# smartNews Project Documentation

This Web-App is supposed to close the gap between the Facebook News Feed and the front page of a typical news site. By analysing the Facebook data of a user it is easily possible to customize a front page for each user. Information of each user like the current position, the hometown or Likes for specific themes can be utilized. In this way a customized front page for each user is possible. Directly in Facebook.

## Features

* Connecting a user to Facebook
* Getting Facebook User Data from their Social Graph API
* Getting news stories from The Guardian API
* Order this news stories with the help of the Facebook User Data in sections
* Build sections after the users location and home on country scale
* Build sections after Facebook Likes
* Nice UI and easy control over sections
* Storing user preferences in local storage
* App works in Facebook Canvas and as standalone website
* Objects used to communicate between modules

## High Level Architecture

The app is built in four modules: app. main, app.storage, app.content, app.social.

The module app.main handles the main logic of the app. App.main receives the facebook data and determines with its help which content to ask for. It also handles the UI of the app.

Reading and writing to the local storage is handled by the app.storage module. This module provides interfaces to write and read from the local storage and to determine if a user has visited before.

The Facebook connection is made with app.social. This module is responsible for connecting to Facebook and asking for Facebook data.

The app.content module calls the Guardian API for specific content. It also brings the content on the screen.

## Design Decisions

The overall design pattern of the app is a reminder on old newspapers. Everything is in black and white. Lines separate the articles and sections. The font also looks like a newspaper font. The design mocks in a way the classical newspapers. While it looks like a newspaper from the 1920s it has most modern technique behind it. The only colour inside the design are the red and green on-hoover colours of the up- and down-voting buttons. This emphasizes the buttons (the only real interaction objects, besides deciding what to read).

For the UI JQuery UI is used. Some fade-in and fade-outs smooth things. On start of the app a loading bar signals the loading of the app. After loading tool-tips explain the interaction possibilities. The process to grant Facebook data is sufficiently explained.

The app works fine with many different resolutions found on desktop computers. The app scales itself well with down to 800 pixel width.

## Technical Decisions

I decided to use two libraries inside the app. JQuery UI for UI issues and JQuery for AJAX calls. Both libraries make things easier to code. With JQuery UI animations, the loading bar and the tool tips are easy to implement. And JQuery helps with calling the APIs without running in troubles because of cross origin restrictions.

The smartNews app has a lot of custom code. So I decided to build the app after the module pattern. I only needed four modules but it helped me a lot to structure the code. It also helped with the data patterns. The app receives a lot of different data from the two APIs. The module pattern helped dealing with this different types of data.

Because most of the computing is done at the start of the app, the modules (and their methods) are mostly called in a strict order. Starting with connecting the user to Facebook and asking for the Facebook data. (If the user is not connected to Facebook a dialog to connect the user is started.) After that the main logic is launched. Here it is determined which content should be shown. The main logic uses the local storage to access user preferences and the previously shown sections. A custom algorithm is used to determine which sections to use. Each Facebook like of a user has a category (like sports, book, music and many more. I have around 200 categories in my list and they are still not all). These category occurrences are counted. I use two custom json files to find corresponding Facebook like-categories and Guardian sections. Now every guardian section has a count. If the user has visited before his preferences and last seen sections are counted in. The now highest counts get shown. (Sections with user location and hometown are not touched by the algorithm at the moment. If these information are accessible corresponding sections are always shown.)

The main logic also starts up the UI and event handler. Now the content module gets launched and calls the Guardian API for specific content. After receiving the content the module displays it in the four sections of the app. Three sections are simple listings of articles, while the fourth section is a ticker. The ticker logic is also in the jurisdiction of the content module.

The communication between the modules is done in objects. As written above I get a bunch of data from Facebook and also want very specific outcomes from the Guardian API. To get a satisfying outcome I decided to use objects to communicate. This makes the code much more complicated but I couldn’t achieve my goal without it.

I decided to develop the App for both Facebook canvas and as a standalone website. There is not that much difference between those two platforms, so I could easily do both. But I don’t have the permission to reload the site in the Facebook canvas. (It runs in an Iframe and crashes when tried.) For that reason I built an extra version (with just commenting out two lines of code) for Facebook Canvas.

## HTML5 Technologies

HTML5 specific tags are used. As HTML5 technology we didn’t cover in class I used the local storage. I also would have liked to also use the geolocation technology. But it doesn’t make that much sense for an application which only runs in a desktop browser. So I decided to use my time resources to improve the implementation of the local storage and the algorithm to decide which content to show.

## Above and beyond

I went above and beyond with choosing my APIs. While the Guardian API is quiet simple (but still very powerful) there is nothing like THE Facebook API. I used in fact two APIs for Facebook alone. One to connect the user to Facebook and get authorisation and the second to read the Facebook Graph API. Both Facebook APIs come with a lot of different options and at least the Graph API is very powerful and gives many opportunities. I definitely didn’t choose an easy way here.

I went also above and beyond with my code. In class we never send objects from one module to another and implementing building/reading those objects was quite challenging.

The app also uses a self-made algorithm and some hand produced json files.

And the app works in the Facebook Canvas and as a standalone website.

## External Code

As mentioned above I use JQuery (<http://jquery.com/>) and JQuery UI (<http://jqueryui.com/>) as external libraries. I use the versions hosted by the Google CDN (<https://developers.google.com/speed/libraries/>). I also use a font from Google fonts: Roboto Condensed.

I used the documentation for the Facebook APIs (<https://developers.facebook.com/docs>) and the Guardian API (<http://open-platform.theguardian.com/documentation/>). I didn’t use the sample code directly but there are for sure some similarities. The Guardian logo is provided by the Guardian and their terms of service require me to add the logo.

And I used code from class examples and also from stack overflow. The stack overflow code is documented. Here is the full list:

<http://stackoverflow.com/questions/4216648/facebook-pages-authoritative-list-of-categories> (There is no official list of all the Facebook Like categories. So I used (and enhanced) this list.)

<http://stackoverflow.com/questions/1069666/sorting-javascript-object-by-property-value>

<http://stackoverflow.com/questions/3357553/how-to-store-an-array-in-localstorage>

<http://stackoverflow.com/questions/326069/how-to-identify-if-a-webpage-is-being-loaded-inside-an-iframe-or-directly-into-t>

## Grade

I think I meet all five requirements. I have a sufficient HTML/CSS page design. I use JQuery UI for UI effects. My apps doesn’t require lots of control options but the ones I have work fine. The user always knows what is going on.

I use two web APIs. (Or three, if Facebook login and Graph API are counted separately.) With using the local storage I also meet the requirement of having a “out of class” HTML5 technology.

I think that my app does something useful. I also like the style of the app and think it has definitely portfolio quality. Also my programming goes beyond what we did in class. I didn’t use much code from ICE or examples and I use custom objects to send data from one module to another. I worked inside the coding standards and my code runs with no errors.

I also went above and beyond with my app, as described above. I would grade myself between 95 and 100%.