

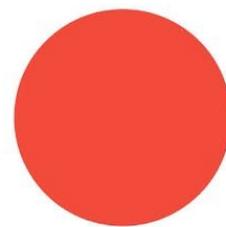
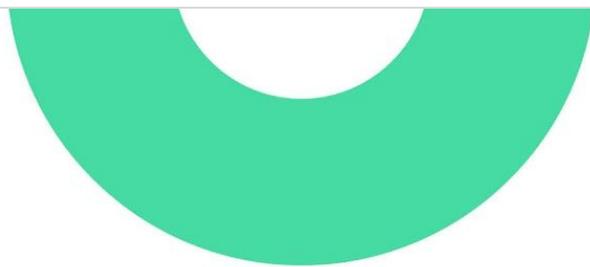


Access Flanders

At open summer of code

2 — 26 July 2018

Output of the CEF project OASIS.team



**open
summer
of code**

INTRODUCTION

AccessFlanders is a **web application** that uses the data gathered by the [Smart Flanders team](#). This data is displayed on **an interactive map**. The user can search for public buildings and the web app will show the user extra information about the **accessibility of that building**: is there an elevator, what's the width of the door,...

All of this information makes it easier for people with disabilities or care providers to plan their trips beforehand. The first filters will be implemented during oSoc18, more filters will be added in the future.

CONTEXT

Target group

Our target group is divided in two groups. The first group are the people with a disability and more specifically people in wheelchairs. The second group are the people who take care of people with a disability, these can be friends, family, etc.

Our priority for this project was to make a user friendly web app to facilitate the experience of a user when he wants to get accessibility data of a public building. We did build an interactive map where the users can search for public buildings and get extra information about the accessibility of that building: is there an elevator, what's the width of the door, etc.

The only "real" competitor in the way we present it is onWheelsApp, this also shows a map with locations and their accessibility information. But as explained above there is no icon legend. In that way our app does it better as it is compact and clear. We also actively use Linked Open Data, via SmartFlanders, this means that if someone adds their building to the catalog our app automatically 'knows' it.

THE PRODUCT OR SERVICE

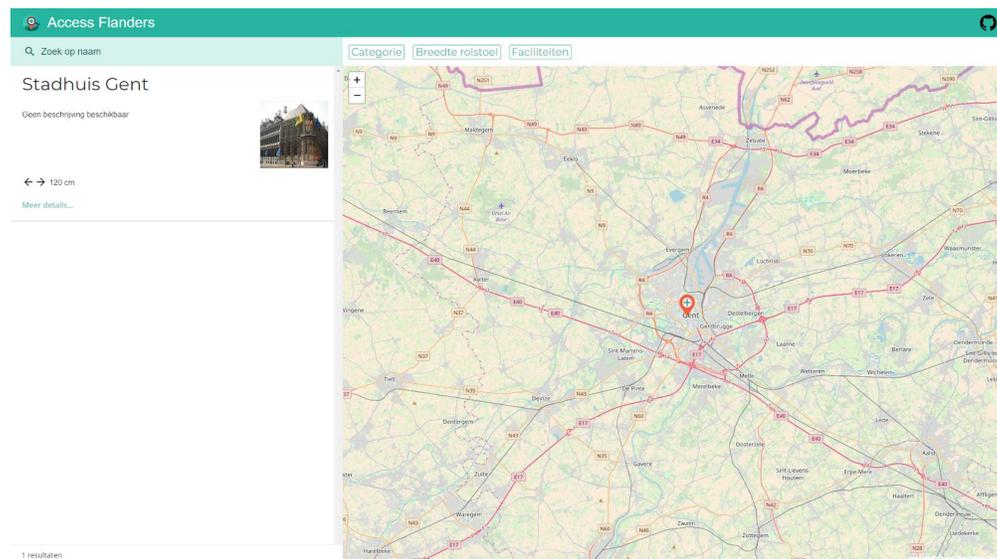
We built a web app that shows all the buildings that SmartFlanders provide on a map. The user can then search by name of the building and filter by wheelchair width to check if the user can pass the door.

Scope of the project, what did you build? Provide all useful links:

- <https://accessflanders.be/>
- Link to the landing page
- <https://github.com/oSoc18/OASIS>

Functionality

On first load of the web application it starts loading in the available linked open data from the SmartFlanders master catalog. This data will be shown to the user in the sidebar as either a building or just a public service (this happens if there is public service information available but linked to a building that is not available). If it is a building its location will be put on the map via a marker.



Stadhuis Gent

Description not available

Details

 The entrance: 120 cm

 The elevators: 110 cm

Services

[Return to search results](#)

Breedte rolstoel: 94cm

Faciliteiten



When clicking on 'More details' or the map marker you will see any information we have on a certain service or building. This is the name, description the available accessibility information and if there are any public services inside this building.

You can search through the results based on name via the search bar. This will dynamically update the list shown as well as the map markers. There is also a filter available to let the program know how wide your wheelchair is, this also updates the results / map.

CREATING THE PRODUCT

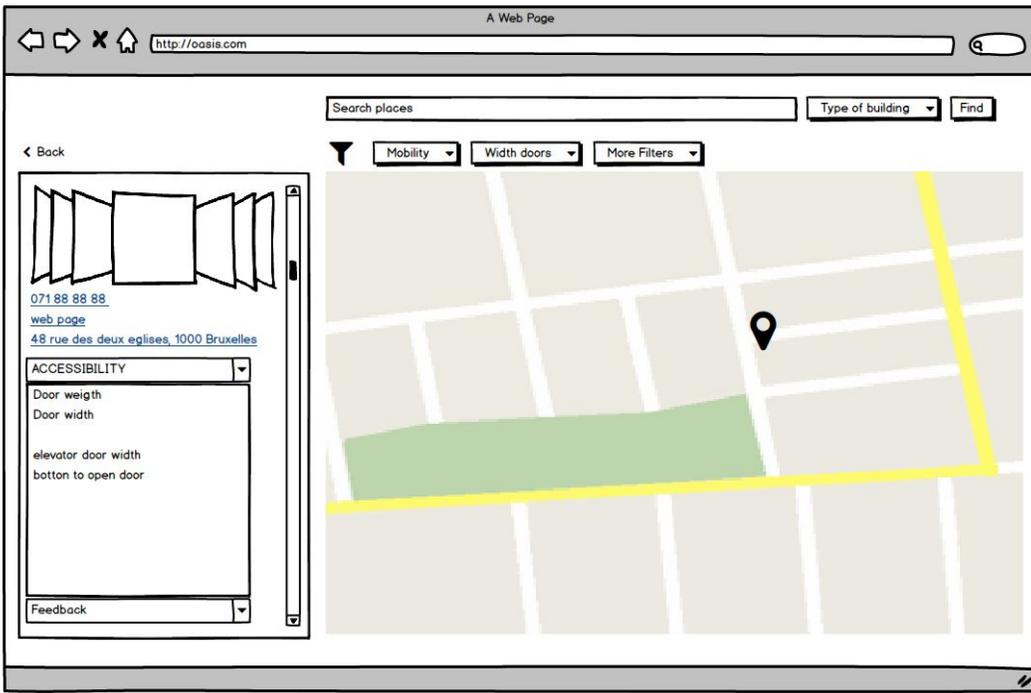
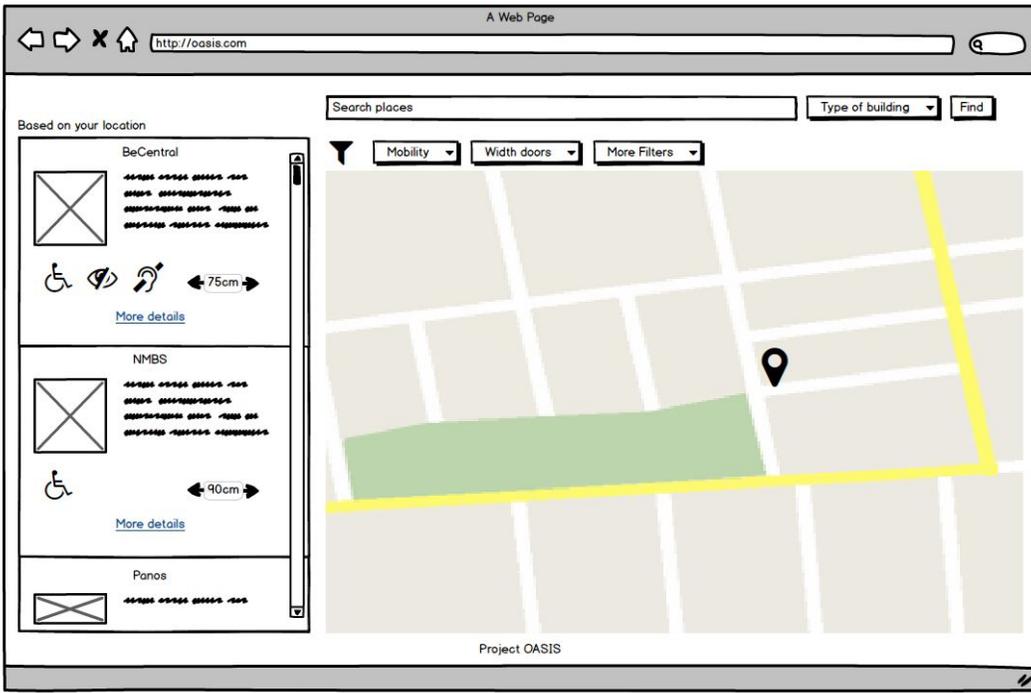
Branding

We created a new branding named "AccessFlanders" with its corresponding logo. All the elements on the logo have a link to what the application is about.



Sketches, wireframes & mockups

We created wireframes based upon first sketches that everyone of the team drew. We took elements that were returning in every sketch. A sidebar which contains all the results, a map more centrally on the screen that shows the locations of the buildings via a marker. As well as a search bar and some filters.



Frontend

The program uses the React framework with mobx as state manager and Ldfetch. From a technical point of view you have the App component which is the core component. This component renders 2 children, namely Sidebar & Map. It also loads in the Linked Open Data via Ldfetch. It 'maps' the information we get to a Building class (component) and pushes it into an array. This array gets stored in the store via mobx.

The choice for React was because we wanted to create a client side website and React has a huge amount of resources available. We chose mobx for the state management because we needed something stronger than just the internal state of React. As the state of the sidebar needed to be updated by different components (for example child, parent or sibling). Also for the time we had, redux was out of the question because of the somewhat steep learning curve.

Ldfetch or Linked Data Fetch is a HTTP client for RDF resources to fetch json-ld data from a URL.

For a detailed overview on how to get it working on your computer visit the github [readme page](#).

Feedback

User testing

We did make sketches with wireframes and showed them to coaches to have feedback and we had discussions with them. On the basis of the discussions we fine tuned the wireframes and based our web app from it.

Improvements

The application has been tested by our partner, the feedback was that the web app was good but that the design could have been better. We did the necessary and the application did improve design oriented.

THE FUTURE

Added value for the open source / open data community

We provide them with a website that reads in the SmartFlanders catalog, if the cities start working and providing more data, our code / web app can evolve to a more allround visualization platform of different types. It is focussing now on accessibility but it could focus on a broad amount of subjects if data is provided and people are willing to visualize that new data.

As an added value for the open data community we can kickstart our project to become a more central platform for visualizing linked open data ?

To be implemented

The product will be improved by adding extra filters to search more easily and enhance the user experience of the web app. Filters like category and type of facility would be the two first filters to be implemented in the future. Another feature that has to be implemented in the details page is to show the address of it.



Fixes & Bugs

Make a fix to not show the public service if there is no link to a building, currently we force hide it by *activating a filter* because we know these public services will be hidden by it if it doesn't contain a building.

Minify the javascript code on the server.

Future features and ideas

The product would be even better if there would be more filters, more data to show and a first try to find a solution for the chain of accessibility.

A nice added value would be a mobile version of the web app.

A nice project for a next open summer of code would be to either mix data (for example the openinghour of buildings and the accessibility data) and show the different type of data on the web app, or to create a template so that everybody can reuse the web application for his data.

Implement an URI query so people can link to our website with a search result and land straight on the information of a building.

TEAM MEMBERS



Brecht Van de Vyvere

Coach

brecht.vandevyvere1@gmail.com

@brechtvdv

[LinkedIn](#)

<https://github.com/brechtvdv>



Bert Marcelis

Coach & Front End Developer

todo:email

@BertMarcelis

[LinkedIn](#)

<https://github.com/Bertware>



Eveline Vlassenroot

Coach

eveline.vlassenroot@ugent.be

@EvelineVlas

[LinkedIn](#) <https://github.com/evelinevlas>

**Ilias El Mesaoudi**

Front end developer

elmesaoudi.ilias@hotmail.be

[LinkedIn](#)

<https://github.com/iliaselme>

**Brian Dendauw**

Front end developer

briandendauw@hotmail.be

[LinkedIn](#)

<https://github.com/DendauwBrian>



Lisa Hennecart
Front end developer

[LinkedIn](#)

<https://github.com/HennecartLisa>



Milena Vergara Santiago
Designer

milena.vergarasantiago@outlook.com

@MilenaVergaraS

[LinkedIn](#)

<https://github.com/MilenaVergaraS>