

# Programmeerproject Databases: SmartHome

**Opmerking: Er is geen tweede zittijd voor dit vak!**

## *Doelstellingen*

- een uitgebreid software project tot een goed einde brengen
- de theorie van het vak Databases in de praktijk omzetten
- in groep werken, onafhankelijk van de begeleiding<sup>1</sup>
- creatief denken
- een nauwkeurige planning maken
- duidelijke verslagen schrijven

## *SmartHome?*

Het project omvat de ontwikkeling van een web applicatie waarbij gebruikers hun thuis-elektriciteitsverbruik kunnen **monitoren**, en desgewenst, **optimaliseren**.

De applicatie laat toe om de **ruwe verbruiksgegevens** van 1 of meerdere **sensoren** (totaal-elektriciteitsverbruik, verbruik van 1 of meerdere toestellen) automatisch te **uploaden** naar de *centrale database*, die aan de basis ligt van jouw applicatie.

De applicatie **bewaart** en **aggregeert alle informatie** op een efficiënte manier, en je applicatie beheert hierbij de historiek van het verbruik **geaggregeerd per dag, week, maand en jaar**.

Gebruikers kunnen, na het **inloggen**, hun gegevens raadplegen, en bij eventuele veranderingen in een locatie (bijv. overschakeling naar led verlichting, of toestellen uitschakelen ipv. sluimerstand, nieuwe sensoren), controleren wat het verschil is. Hierbij worden ze ondersteund door een grafische interface, met meerdere grafieken.

Daarnaast kunnen SmartHome gebruikers, gegevens delen, en tips uitwisselen, via het geïntegreerde **sociaal platform**.

Daarnaast is er ook een **organisatie modus**, voor de personeelsleden van SmartHome Inc (fictief bedrijf), die toelaat om het algemeen **verbruik van alle gebruikers** te aggregeren. Deze informatie kan dan bijvoorbeeld gedeeld worden met elektriciteit-producenten, zodat deze voor een slimmer elektriciteitsnetwerk kunnen zorgen dat rekening houdt met de vraag en aanbod van elektriciteit op een bepaald moment.

---

<sup>1</sup> Dit betekent natuurlijk niet dat er geen vragen gesteld kunnen worden.

## ***Basis vereisten***

Elke gebruiker, heeft 1 of meerdere locaties. Op elke locatie bevinden zich 1 of meerdere sensoren. Een sensor kan hierbij gekoppeld zijn aan de algemene elektriciteitsmeter, maar even goed aan een specifiek toestel.

De sensor-upload service zorgt dat elke klant ruwe sensor gegevens kan uploaden:

- Je implementeert een REST interface [1] waarmee je, over het internet, de ruwe sensor gegevens doorstuurt.
- Bij het versturen van gegevens, stuur je minimaal de gebruikers code, locatie code en sensor code door, alsook het tijdstip van de meting, en de decimale waarde van de meting.
- Je service wordt vaak aangeroepen, en moet dus efficiënt geïmplementeerd zijn. Hou rekening mee dat sensoren zeer frequent een meting doorsturen. Denk dus goed na op welke manier je gegevens opslaat in de onderliggende database.
- We voorzien hierbij een python programma dat je kan gebruiken (en aanpassen) voor de simulatie van een reeks van sensoren.

De data-management service zorgt dat enkel geaggregeerde historische gegevens worden bijgehouden:

- Na elke dag worden de ruwe sensor gegevens opgekuist, outliers (extreme waarde in interkwantiel 1 of 4) verwijderd, en enkel de uur/dag/week en maand gemiddeldes bijgehouden. Je voorziet, afzonderlijke tabellen, zodat elke eindgebruiker, snel zijn gegevens kan opzoeken per dag/week of maand, en vergelijken met historische gegevens.
- Hou er rekening met dat gebruikers geïnteresseerd kunnen zijn in details van de dag of week, alsook in historische gegevens over maanden en jaren.

Met de front-end web-applicatie, kan elke gebruiker:

- Inloggen
- De sensorgegevens zelf benoemen
  - Voor elke sensor een titel, beschrijving, eenheid (e.g. kwh) en opmerkingen geven.
  - De sensoren taggen, met meer algemene concepten (zoals verwarming, koelkast, droogkast, computer, verlichting).
  - Per sensor, verandering kunnen aangeven op een bepaald tijdstip (zoals nieuwe verwarmingsketel).
  - Er is ook een optie, waarmee de gebruiker zijn kostprijs per eenheid kan invoeren.
- Historische, opgekuiste en geaggregeerde gegevens bekijken, aan de hand van grafieken
  - filteren op periode
  - sorteren op verbruik of kostprijs
  - verschil maken met een vorige periode

- Dit moet ook mogelijk zijn, voor een selectie/groep van sensoren, bijv. bepaald door de tags.
- Sociale aspecten:
  - vrienden (of groepen) maken
  - messages schrijven, zichtbaar op de wall van elke vriend
  - 'verbruik' messages maken, waarbij een grafiek (zie vorige) gedeeld kan worden. (bijv. na installatie nieuwe condensatie ketel, betaal ik 100 euro minder per maand in de wintermaanden!)

Andere eigenschappen:

- De interface (inclusief de vragen) vertalen naar een andere taal. Standaard is Engels en Nederlands beschikbaar.
- De applicatie moet het toelaten dat verschillende gebruikers op het zelfde moment hun gegevens kunnen bekijken (multi-user).

In de organisatie modus kunnen personeelsleden van SmartHome Inc

- Inloggen.
- Een groep gebruikers selecteren (straat, postcode, ...) en hiervan de sensor gegevens aggregeren.
- Een rapport genereren dat aan de producenten kan worden meegedeeld.

## ***Extra vereisten***

Deze basisvereisten tellen voor 10 van de 20 punten, de rest krijgen jullie op basis van extra functionaliteit. Het wordt sterk aangeraden dat jullie zelf extra functionaliteit bedenken en implementeren. Hierbij is het vooral belangrijk dat het nuttig en interessant voor de gebruiker is. Tijdens jullie presentaties krijgen jullie een kans om ons te overtuigen dat jullie ideeën inderdaad origineel en nuttig zijn.

Mogelijke voorbeelden zijn:

- Een 'versnelde' simulatie te doen, waarbij de gebruiker op enkele seconden, een animatie krijgt van het verbruik van enkele uren of weken.
- Te voorspellen wat het toekomstige verbruik is, aan de hand van regressie of andere methode.
- Aanbevelingen doen, om het verbruik te verlagen, alsook te voorspellen wat de besparing is.
- Social media integratie (Facebook, Twitter...), waarbij je ook je verbruik (via een grafiek bijv.) kan delen via deze tools.

Behalve de functionaliteit, worden jullie ook op de volgende aspecten beoordeeld:

- kwaliteit van de verslaggeving en planning
- teamwork

- kwaliteit van de presentaties
- kwaliteit van de programma-code en het ontwerp

## ***Uitvoering van het Project***

Dit project is bedoeld voor groepen van vijf studenten. De groepen worden door de studenten ***zelf voorgesteld*** en aan ons (via [len.feremans@uantwerpen.be](mailto:len.feremans@uantwerpen.be)) gemeld, ten laatste 15 februari 2016. Studenten die na deze deadline nog steeds geen groep hebben zullen dan door ons in groepen ingedeeld worden (hierbij bestaat dus de mogelijkheid dat de bestaande groepen een extra lid krijgen). De ***officiële groepslijsten zullen wij*** ten laatste 16 februari op Blackboard bekendmaken.

## ***Rapportering***

Er wordt een uitgebreide rapportering verwacht:

- Wekelijkse Verslagen
- 2 Tussentijdse Rapporten
- 1 Eindrapport

In de rapporten dient in detail te worden ingegaan op het design van het programma als geheel, alsook van de database (ER-diagrammen, vertaling naar het relationeel model) en, in het eindrapport, alle SQL queries (incl. CREATE TABLE statements). Hierbij dient ook duidelijk beschreven te worden welke extra mogelijkheden (bovenop de basisvereisten) geïmplementeerd zijn. Alle gemaakte keuzes moeten degelijk gemotiveerd worden. Ook de taakverdeling binnen jullie team moet duidelijk vermeld worden in de rapportering aangezien tijdens de presentaties hier op ingegaan zal worden. Zorg ervoor dat iedereen aan het woord komt en zijn eigen (geïmplementeerde) projectonderdeel presenteert. Let op: het maken van presentaties/rapporten/verslagen is geen taak op zich maar dient door het gehele team verzorgd te worden. Elk teamlid moet dus bij de werkelijke implementatie betrokken zijn.

De tussentijdse rapporten moeten op **17/03/2016** en **25/04/2016** ingediend worden. Voor deze rapporten stellen we jullie een template ter beschikking als document “Rapport Project Databases”. De elementen aanwezig in deze template moeten minimaal in jullie rapporten aanwezig zijn.

Verder verwachten we dat jullie minstens elke week een vergadering houden waarbij jullie de status en de planning van jullie project samen bespreken. Hiervan verwachten we telkens ook een verslag met aanwezigheden, genomen beslissingen met motivatie, opvolging geplande taken... We zullen deze verslagen gebruiken om eventuele problemen binnen de groepen tijdig op te merken, en eventueel ook als motivatie voor het geven van individuele punten. Als de taakverdeling eerlijk was, en iedereen in staat is zijn deel te presenteren en vragen te beantwoorden, krijgen alle leden van de groep dezelfde punten. Ook hier stellen we jullie een template ter beschikking als document “Verslag Project Databases”. De elementen aanwezig in deze template moeten minimaal in jullie verslagen aanwezig zijn.

Een computerlokaal is gereserveerd voor jullie op de volgende data:

G 026 10:30 – 16:00 op 11/2, 18/2, 25/2, 3/3, 10/3, 21/4, 28/4 en 12/5.

G 026 8:30-13:00 op 17/3

Tussen **10:30** en ten laatste **11:30** is Len Feremans aanwezig en kunnen vragen en problemen bij de uitvoering van je project besproken worden.

## ***Presentaties***

Tijdens het semester zullen er twee tussentijdse presentaties georganiseerd worden (**21/3** en **28/4**), gevolgd door een eindpresentatie op **19/5**. Tijdens elke presentatie moet er een werkende demo voorzien zijn. Een demo mag je op je eigen laptop doen. We raden sterk aan dat jullie op voorhand alles grondig controleren (internetverbinding, connectie met projector, etc.) opdat de demo vlekkeloos verloopt.

Voor de eerste tussentijds presentatie verwachten we dat alle belangrijke keuzes gemaakt zijn. Dit voornamelijk rond het design, de database en de taakverdeling. Voor de tweede tussentijdse presentatie worden jullie geëvalueerd op basis van een werkende demo van het product, dat voldoet aan de basisvereisten. Daarnaast verwachten we een voorstel van de geplande extra functionaliteit.

Tijdens jullie tussentijdse presentaties krijgen jullie feedback en worden jullie ook geïnformeerd over wat we verder verwachten en of er aanpassingen in jullie planning nodig zijn.

## ***Deadline***

De deadline voor het indienen van het project en eindrapport is **17/5**. Je moet de volledige database-dump en je code indienen, samen met een handleiding die aangeeft hoe het project geïnstalleerd en gebruikt moet worden (alles in een zipfile, indienen via email). Bij het indienen van het project, rapporten en verslagen moeten alle groepsleden in cc staan.

Tenslotte enkele voorbeelden van bestaande systemen:

<http://www.smappee.com/>

<https://www.bidgely.com>

<http://neur.io/>

## ***Aanbevelingen/vragen***

- *Er is een enorme diversiteit aan programmeertalen en bibliotheken voor het maken van een web applicatie, met een achterliggende databank. Het is belangrijk dat jullie in een eerste fase zelfstandig uitzoeken, welke opties er zijn, en vervolgens een keuze maken.*
- *Focus tijdens het initiële ontwerp, vooral op de basisvereisten, en pas dit ontwerp aan, nadat de basisvereisten geïmplementeerd zijn.*

- *Gezien de beperkte tijd, is het belangrijk geen tijd te verliezen met de implementatie van uitgebreide, of complexe deel-onderdelen. Het is verstandiger om kant-en-klare libraries en tools slim in te zetten. Voor visualisatie van gegevens over de tijd, meertalige applicaties, SQL updates op de databank, of het weergeven en aansturen van interactieve web pagina's zijn er voldoende mogelijkheden van geïntegreerde ondersteunende libraries en tools.*
- *Zorg dat de presentaties vlot gebracht worden en interessant zijn, met zowel een focus op het eindresultaat als de onderliggende technologie.*

## Referenties

[1] [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)