Rapport 1

Filip Smets
Nisse Strauven
Jens Bijtebier
Thanh Danh Le
David Danssaert

17 maart 2016

Inhoudsopgave

1 Status

2 Design

2.1 Keuzes

2.1.1 Dependencies

Clientside

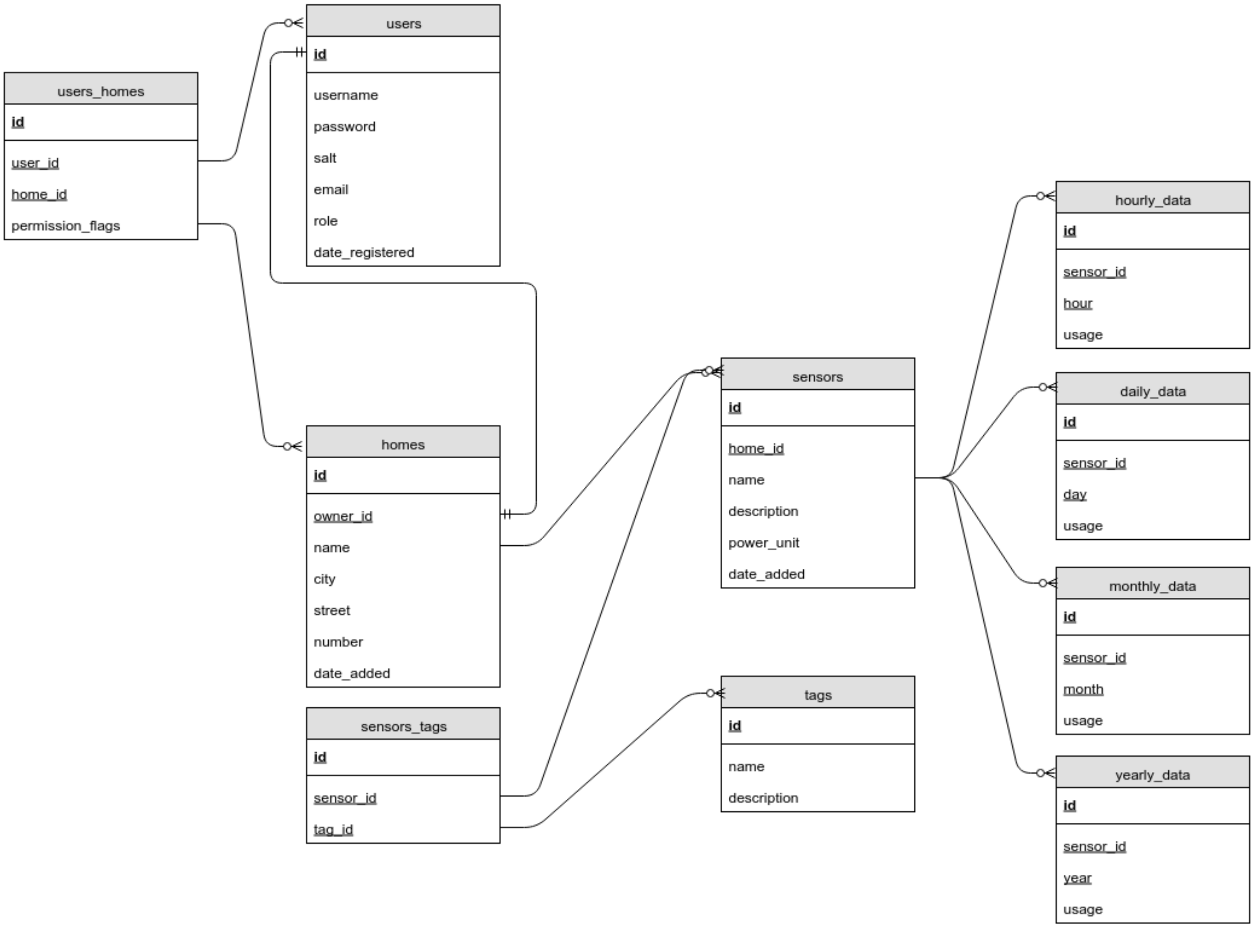
- node.js
- npm
- ember-cli (via npm)
- bower (via npm)
- ember-material-design? (via ember, not yet included)

Serverside

- python 3
- django (via pip)
- djangorestframework (via pip)
- markdown (via pip)
- django-filter (via pip)

2.2 Database Schema

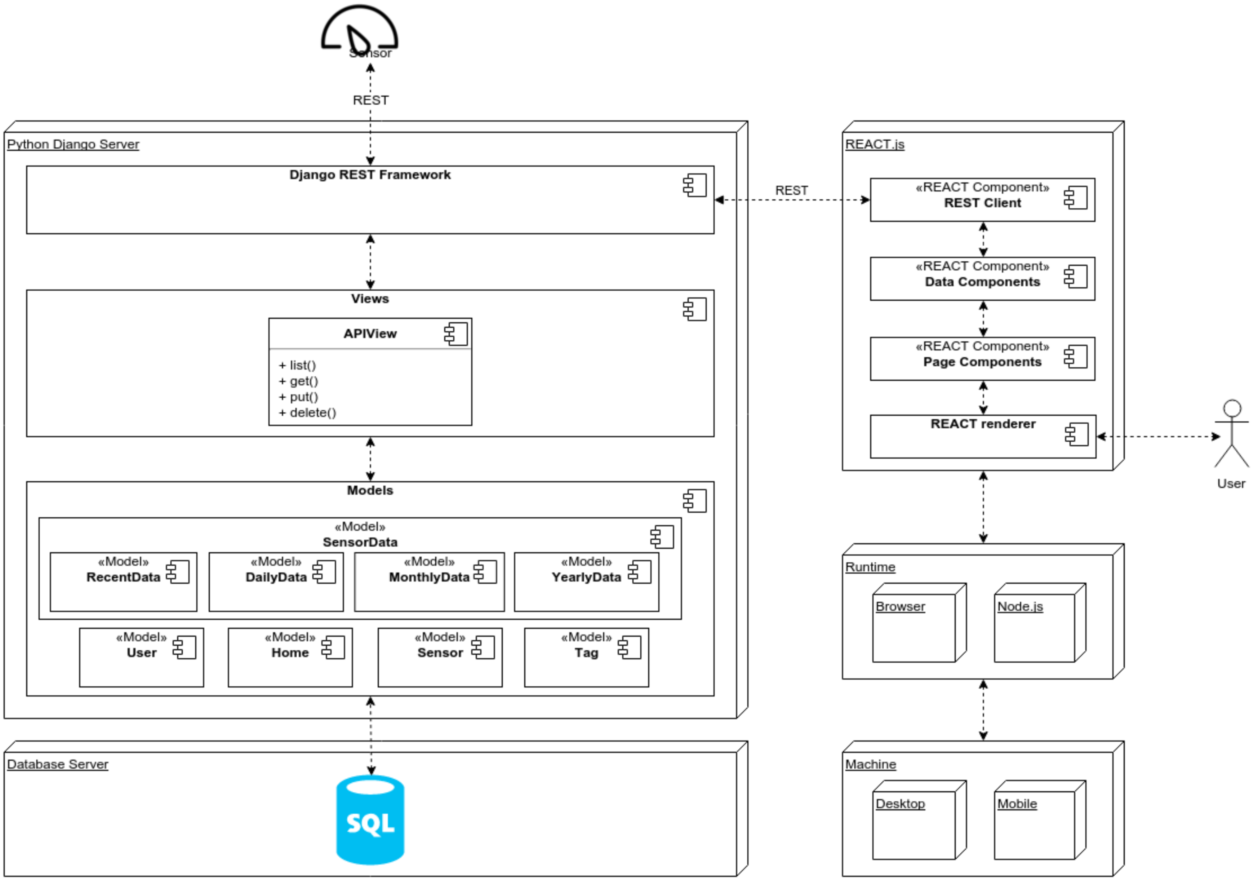
Zie **figuur 1**.



Figuur 1: Het 'Entity Relation Diagram'.

2.3 UML Schema

Zie **figuur 2**.



Figuur 2: Deployment diagram voor server (links) en client (rechts).

3 Product

3.1 Basisvereisten

3.2 Extra functionaliteit

4 Planning

5 Appendix

5.1 SQL Queries

5.1.1 Households

```
/* list all households */
SELECT homes.id, homes.owner_id, homes.name, homes.country, homes.
   city, homes.zipcode, homes.street, homes.house_number, homes.
   date_added
```

```
FROM homes;

/* retrieve household by home ID */
SELECT homes.id, homes.owner_id, homes.name, homes.country, homes.
   city, homes.zipcode, homes.street, homes.house_number, homes.
   date_added
FROM homes
WHERE homes.id = $ID;

/* retrieve household by user ID */
SELECT homes.id, homes.owner_id, homes.name, homes.country, homes.
   city, homes.zipcode, homes.street, homes.house_number, homes.
   date_added
FROM homes
INNER JOIN users_homes
ON (homes.id = users_homes.home_id)
WHERE users_homes.user_id = $ID;
```

5.1.2 Sensors

```
/* list all sensors */
SELECT sensors.id, sensors.home_id, sensors.name, sensors.
   description, sensors.power_unit, sensors.date_created
FROM sensors;

/* retrieve sensor by id */
SELECT sensors.id, sensors.home_id, sensors.name, sensors.
   description, sensors.power_unit, sensors.date_created
FROM sensors
WHERE sensors.id = $ID;

/* retreive sensors by tag id */
SELECT sensors.id, sensors.home_id, sensors.name, sensors.
   description, sensors.power_unit, sensors.date_created
FROM sensors
INNER JOIN sensors_tags
ON (sensors.id = sensors_tags.sensor_id)
WHERE sensors_tags.tag_id = $ID;
```

5.1.3 Sensor Tags

```
/* list all tags */
SELECT tags.id, tags.name, tags.description
FROM tags;

/* retrieve tag by id */
SELECT tags.id, tags.name, tags.description
```

```
FROM tags
WHERE tags.id = $ID;

/* retrieve tags for a given sensor (by id) */
SELECT tags.id, tags.name, tags.description
FROM tags
INNER JOIN sensors_tags
ON (tags.id = sensors_tags.tag_id)
WHERE sensors_tags.sensor_id = $ID;
```

5.1.4 Sensor Data

```
/* list the sum total usage of all sensors for each minute of today
   for a given user */
SELECT recent_data.timestamp, sensors.home_id AS home_id, homes.name
    AS home_name, SUM(recent_data.usage) AS usage
FROM recent_data
INNER JOIN sensors
ON (recent_data.sensor_id = sensors.id)
INNER JOIN homes ON (sensors.home_id = homes.id)
WHERE (homes.owner_id = $ID AND recent_data.timestamp >= DATE_SUB(
   NOW(), INTERVAL 1 DAY) AND recent_data.timestamp < NOW() )
GROUP BY recent_data.timestamp, sensors.home_id, homes.name
ORDER BY home_id ASC, recent_data.timestamp ASC;

/* Aggregate all minutely data from today into one record for all
   sensors */
INSERT INTO daily_data (sensor_id, timestamp, usage, n_measurements)
SELECT recent_data.sensor_id, CONCAT(DATE(NOW()), ’␣00:00:00’) AS
   new_date, AVG(recent_data.usage), SUM(recent_data.n_measurements)
FROM recent_data
WHERE ( recent_data.timestamp >= new_date AND recent_data.timestamp
   < DATE_ADD(new_date, INTERVAL 1 DAY) )
GROUP BY recent_data.sensor_id
```