water-fountains.org | work package 3

Modern front-end

FEATURES

- Cleaner visual design
- Multi-lingual ready
- Rich URLs for better sharing and browser navigation
- Intuitive and responsive interface
- Display Wikidata & Wikimedia photos/content in a detailed view
    - Requires WP2 - Data consolidation
- new features (see below for details)
    - Filter fountains
        - By text search
        - By "noteworthiness" (fountains having a Wiki entry)
        - By water type
        - sort by distance to current location
    - Routing by foot/bike/car
    - Links to view fountain in Google street view/Google Maps/Mapillary
    - Links to related social events and communities
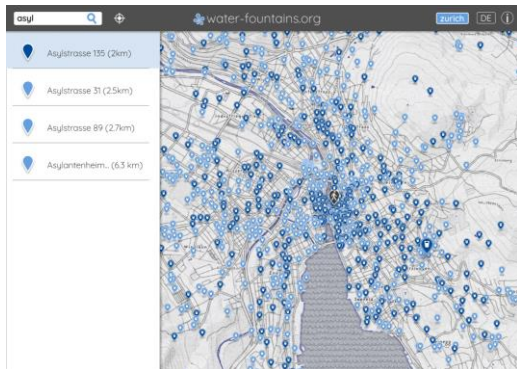   - Suggested walking tours
    - Featured fountains

INTERACTIVE MOCKUP

https://www.figma.com/proto/VtlRvM4aWOWYIEL2j7EcCeVk/water-fountains.org?scaling=contain&node-id=1%3A30 (works best in Google Chrome)
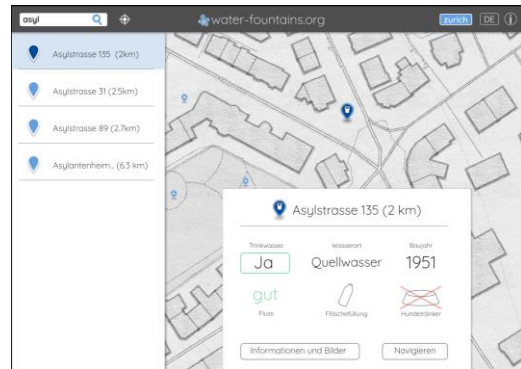
The following features are drawn in the mockup but not included in the WP

- Multicity support
- Edit fountain details / add photo
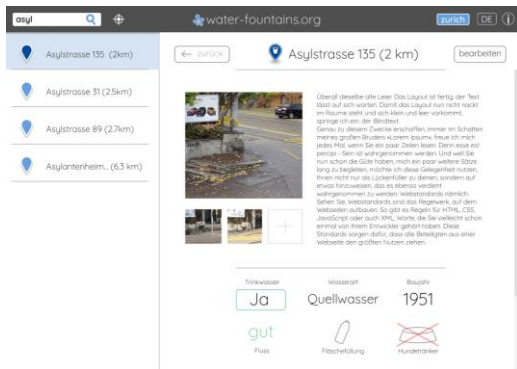
CORE USER FLOW

The user interaction core is the map and/or list of fountains, which can be filtered by the user using the search function. The list is ordered by default according to the distance from the user.
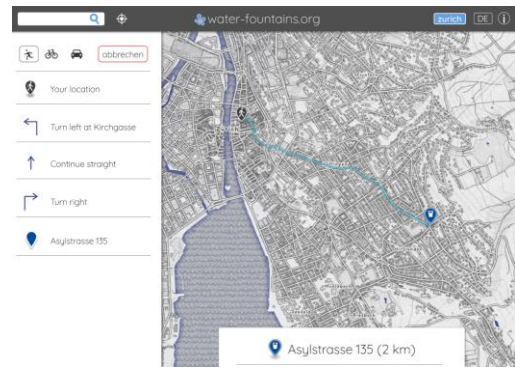

When a fountain is selected, the map zooms in and a 'small information tab' is opened.

From this view, the user has the possibility to view more details (photos and descriptions from Wikimedia/data)

Or navigate to the fountain using a method of choice

Further interaction possibilities like municipal water fountain information, walking tours, and featured fountains can be accessed from the information icon (maybe change to a menu icon) in the upper right.

APPLICATION DATA MANAGEMENT

The website will download all city-specific fountain information excluding photos and store them in memory. The photos will only be hotlinked from Wikimedia and only downloaded when the details of a fountain are viewed.

Software / Framework / Services

- Framework: Angular
  - Matthew has experience with AngularJS, and Angular is the most recent and stable descendant of AngularJS
  - Angular has the possibility to be ported to mobile with wrappers like Ionic or NativeScript.
- Mapping library: Leaflet
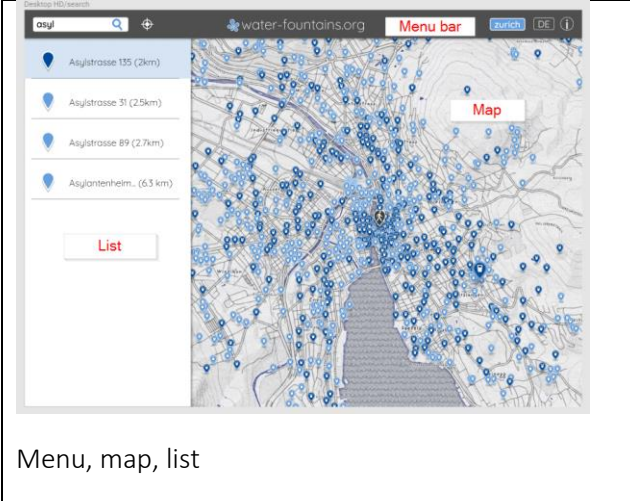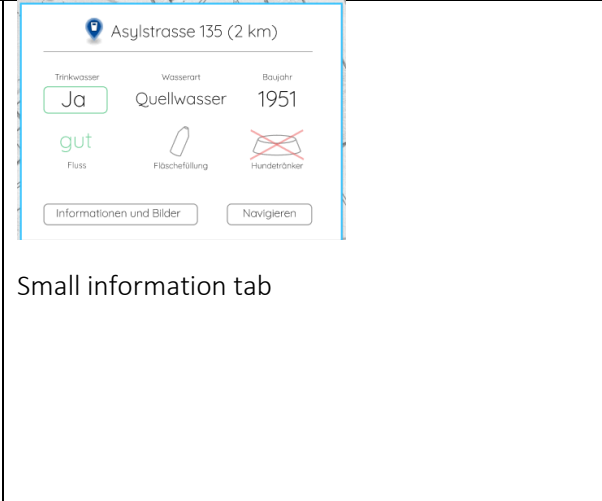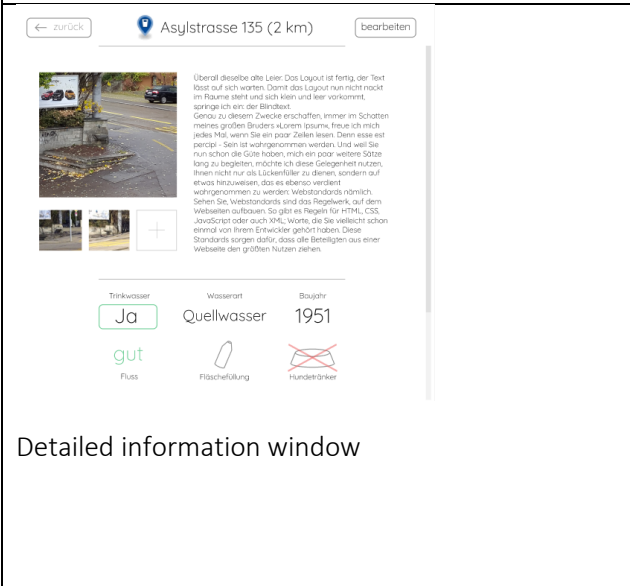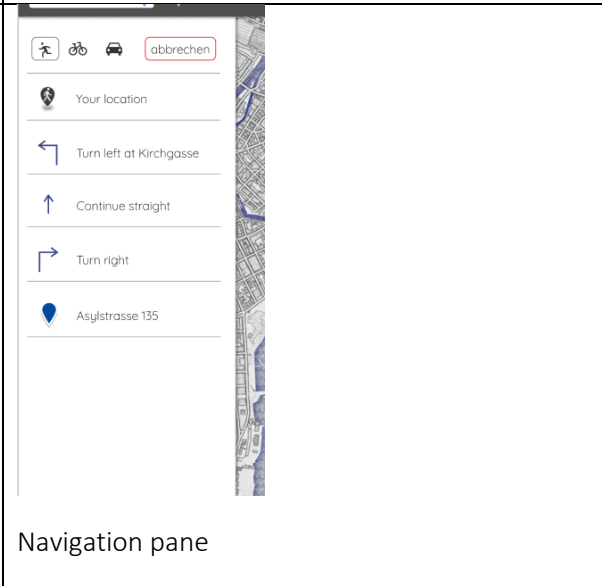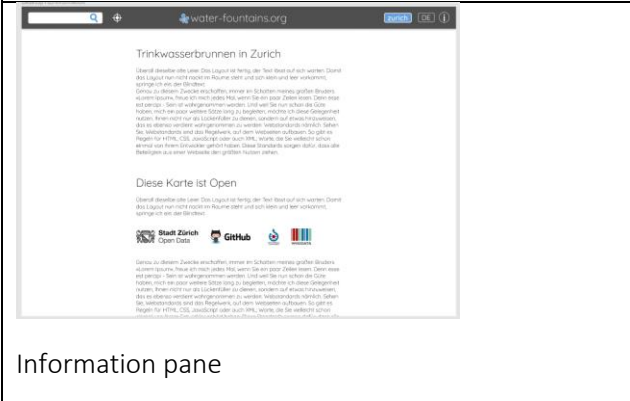- Map services: Mapbox custom map and other OSM sources.

DELIVERABLES

This Work Package will enhance the Web app with the following additional features:

3.a) Improved application architecture

The architecture of the application must be re-written as a single-page web application in Angular, which behaves as an application and has very fast load times between views. This deliverable provides the framework (**NOT content**) for the following panels:

- **Menu bar**: contains logo, search bar, information button, language selection and city selection
- **Map**: interactive map where fountains, the user's location, and navigation paths can be viewed.
- **List**: list in which fountains are displayed and can be filtered (see 3c). The fountains are displayed in order of proximity to user (if user location available) or in alphabetical order.
- **Small information tab**: when a fountain is selected, a summary of information about the fountain is displayed over the map. Information included in this tab depends on what is made available from the data sets, but it could include:
  - Construction year of fountain (e.g. 1951)
  - Type of water (e.g. Well water)
  - Water quality (e.g. drinking water/not drinking water)
- **Detailed information window**: provided more information about the fountain is available from Wikidata and/or Wikimedia, the detailed information window contains a detailed description and photos of the fountain.
- **Navigation pane**: turn-by-turn instructions for how to get to a fountain.
- **Information pane**: full-screen information about Zurich's fountains and about the application. Lots of pointers to
  a) open data and open source concepts
  b) call for action
  b1) on content completion/extension
  b2) technical efforts to make even more open

Menu, map, list

Small information tab

Detailed information window

Navigation pane

Information pane

Responsive: The panels are organized on the page in a responsive way, adapting to screen size. CSS is used instead of JavaScript when possible.

3.b) Map

The map is central to the application. It both displays information and allows the user to interact:

- Display
  - Context in the form of a base map
    - Stylized map
    - Road map

- ▪ Satellite map
  - o Fountain locations
  - o User location
  - o Navigation path to selected fountain
- Interactions
  - o Set user's location (by clicking/tapping on the map)
  - o Select fountain (by clicking/tapping on a fountain icon)
  - o Adjust center and zoom of map (by dragging or pinching map)
  - o Change map background (which can be changed with an icon on the map)

|		
Example Stylized background	Example satellite background	Example road map background

3.c) Fountain filtering

With over 1200 fountains in Zurich alone, search and filtering tools are essential. The search box will allow instantaneous full-text search of the fountain's given names. Additionally, filtering options will allow results to be filtered by

Noteworthiness (only fountains with WikiMedia/Data entry are shown).
Well water (only fountains with well water are shown).
Nearest fountains (only 5 nearest fountains are shown).

The search and filtering tools specify which fountains are displayed in the list and on the map. The map extent adjusts automatically to the search/filter results. Fountain information can be consulted by selecting a fountain either in the list or on the map. An icon in the search box allows the filters to be reset.
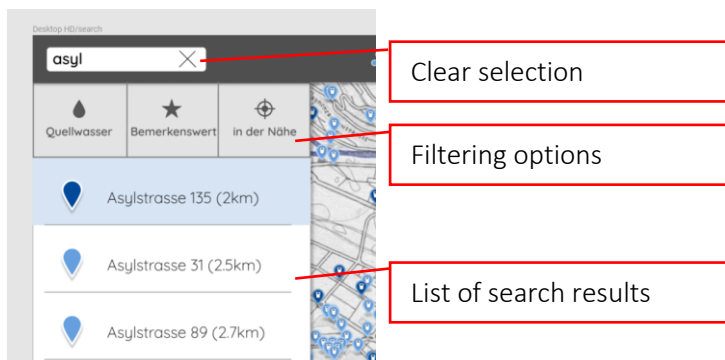
3.d) Navigation

Once the user has found a fountain and wishes to visit it, the app offers a directions service. This service is accessed from the fountain information tabs (small or detailed). When the service is active, a list of turn-by turn directions appears and the route to be followed is drawn on the map. The mode of travel can be selected. Possibly, if localization is activated, the user's position can be shown in real-time on the map.
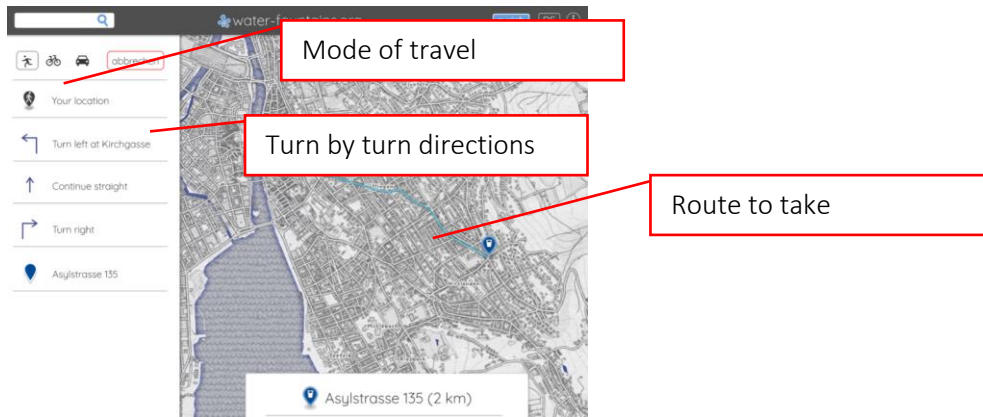
3.e) Load Wikimedia Photos

When a wikidata entry for a fountain is available, the photos directly associated to the entry are hotlinked in the detailed information pane and only loaded when the user views this information.

- Copyright information and photo description are displayed below image.
- A placeholder is shown if no image is available, to suggests to the user that they can contribute their own image.
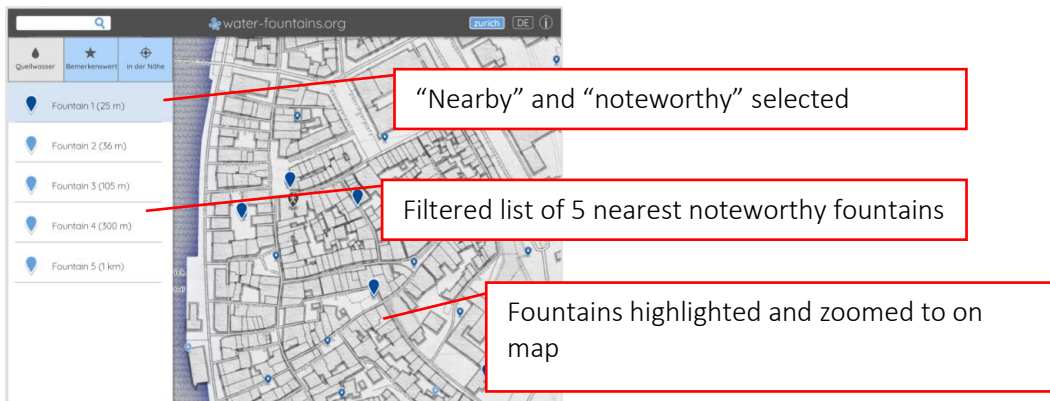
The differences from Lucerne water (to the right) is that the image is not displayed in the small information tab and the image is sourced from Wikimedia, so anyone can upload any image, even inappropriate ones. The aim of the app is not to recreate a Wikimedia/Wikidata browser. Text and attribution content will be used as-is from the Wikidata API, without modification or creation of links (no hyperlinks will be created automatically if e.g. and artist is mentioned in text).

3.f) Nearest Fountain

Users may want to find the nearest fountain, nearest well water fountain, or nearest historical (or noteworthy) fountain. The "nearby" function in the filter pane (see 3c), combined with the other two quick filters, allows this query with ease. When the "nearby" filter is selected, only the five nearest

fountains which fulfill all other search and filter criteria are shown in the list and on the map. Underground fountains of Zurich HB should be excluded from the database until a technical solution for mapping/specifying the location of such fountains is found.

"Nearby" and "noteworthy" selected

Filtered list of 5 nearest noteworthy fountains

Fountains highlighted and zoomed to on map

3.g) Community involvement

Call to actions will be placed around the application:

- In general application information, a paragraph will be dedicated to the possibility to contribute, report omissions and mistakes. The paragraph will have contact information, but not detailed instructions on how to go about modifying Wikidata information. The paragraph will also have links for opendata.ch, https://okfn.org/, https://opensource.org/, https://data.stadt-zuerich.ch/
- At the bottom of the search results list, a notice links to the general information paragraph.
- The placeholder image for missing fountain images links to the same paragraph.
- At the end of the detailed fountain description, a link to the same paragraph.

Message at end of list

Call to add own picture

Call to contribute to description
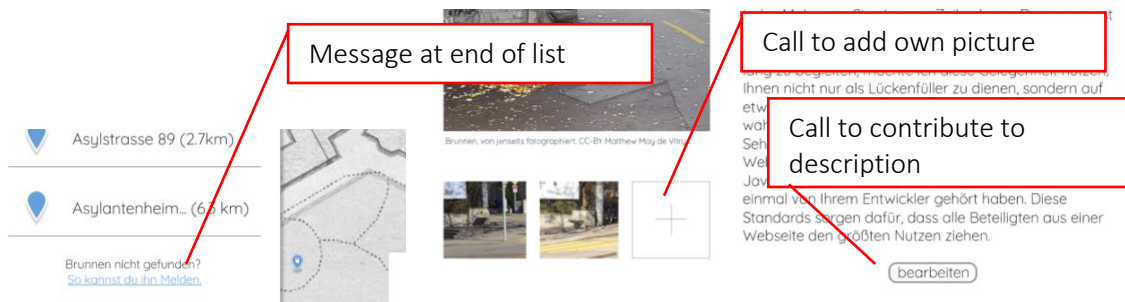
3.h) Pre-Configured walking tours

Recommended walking tours can be viewed within the application. They are accessed from the general information pane.

Featured fountain

Walking tour

They are displayed as custom fountain selection in the map, accompanied by line suggesting how to get from one fountain to the other. The information is supplied by Zurich Tourism and stored in dedicated files on the server (a disaster for scalability, but it will work in the short run). Possibly, if localization is activated, the user's position can be shown in real-time on the map.
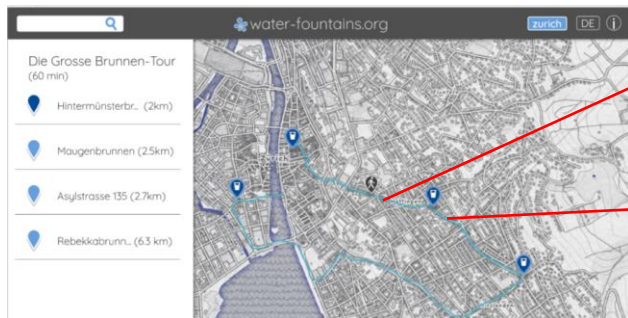
Current user location

Fountains on the tour

3.i) Featured fountains

The general information pane can also have direct links to specific fountains, e.g. "the oldest fountain in Zurich", or "the fountain from which Zwingli drank".

3.j) Links to communities, social groups, and events

The application needs a community to survive in the long run. To encourage this, links to fan groups, events, and the neighbor ambassador are provided in the detailed fountain information, when available.

Place link into "Züri-wie-neu" here?

3.k) Open in external viewer

Links will be provided to external web map services like Google Maps, Google Street View, and Mapillary (the more open ones are displayed more prominently). When possible, the links will point directly to the correct location.

This feature requires links to be generated automatically for each fountain. In locations where no data/imagery is available, a solution must be found to deactivate the option. Alternatively, an interactive Google Street View panorama could be embedded in the website.

The most basic solution, which risks being buggy, will only take an hour to implement while more foolproof solutions with multiple data sources could require up to a day.

3.l) Multilingual

The app interface language can be changed, and the language of fountain information is updated as well. This makes the app more useful to tourists and paves the way for regional generalization.

Drawback: the availability of translations of fountain descriptions, image captions in the user specified language cannot be guaranteed. In fact, multilingual image captions will very rarely be available, if at all.

- The language can be changed with an icon in the menu bar.
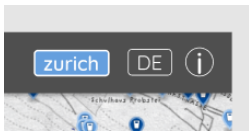
- The language of the app is automatically selected based on the user's browser settings
- The default language of the app is English; translation files (for translators) for French, German, Italian, and Chinese will be made available

3.m) Rich URLs

Rich URLs incorporate information about the current view in query parameters. This makes it possible to share specific views (of a fountain, map style, language) even though the app is a single-page website. It will also make it possible to use the browser's "back" button. This feature requires route parameter parsing in the framework.

Example URL: https://water-fountains.org/map?fountain=1594&lang=en&basemap=coolblue

TASKS

1	Refine design and user experience
1.a	Logo and branding
1.b	Interface, user experience (D/E language translations)

1.c	Map styles and icons
2	**Code management**
2.a	Overall project organization in Github
2.b	Deployment scripts
2.c	Documentation
3	**Coding**
3.a	App architecture
3.b	Map controller
3.c	Data manager, Fountain filtering and searching
3.d	Navigation
3.e	Load and display Wikimedia photos
3.f	Find nearest fountain (requires 3.c)
3.g	Community involvement
3.h	Pre-configured walking tours (requires additional messy or complicated data management)
3.i	Featured fountains
3.j	Links to communities, social groups, and events (requires adjustment to data preprocessing)
3.k	Open in Google Street View
3.l	Multilingual
3.m	Rich URLs