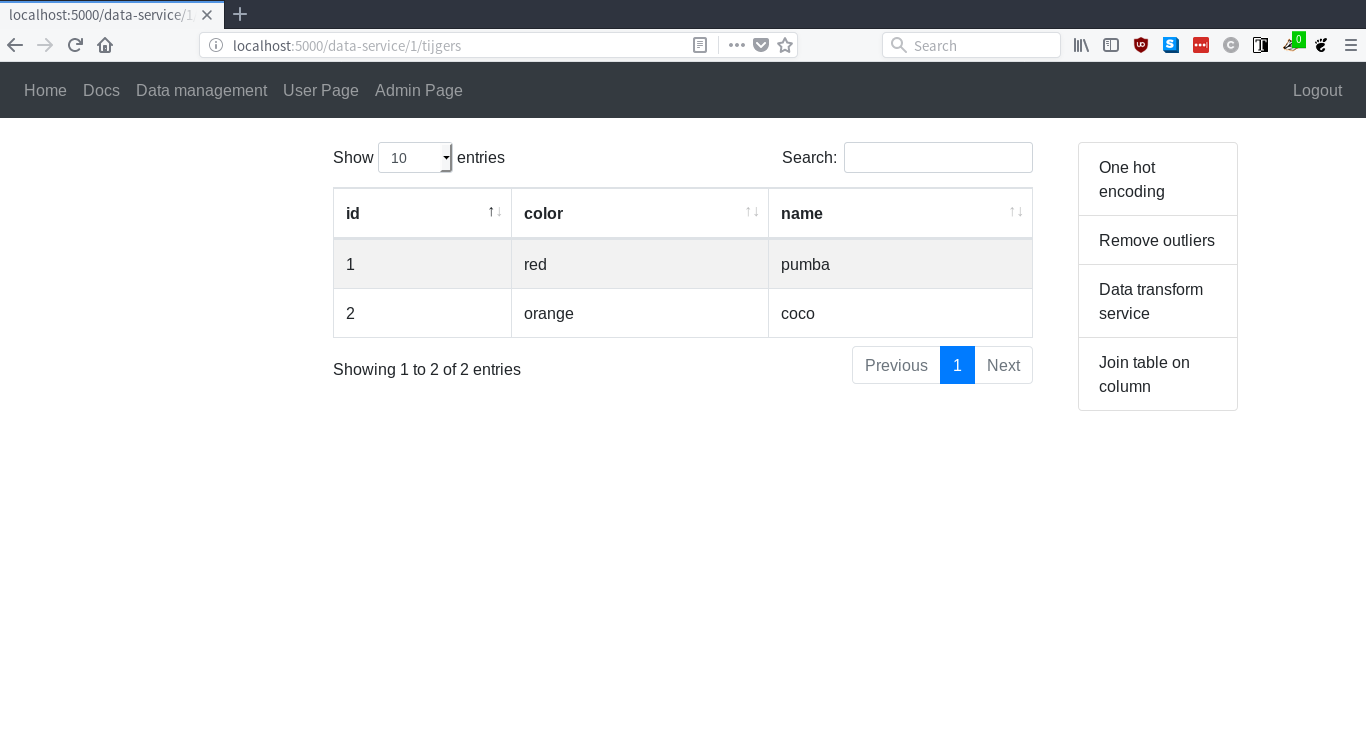
Hier kunnen gebruikers hun datasets zien en nieuwe datasets aanmaken.



Gebruikers kunnen hun datasets bekijken en verwijderen.



Hier zien gebruikers de data die behoort tot een gegeven dataset. Gebruikers kunnen ook nieuwe files uploaden.



Hier zien gebruikers de werkelijke data. Rechts kan men ook de data transform services zien.

## 4. Status

Voor meer details over waar aan gewerkt is, door wie en wanneer verwijzen we naar de Github  
(<https://github.com/sergiofenoll/project-databases>)

*Legende: done = ✔️, in progress = ❗, unfinished = ❌*

***User Service:***

Alles gerelateerd aan het onderdeel ‘user service’ van de opdracht. De implementatie hiervan is grotendeels volbracht.

* ***User login (✔️)***
* ***User registration (✔️)***
* ***User sessions (✔️)***
* ***Change user information (✔️)***
* ***Restricted user access (✔️)***
* ***Admin has user management privileges (✔️)***

***App layout:***

De layout van de applicatie, d.i. ‘Wat de gebruiker ziet’. Dit is een milestone die gedurende het hele project geldig zal blijven, aangezien we onderdelen blijven toevoegen en reeds bestaande onderdelen mogelijks blijven verbeteren.

* ***General app layout (❗)***
* ***Mockups for planned features (HTML) (❌)***
  + ***Data cleaning***
  + ***Data view***
* ***Table view (❗)***

***Data-import Service:***

Alles gerelateerd aan het importeren van data. De gebruiker moet in staat zijn data te uploaden naar de server. De backend hiervoor is vrijwel afgewerkt.

* ***Designing schemas for the raw data and ER-diagrams (✔️)***
* ***Loading table from CSV / ZIP (✔️)***
* ***Loading table from SQL dump (****✔️****)***
* ***Adding metadata (✔️)***
* ***Join uploaded raw SQL on columns (❌)***

***Visual polish:***

Dit gaat om het verbeteren van de weergave van de applicatie (dit slaat op alles wat de gebruiker te zien krijgt). Het voornaamste doel is hier gebruiksvriendelijkheid.

* ***Landing page (✔️)***
* ***Login page (✔️)***
* ***Register page (✔️)***

***Misc:***

Andere taken die belangrijk zijn voor het project, maar niet onmiddelijk tot de bovenstaande categorieën behoren.

* ***Presentation (❌)***
* ***Rapports (❗)***
* ***Bugsquashing (❗)***
* ***Cleaning existing code (❗)***