





| | |
|-------------------------|--|
| Project number: | 693729 |
| Project acronym: | smarticipate |
| Project title: | smart open data services and impact assessment for open governance |
| Instrument: | Horizon 2020 |
| Call identifier: | H2020-INSO-2015-CNECT |
| Activity code: | INSO-1-2015 |

| | |
|-------------------------------|------------|
| Start date of Project: | 2016-02-01 |
| Duration: | 36 months |

| | |
|--|---|
| Deliverable reference number and title (as in Annex 1): | D4.1 Software with documentation of intermediate front-end prototypes |
| Due date of deliverable (as in Annex 1): | M20 (shifted from M18); final M24 |
| Actual submission date: | see "History" Table below |
| Revision: | |

| |
|---|
| Organisation name of lead contractor for this deliverable: |
| AIT |

| Project funded by the European Commission, Horizon2020, topic INSO-1-2015 | | |
|--|--|---|
| Dissemination Level | | |
| PU | Public | X |
| PP | Restricted to other programme participants (including the Commission Services) | |
| RE | Restricted to a group specified by the consortium | |
| CO | Confidential, only for members of the consortium (including the Commission Services) | |



European Commission
Information Society and Media

| |
|--|
| Title: |
| Software with documentation of intermediate front-end prototypes |
| Author(s)/Organisation(s): |
| Andreas Sackl / Peter Fröhlich / Jan Peters-Anders (AIT) |
| Working Group: |
| WP4 |
| References: |
| D2.2, D3.1 |

| |
|--|
| Short Description: |
| This document describes the current state of the front-end prototypes of the Smarticipate system, which are used by citizens and city administrators |
| Keywords: |
| front-end, web application, software development |

| History: | | | | |
|-----------------|---------------------------------|--------|-------------------------------|------------|
| Version | Author(s) | Status | Comment | Date |
| 001 | Andreas Sackl Anton Strasser | rfc | Initial draft | 13.09.2017 |
| 002 | Andreas Sackl | Final | Final version after QM review | 26.09.2017 |

| Review: | | | |
|----------------|------------------------------|---------------------------------|------------|
| Version | Reviewer | Comment | Date |
| 001 | Paul McDonald (London) | Smarticipate internal QM review | 22.09.2017 |
| 001 | Patricia Hernandez (Rome) | Smarticipate internal QM review | 25.09.2017 |

smarticipate Glossary of Project Terms

(latest version: http://redmine.smarticipate.eu/projects/smarticipate/wiki/Smarticipate_glossary)

During the conceptual phase of smarticipate the project has defined specific terms for the different parts of the platform and its users. In order to help the reader what the smarticipate concept team means with these terms find here a glossary which you can refer back to while reading the deliverable.

| Term | Meaning |
|-------------------------------|---|
| App or smarticipateApp | Is the piece of smarticipate software that will be available in the App stores of the different mobile platforms (Android, iOS etc.). It will be a Cordova container (https://cordova.apache.org/) that will display an HTML5 site. This site will be loadable in a desktop browser, too. It is planned that every City has its own App e.g. "smarticipateApp Hamburg" holding different Topics like e.g. "Propose a Tree". |
| Application | Is smarticipate's front end/back end bundle, so the final product of the project (end-product name: smarticipate Platform) |
| Frontend | The part of the smarticipate Platform that is running on the users' devices (Graphical User Interface (GUI) etc.) i.e. the part of the application that can be directly accessed by the users, i.e. the smarticipateApp. Frontend is the counterpart of Backend. |
| New Public Service | New Public Services are products or services by citizens which replace or add an existing municipality service. |
| Proposal | In the context of smarticipate a proposal can be e.g. a tree a user proposes to be planted, so proposal is a proposal within a Topic. |
| Service Creation | The purpose of the Service Creation is to make the data sets accessible and easy to use for everyone. |
| Smartathon | The name of a planning exercise within smarticipate. |
| smarticipate | The name of the EU project. Core of it is the smarticipate Platform, which is a software. |
| smarticipateApp | See App. |
| smarticipate Platform | See Application. |
| Topic | Is one Topic of a smarticipateApp. Each city has a smarticipateApp of its own, so the smarticipate Platform will consist of many smarticipateApps holding different city related Topics in the end. E.g. one Hamburg Topic might be "Propose a Tree" within the Hamburg smarticipateApp. |
| Urban Story | Urban Story is defined here as a smarticipate use case (the term Urban Story has been chosen to distinguish it from the term Use Case in the Redmine development planning). The 6 Urban Stories are: <ul style="list-style-type: none"> * City Living, Planning Application * Tree Cadastre, Binding Land Use Planning * Recovery Planning, Regulation for Green Areas |

Table of Contents:

| | |
|---------------------------------------|----|
| 1 Introduction | 5 |
| 2 Frameworks | 5 |
| 2.1 React..... | 5 |
| 2.2 Blueprint..... | 5 |
| 2.3 Redux..... | 6 |
| 2.4 Angular | 6 |
| 2.5 Angular Material..... | 7 |
| 2.6 Leaflet | 7 |
| 2.7 Cesium..... | 8 |
| 3 Current state of the Software | 8 |
| 3.1 End User Interface | 8 |
| 3.2 Admin Interface | 14 |
| 4 Next steps..... | 18 |

Table of Figures:

| | |
|---|----|
| Figure 1: Examples of blueprint UI components (from http://blueprintjs.com)..... | 6 |
| Figure 2: Visualization of Redux concept (from http://www.dotnetcurry.com/reactjs/1356/redux-pattern-tutorial) | 6 |
| Figure 3: Angular architecture overview (from: https://angular.io/guide/architecture) | 7 |
| Figure 4: Examples of Angular Material UI components (from https://material.angular.io/components) | 7 |
| Figure 5: Examples of Leaflet maps (from http://leafletjs.com)..... | 8 |
| Figure 6: Examples of cesium (from https://cesiumjs.org)..... | 8 |
| Figure 7: Entry screen of the Smarticipate installation of the city of Hamburg | 10 |
| Figure 8: Login screen of the web front end | 11 |
| Figure 9: Sign up screen | 11 |
| Figure 10: Initial screen of the topic "Hamburg Tree Planting"..... | 12 |
| Figure 11: (from left to right) Layer selection and additional information selection; "Hamburg Geobasisdaten"-View without tree information; "OpenStreetMap Mapnik" View without tree information; "OpenStreetMap Mapnik" View with tree information..... | 13 |
| Figure 12: Details of a selected proposal | 13 |
| Figure 13: Creation of a new proposal..... | 14 |
| Figure 14: Start screen of the Smarticipate admin interface..... | 14 |
| Figure 15: Dialogue for creating a new topic..... | 16 |
| Figure 16: Option 1 view for topic creation | 16 |
| Figure 17: Print out for onsite information campaign | 17 |
| Figure 18: defining an area for a topic..... | 17 |

1 Introduction

This document gives an overview about the current state of the front end software development. Additionally, the used web development frameworks are discussed. Please note that the implementation of the software is a continuous process, i.e., during the duration of the project, updated versions of this deliverable will be provided.

2 Frameworks

2.1 React

| | | | |
|----------|-------------------------------------|-----------|---|
| Name: | React | URL: | https://facebook.github.io/react/ |
| Version: | v15.6 | Used in: | End user interface |
| Type: | JavaScript library for building UIs | Language: | JavaScript |

React is a popular JavaScript library for building user interfaces and single page applications (SPA). It is maintained by Facebook, Instagram and community and is currently amongst the most popular JavaScript libraries for Web interface development. Notable features include one-way data flow, virtual DOM, and JSX. These features allow the developer to decompose components, e.g. a root component, into smaller components as needed, thereby greatly improving modularity and reusability. The main concern of the React library is to render UI components. A Web developer can implement additional functionality as needed or choose from many existing third party JavaScript libraries such as libraries for state management, styling, routing, form components, and map components. We use React in Smarticipate for building its end user Web interface.

2.2 Blueprint

| | | | |
|----------|------------|------------|---|
| Name: | Blueprint | Web: | http://blueprintjs.com |
| Version: | v1.27 | Used in: | End user interface |
| Type: | UI toolkit | Languages: | HTML, CSS, JavaScript |

“Blueprint is a collection of React UI components that cover the majority of the common interface elements, patterns, and interactions on the web.” The Blueprint UI toolkit was evaluated against similar UI toolkits like Bootstrap, Semantic UI, and toolkits based on Material Design. Using a UI toolkit helps creating a consistent look and feel of the user interface and reduce boilerplate. Blueprint provides ready to use components such as menus, buttons, icons, dialogs, navbars, and the corresponding CSS styles. In contrast to plain CSS toolkits, the Blueprint components are based on React, which allows for easy integration simply by importing said components. It also comes with Sketch assets that can be used in the Sketch-App for easy prototyping.

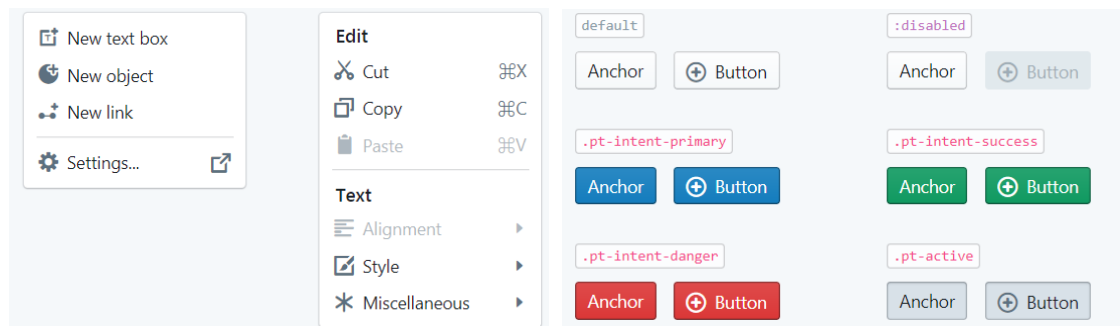


Figure 1: Examples of blueprint UI components (from <http://blueprintjs.com>)

2.3 Redux

| | | | |
|----------|------------------|-----------|---|
| Name: | Redux | Web: | http://redux.js.org/ |
| Version: | V3.7 | Used in: | End user interface |
| Type: | State management | Language: | JavaScript |

Redux is a predictable state container for JavaScript apps. The Web interface is a single page application (SPA) that runs in the users Web browser. The application fetches data from the server or creates data locally and stores it in the local Redux store, which is a single object tree. The Redux store acts a single source of truth and the only way to change it is by emitting Redux actions. React components that are connected to the Redux state can be updated automatically on state changes. We currently use Redux to manage the local state of topics, proposals, automated feedback, authentication, and form state (using the Redux Form library).

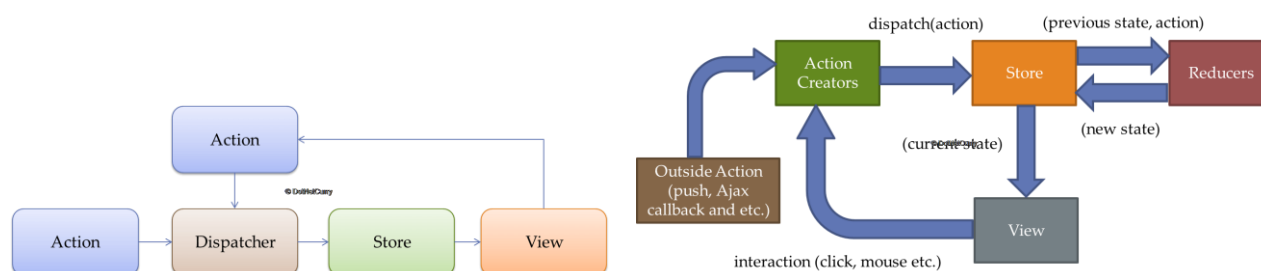


Figure 2: Visualization of Redux concept (from <http://www.dotnetcurry.com/reactjs/1356/redux-pattern-tutorial>)

2.4 Angular

| | | | |
|----------|---------|----------|---|
| Name: | Angular | URL: | https://angular.io |
| Version: | V4.3. | Used in: | Admin-Interface |

Angular is a TypeScript-based open-source front-end web application platform led by the Angular Team at Google and by a community of individuals and corporations to address all of the parts of the developer's workflow while building complex web applications. According to JavaScript analytics service Libscore, Angular is used on the websites of Wolfram Alpha, NBC, Walgreens, Intel, Sprint, ABC News, and about

12,000 other sites out of 1 million tested in October 2016. Angular is currently in the top 100 of the most starred projects on GitHub.

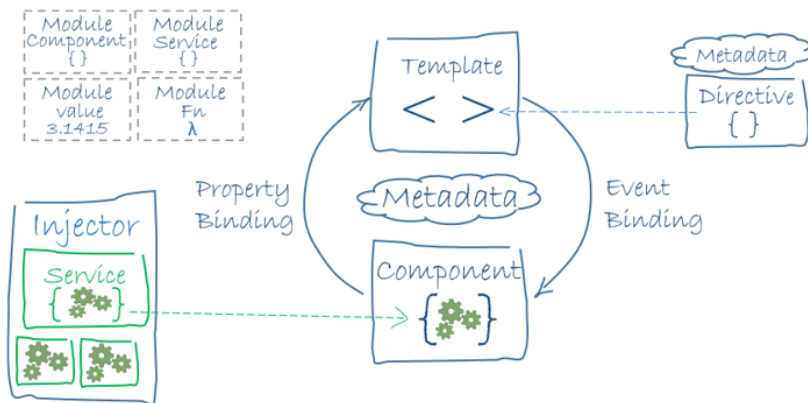


Figure 3: Angular architecture overview (from: <https://angular.io/guide/architecture>)

2.5 Angular Material

| | | | |
|----------|------------------|----------|---|
| Name: | Angular Material | URL: | https://material.angular.io |
| Version: | V2.0.0-beta.10 | Used in: | Admin-Interface |

Angular Material is a UI component library for Angular JS developers. Angular Material components help in constructing attractive, consistent, and functional web pages and web applications while adhering to modern web design principles like browser portability, device independence, and graceful degradation.

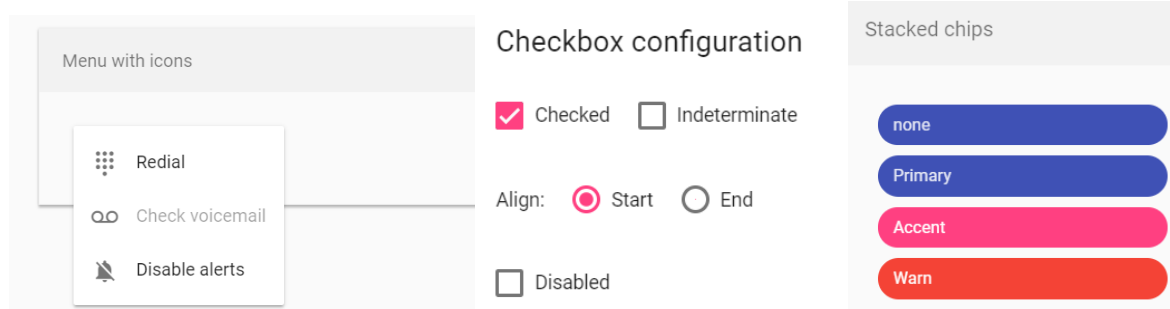


Figure 4: Examples of Angular Material UI components (from <https://material.angular.io/components>)

2.6 Leaflet

| | | | |
|----------|------------|----------|---|
| Name: | leaflet.js | Web: | http://leafletjs.com |
| Version: | v1.2.0 | Used in: | Admin-Interface, End user interface |

Leaflet is the leading open-source JavaScript library for mobile-friendly interactive maps. Weighing just about 38 KB of JS, it has all the mapping features most developers ever need. Leaflet is designed with simplicity, performance and usability in mind. It works efficiently across all major desktop and mobile

platforms, can be extended with lots of plugins, has a beautiful, easy to use and well-documented API and a simple, readable source code that is a joy to contribute to.

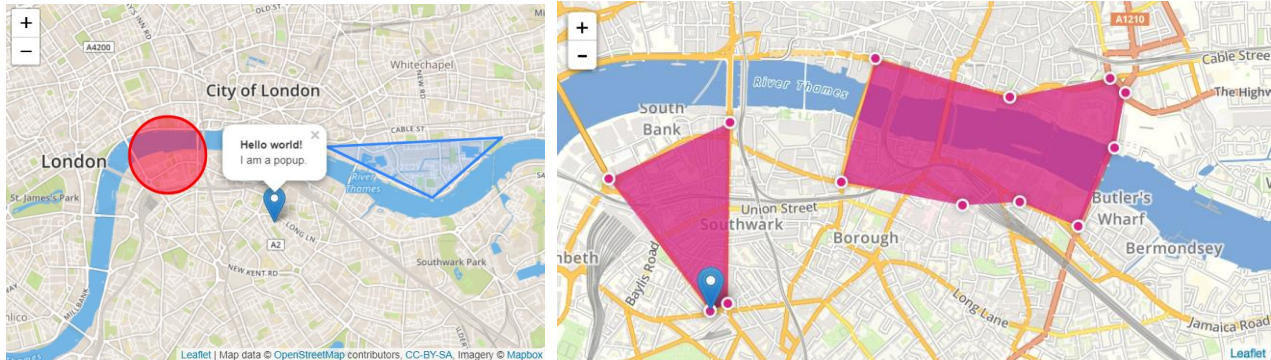


Figure 5: Examples of Leaflet maps (from <http://leafletjs.com>)

2.7 Cesium

| | | | |
|----------|-----------|----------|---|
| Name: | cesium.js | Web: | https://cesiumjs.org/ |
| Version: | v1.37 | Used in: | End user interface |

Cesium is an open-source JavaScript library for world-class 3D globes and maps. In Smarticipate, it will be used in future implementations to visualize 3D objects, which are part of a proposal, via the Smarticipate platform. So, the end user is able to see how a proposed object, e.g., a tree would look like in the proposed area.



Figure 6: Examples of cesium (from <https://cesiumjs.org>)

3 Current state of the Software

In this section, the current state of the front end is described via screenshots.

3.1 End User Interface

Based on the work presented in previous deliverables and on regular meetings with project partners, the citizen front end is developed in WP4. In this section, an overview is given about the current state of the interface via screenshots and descriptions.

In the following list, some information about the architecture and mechanism, which cannot be displayed via screenshots:

- Communication: The Web application communicates with the micro-services provided by the Smarticipate platform over a REST-API
- Layout: the layout is composed of custom React components that use CSS Flexible Boxes. The layout can dynamically change its appearance based on given property values, e.g., the show/hide the sidebar.
- Routing: The URLs of the application define what is presented to the user, this configuration is often referred to as routing. Given some URL, the routing specifies which components are rendered, this includes changes to the layout based on the URL and fetching relevant data from the server that is needed by the components.
- Custom building blocks: some modules and components specific to this application are not part of any library and needed to be developed from scratch, this includes UI components such as custom forms and views as well as middleware components for server communication and data normalization.
- Data fetching, error and success handling: a crucial part of this application is transparent communication with the various services. For the user the interaction should feel like it is a local application. Loading data is indicated to the user with progress bars and spinners and, if communication fails, a proper error messages is presented to the user. Some actions, such a successful login, are communicated with success messages. The server communication is implemented in such a way that it can be extended easily, simply by adding corresponding Redux actions and reducers for server responses. As described in the Redux section, components that are connected to the Redux store do have all the necessary information to (possibly asynchronous) events created during server communication.
- Authentication: a user can sign up, activate his account, login, and reset his password thru the Web interface (see screenshots below). The authentication is implemented using JSON Web Tokens (JWT), which are requested from the user service and stored in the application using Redux.

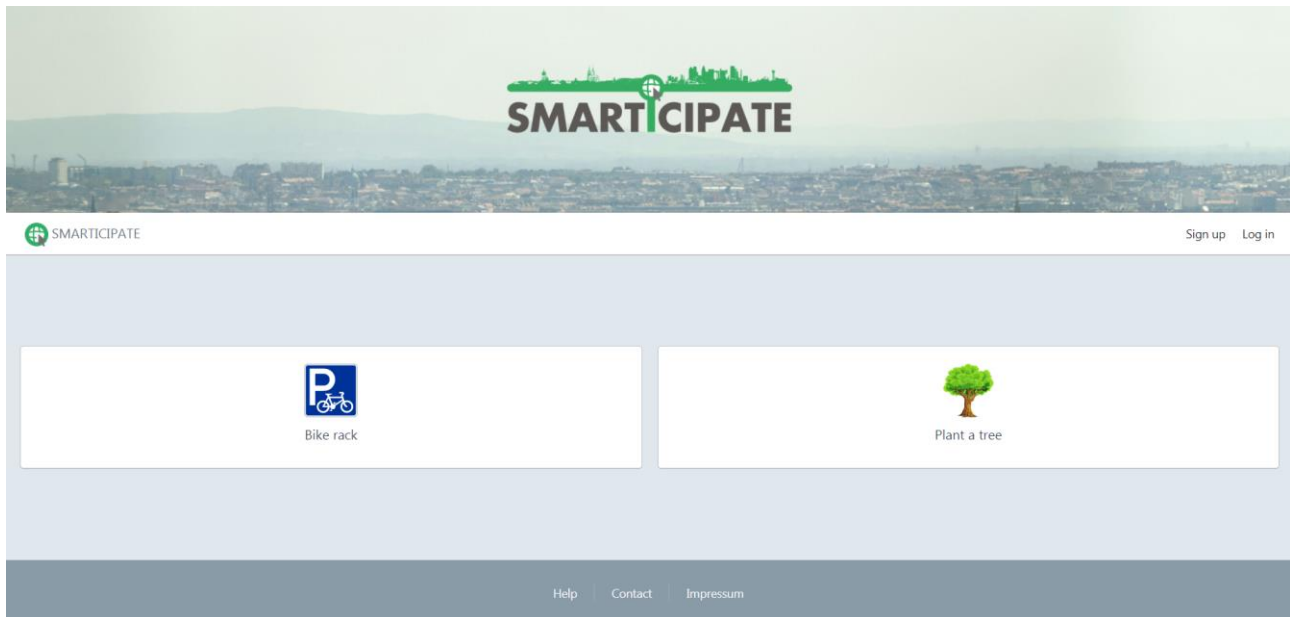
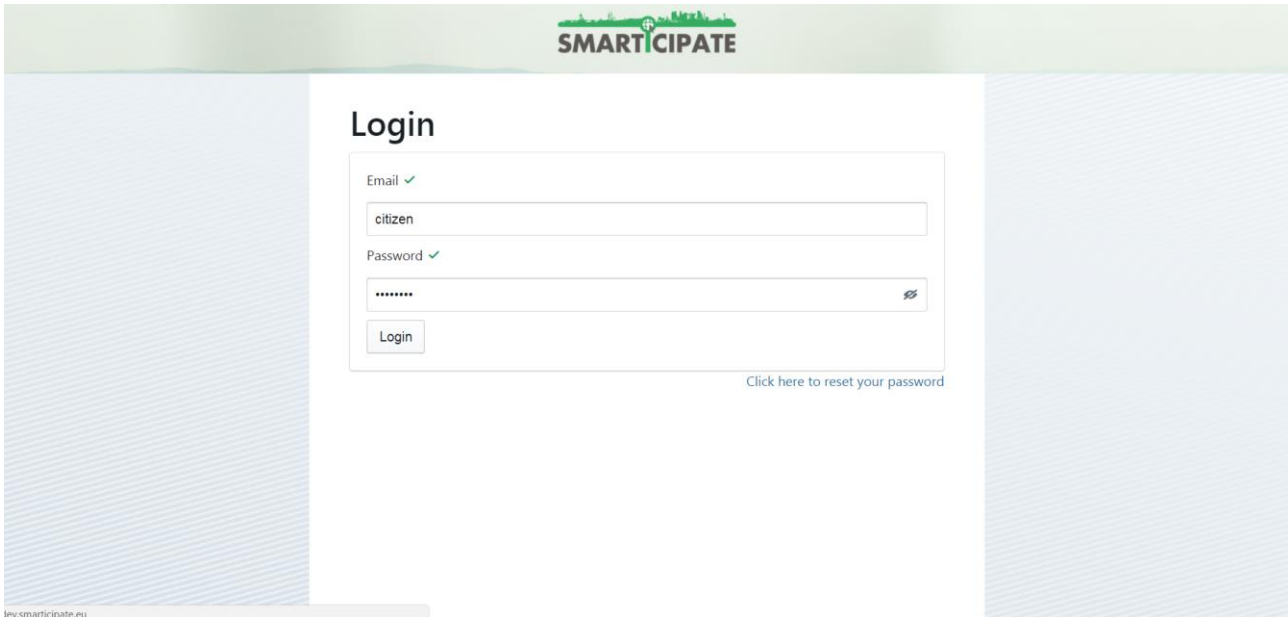


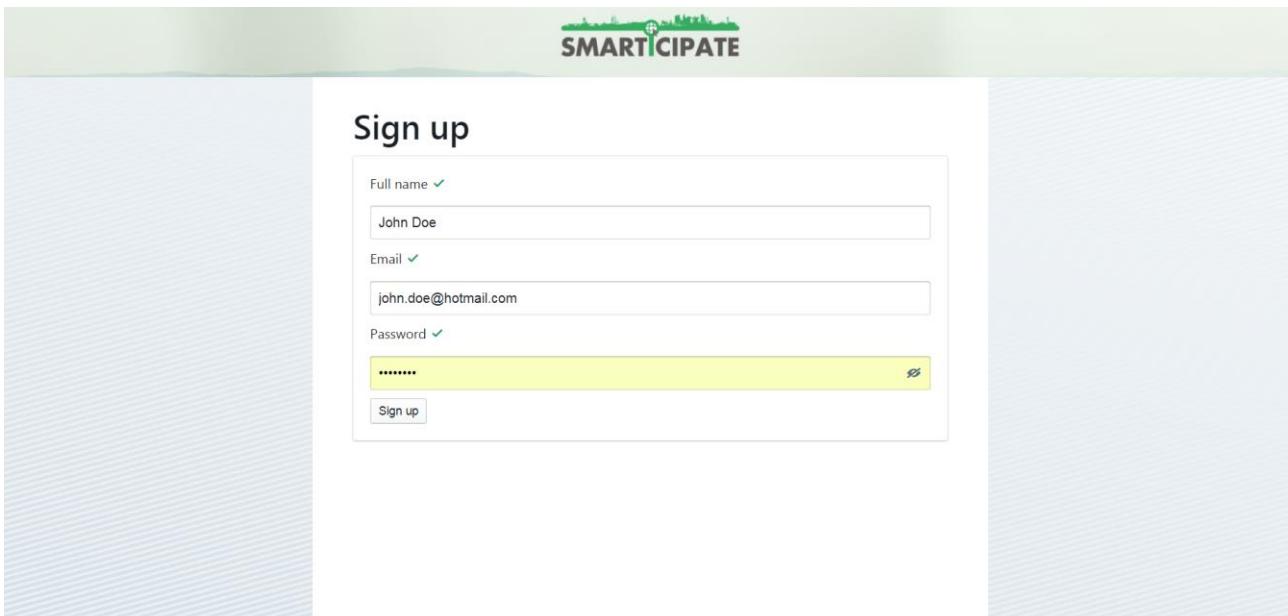
Figure 7: Entry screen of the Smarticipate installation of the city of Hamburg.

In Figure 7, the entry screen which is shown to citizens is depicted. In this example, two topics are presented: The topic on the left side should motivate citizens to provide suggestions where to install bike racks in a defined area of the city of Hamburg. The topic on the right side provides the possibility of making tree planting proposals. The creation of topics is done in the admin interface, which is shown in the next section. At the moment, any user can access the available topics and see the already made proposals and read comments, i.e., it is not necessary to sign up to the Smarticipate platform or to log in. For active participation, i.e., making proposals or comment on existing proposals, the user has to register and to log in on the web site. Additionally, the admin of the topic (mostly an employee of the city) has to define which registered users are allowed to actively contribute. For example, only the inhabitants of a certain district are allowed to make tree planting proposals in their area. At the moment, in the footer only links to basic information like “Help”, “Contact” and “Impressum” are displayed. However, additional information and/or larger and also interactive elements can be added easily due to the modular structure of the web system.



The login screen features a header with the SMARTICIPATE logo. The main content area is titled "Login" and contains a form with two input fields: "Email" (with a green checkmark) and "Password" (with a green checkmark). The email field contains the text "citizen" and the password field contains a masked password "*****". Below the password field is a "Login" button. A link "Click here to reset your password" is located below the form. The footer contains the URL "lev.smarticipate.eu".

Figure 8: Login screen of the web front end



The sign up screen features a header with the SMARTICIPATE logo. The main content area is titled "Sign up" and contains a form with three input fields: "Full name" (with a green checkmark), "Email" (with a green checkmark), and "Password" (with a green checkmark). The full name field contains the text "John Doe", the email field contains the text "john.doe@hotmail.com", and the password field contains a masked password "*****". Below the password field is a "Sign up" button. The footer is empty.

Figure 9: Sign up screen

In Figure 8 the login screen and in Figure 9 the sign up screen are shown. Especially in the case of the sign up screen, the current implementation of the system only covers basic functionalities, i.e., only the name, the Email-address and a self-chosen password is set. Obviously, additional fields can be added, also the architecture of the implementation provides an easy way to implement additional features like checking new user registrations with city-related resident information if necessary.

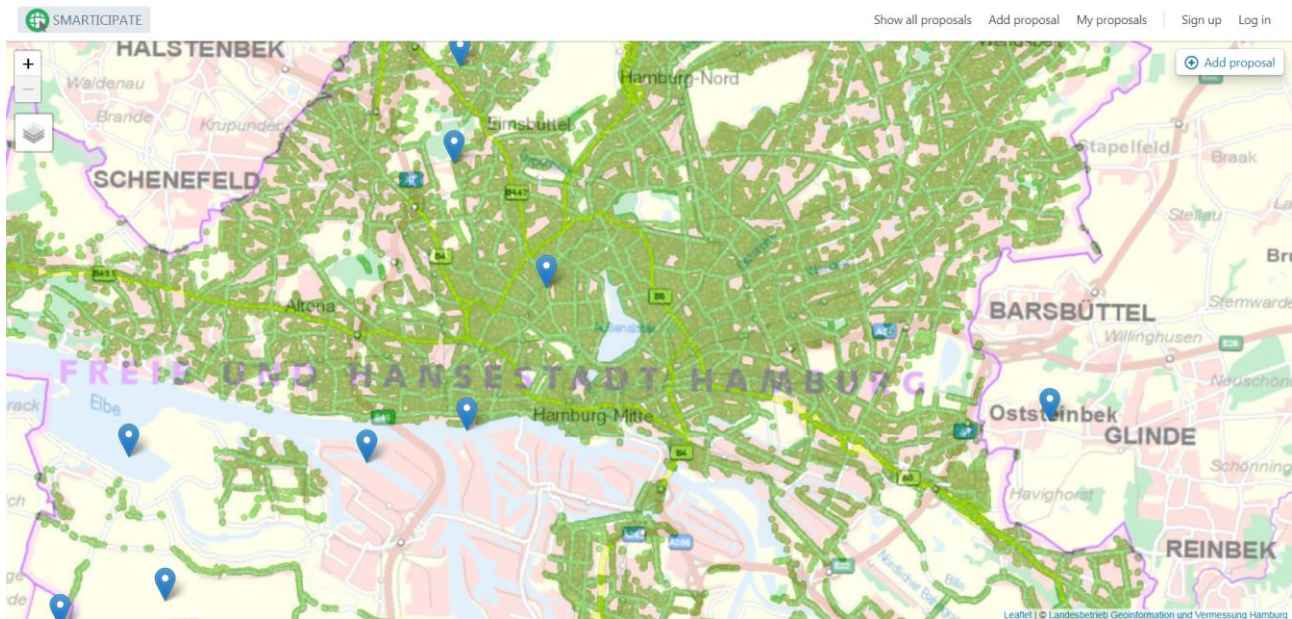


Figure 10: Initial screen of the topic "Hamburg Tree Planting"

Figure 10 shows the home screen of the selected topic "Hamburg Tree Planting". A zooming interface is shown in the top right corner and below a layer-selector is visible. Figure 11 gives an overview about the currently available views (Hamburg Geobasisdaten and OpenStreetMap Mapnik) and the additional layer "Hamburg Baumkataster" which shows every tree in Hamburg which is registered via the city.

In the top right corner of Figure 10 the main navigation area is shown. If the item "Show all proposals" is selected, all proposals are displayed. In this use case, blue markers indicate the position of tree proposals. Each marker can be selected by simply clicking on it. After a marker is selected additional information is shown at the right side of the screen, please see Figure 12. The depicted information does not reflect real data at the moment, but it demonstrates which kind of information will be shown. The interface to the automatic feedback system is already established, i.e., depending on the set rules and databases, for every individual proposal the user is able to receive feedback immediately after a proposal is set.

The basic functionality for creating a new proposal is shown in Figure 13. The user is able to define a title and a short description, and she can choose from the options which were defined in the topic creation process in the admin interface (see next section).

Additional links displayed via icons are visible in the top right corner of Figure 13 which will be used in future implementations to access the user settings (e.g. for changing personal information) and to access the messages sent by admins or other citizens to the user.



Figure 11: (from left to right) Layer selection and additional information selection; “Hamburg Geobasisdaten”-View without tree information; “OpenStreetMap Mapnik” View without tree information; “OpenStreetMap Mapnik” View with tree information

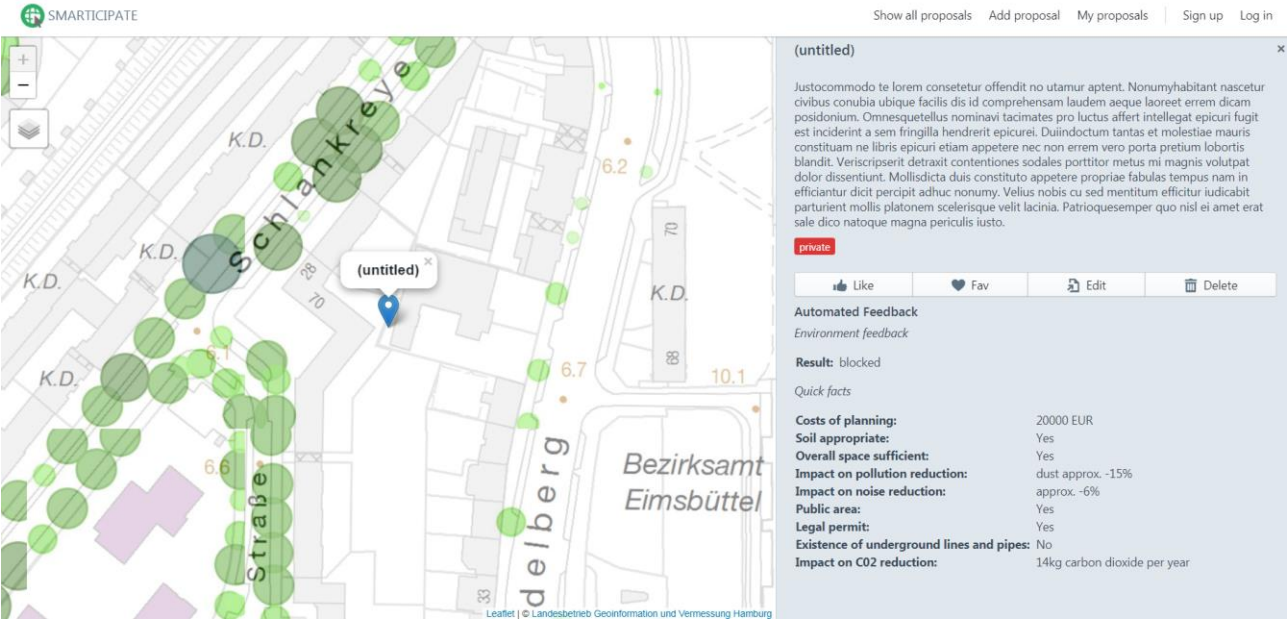


Figure 12: Details of a selected proposal

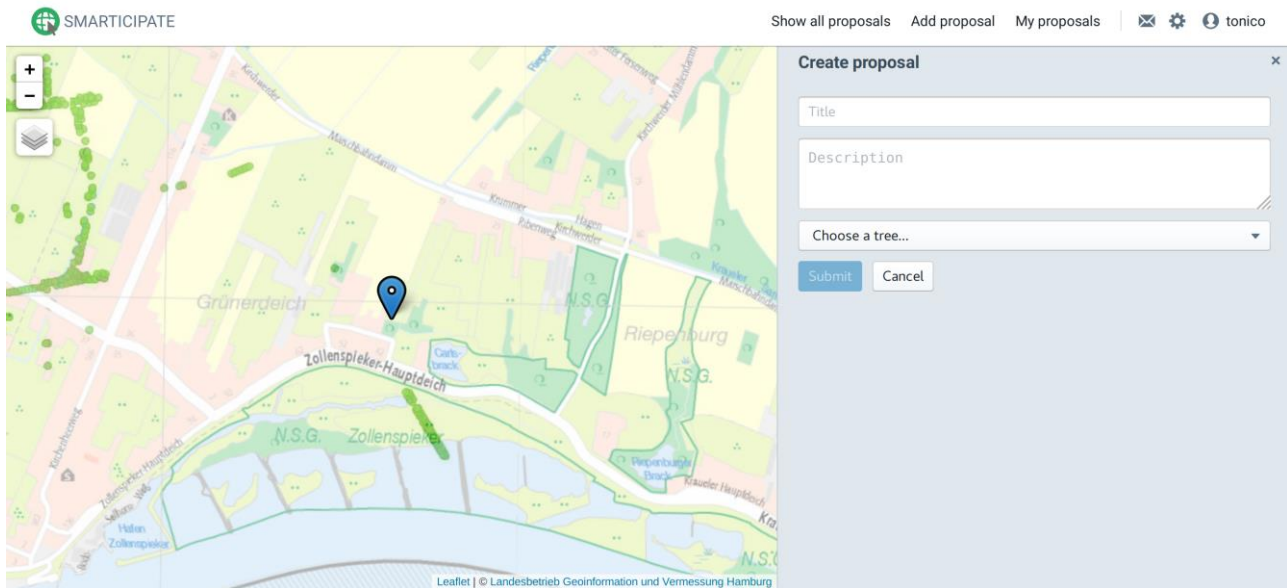


Figure 13: Creation of a new proposal

3.2 Admin Interface

The interface described in section 3.1 is used by the citizens to interact with the Smarticipate system. At the moment a defined list of topics can be accessed and within these topics individual proposals are created, commented and shared. These topics are created by employees of the city, in this section the current state of the administration interface is presented and discussed.

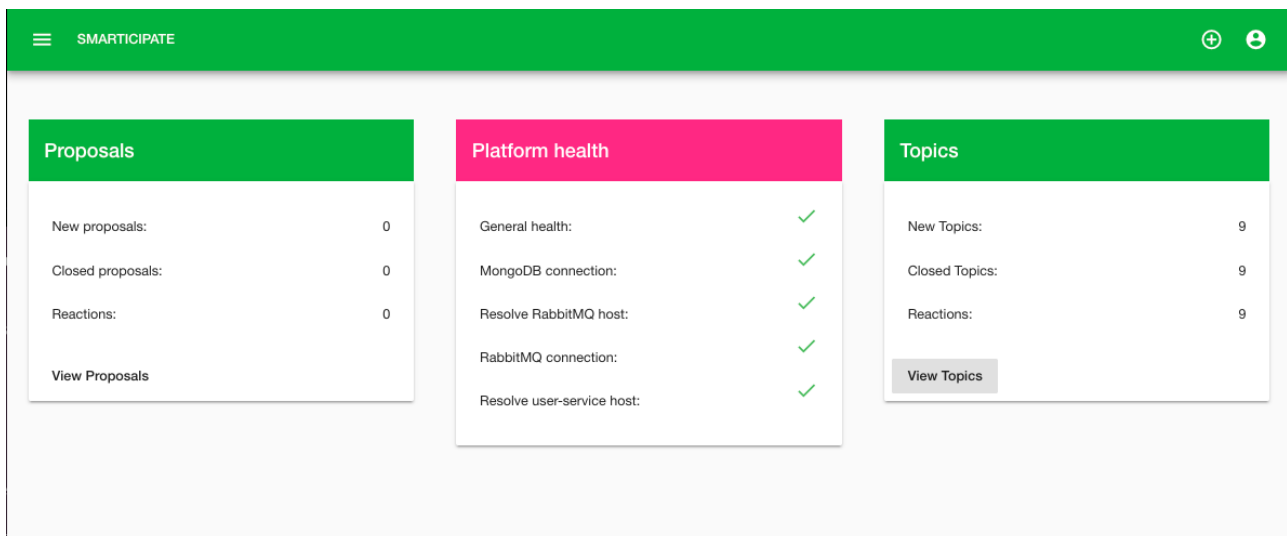


Figure 14: Start screen of the Smarticipate admin interface

In Figure 1 the elements of the administration start screen are depicted. In general, a short overview about existing proposals and topics is given (left and right column) and a short overview about system parameters is given, for example the database connection status.

In the top right corner, a plus symbol can be used to create a new topic, which is shown in Figure 15. The admin is able to define the title and a short description of the topic. At the moment, several options are available, which can be chosen by the citizens. In the example depicted in Figure 15, the inhabitant is able to choose from four available options, i.e., she can propose planting trees, planting flowers, building a fountain or placing chairs. For each option, additional options can be defined. In Figure 16, additional options for object 1 (tree planting) are depicted. In this case, the user can choose from five different tree types (red maple, silver maple, sugar maple, oak and chestnut oak). Additionally, the amount of trees can be defined (one to five trees). For graphical representation, icons for the objects can be uploaded via the “upload icon” - button, which are used in the 2D map representation in the end user interface. Also 3D models for the objects can be uploaded via the “Upload 3D model” -button. In Figure 15, below the object list, three buttons are visible for user selection, area definition and rule adding. The topic admin is able to define which city inhabitants are allowed to actively participate. Hence, a text file can be uploaded to define this group. Obviously, these user selection approach can be optimized, i.e., avoiding text file uploads and using dialogues which display users and user groups directly in the admin interface. Depending on the requirements of the project partners and considering implementation effort, additional user selection features might be implemented. Also, the admin is able to define an area in which proposals can be located. This is done via a 2D map and a polygon, created via simple mouse clicks, see Figure 18. The Smarticipate system enables automatic feedback, based on the individual proposals and on the databases provided by the cities. Additionally, a set of rules need to be defined. For example, trees need free space to grow, also a certain distance is required between a tree and other objects. These rules need to be defined and then uploaded to the system. At the moment, this is done via a simple upload file dialogue, which is triggered via the “add rules” -button in Figure 15. Figure 17 depicts the print out, which is generated with the information provided while a new topic is created. It can be printed out and used onsite, e.g., at a building ground, for which citizens should provide input. Additionally, a QR-code is automatically created which directly leads to the related topic in the Smarticipate system.

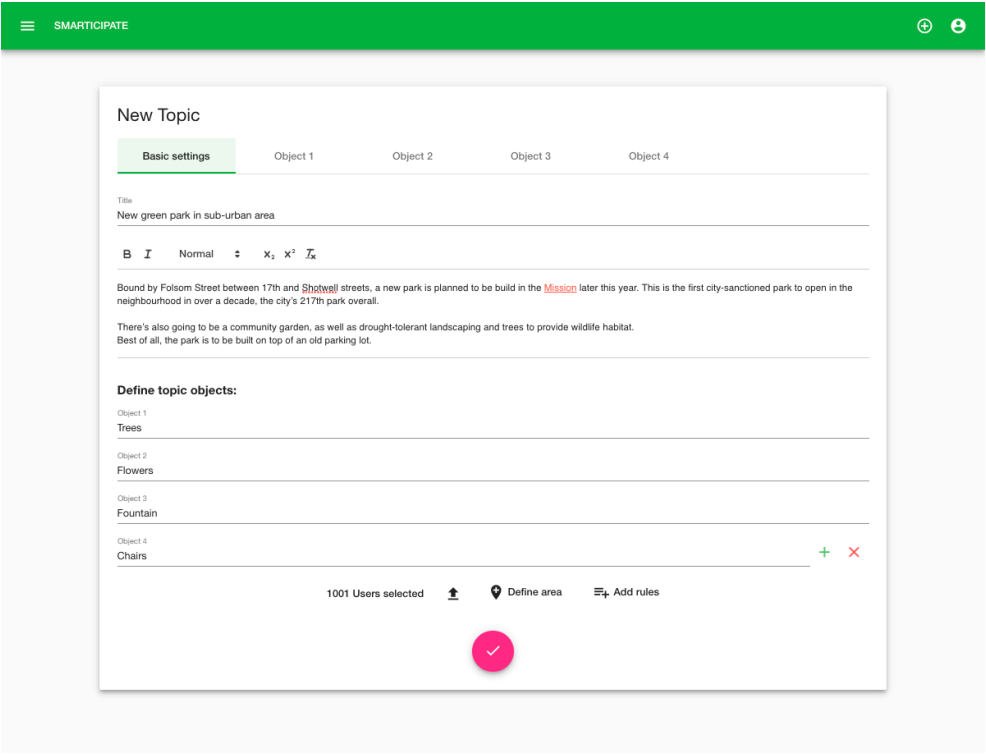


Figure 15: Dialogue for creating a new topic

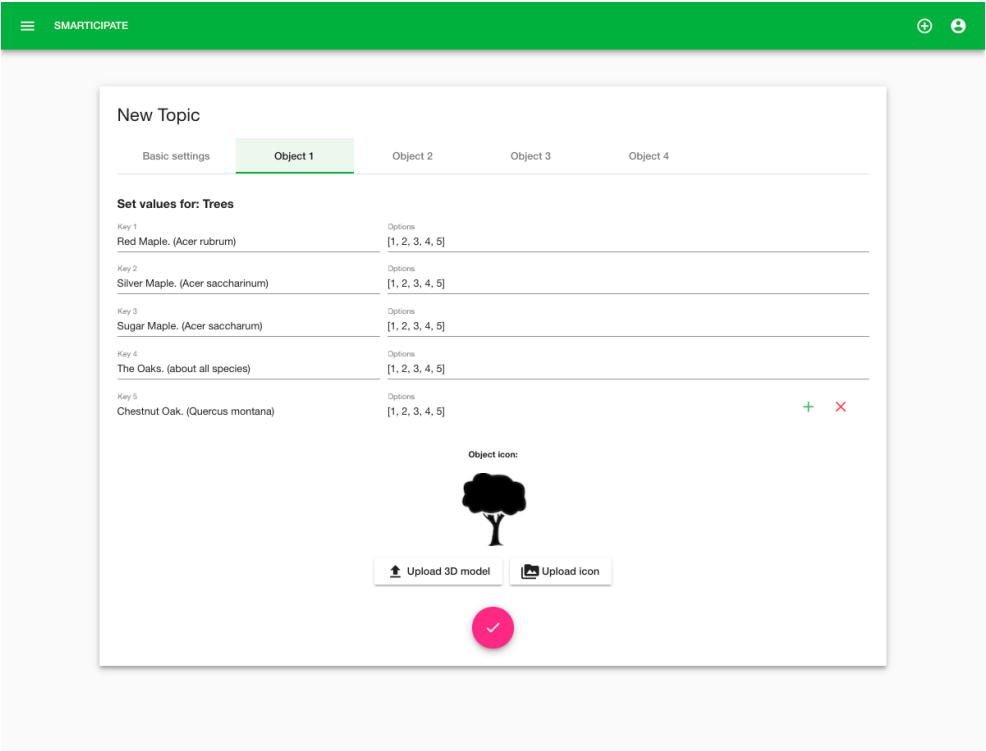


Figure 16: Option 1 view for topic creation

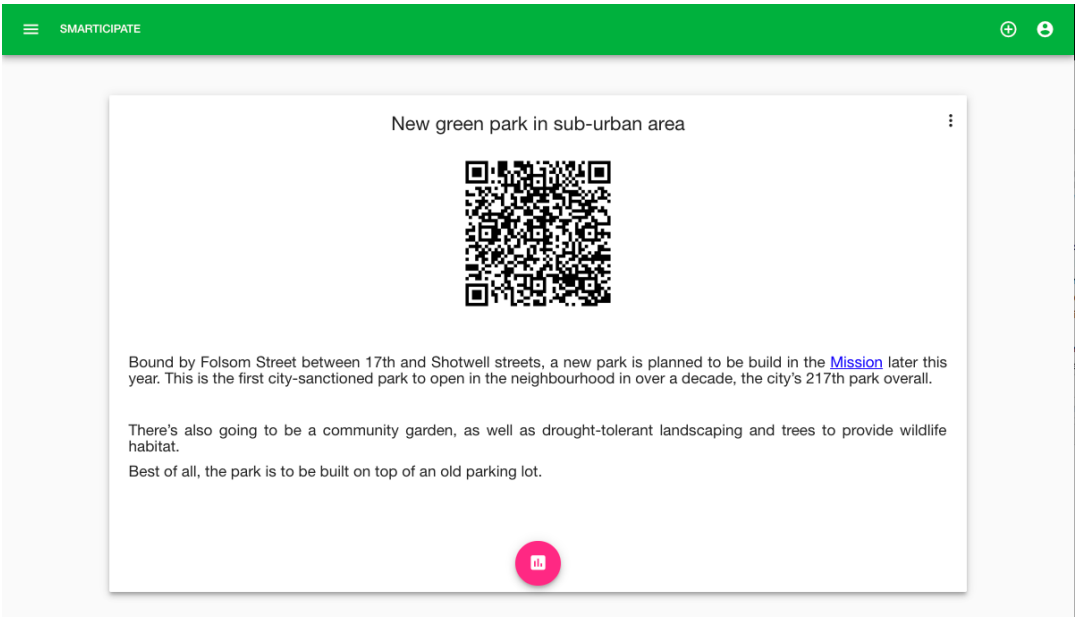


Figure 17: Print out for onsite information campaign

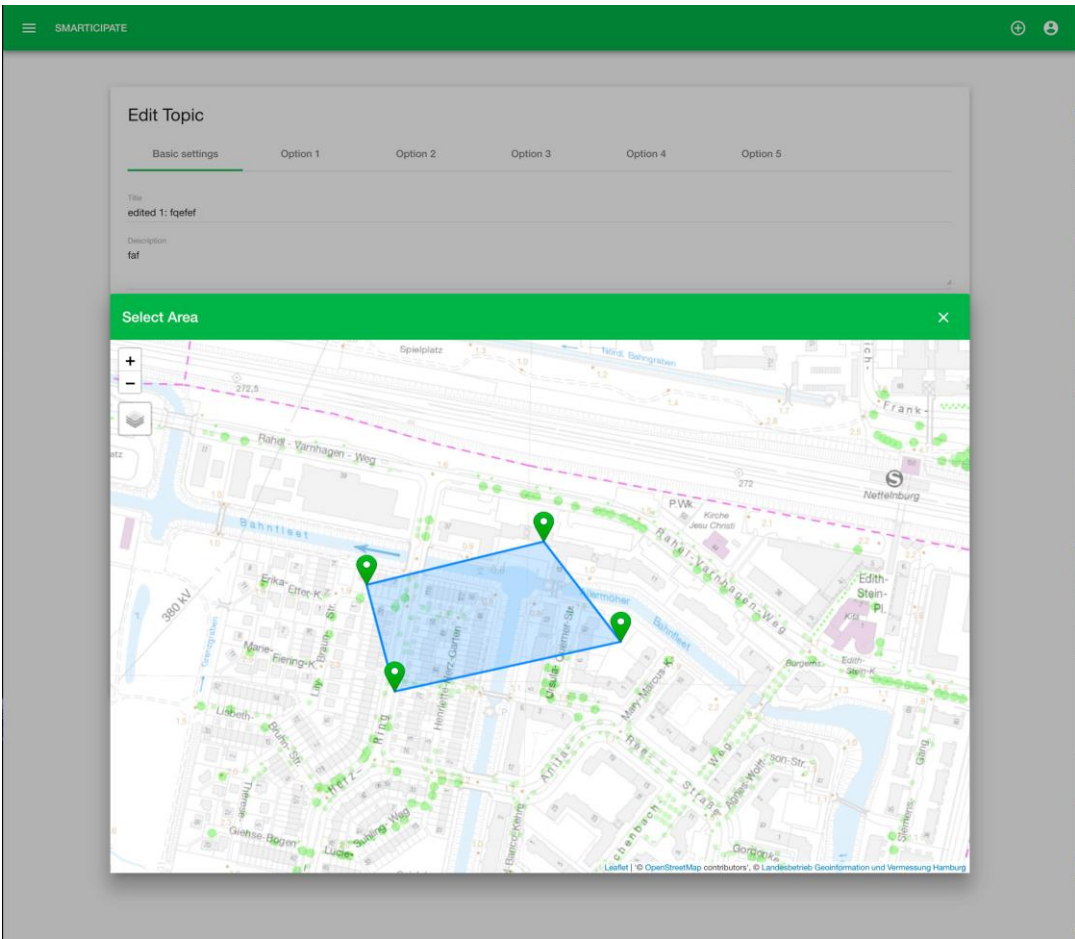


Figure 18: defining an area for a topic

4 Next steps

In general, the next steps in the implementation process are discussed during the bi-weekly SCRUM sprint meetings. However, here are some main tasks which are on the agenda for the next implementation phases:

- Creation of sensible sample data
- Extend admin interface design
- Style objects (GeoJSON Features) based on their state, grouping, owner, etc.
- Use map-layers from topics definition
- Fix various bugs, including troubles with double authentication on reference server, and dead links of proposals, etc.
- Implement topic navigation
- Implement proper routing such that resources can be bookmarked
- Implement add/remove object wrt a proposal
- Implement update proposal
- Implement create proposal
- Implement delete proposal
- Add UI controls for searching and filtering of proposals
- Create UI components for adding multiple objects to a proposal
- Update UI component for proposal view
- Create UI components for proposal lists
- Normalize proposals data from the backend
- Normalize topics data from the backend