Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Civil and Environmental Engineering

Synne Midori Ota Fog

Application for development of cartographic surveys

Master's thesis in Engineering and ICT
Supervisor: Terje Midtbø
June 2019

NTNU
Norwegian University of
Science and Technology

Synne Midori Ota Fog

Application for development of cartographic surveys

Master's thesis in Engineering and ICT
Supervisor: Terje Midtbø
June 2019

Norwegian University of Science and Technology
Faculty of Engineering
Department of Civil and Environmental Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

Online surveys are frequently used to collect information from a wide range of participants. This approach is also used in relation to geospatial data surveys. However, it is difficult to find an existing application that allows for implementation of geospatial data visualization, customization of the data and of the map used.
The specialization project "Creating web-based map experiments - methods and technology" established similarities between the variables and functionality used by surveys using geospatial data. These similarities were used as the foundation for creating an application with the purpose of quick and simple geospatial survey creation. This thesis describes the decisions behind the development of the application, as well as the final application structure.

After development, a usability test of the system were conducted. The results of the test conclude that the participants found the application well integrated and understandable even for inexperienced users. Still, it was concluded that it would be beneficial to improve the design of the application in order to make it more user-friendly and easier to navigate. The participants appreciated the functionality available and generally found the application useful. Future work includes adding more customization and map functionality, as well as developing an improved system for saving and visualizing the survey results.

Sammendrag

Spørreundersøkelser på nett blir ofte brukt til å samle inn informasjon fra et vidt spekter av deltakere. Denne tilnærmingen blir også brukt i sammenheng med geomatikk undersøkelser. Likevel er det vanskelig å finne eksisterende applikasjoner som tillater visualisering av kartdata, som tillater brukerstyrt tilpasning av data og valg av kart for bruk i undersøkelsen.
Prosjektoppgaven "Creating web-based map experiments - methods and technology" fant likheter i variabler og funksjonalitet benyttet av spørreundersøkelser som tar i bruk geografiske data. Med utgangspunkt i disse likhetene er det i denne masteroppgaven blitt utviklet en applikasjon med hovedfokus på å raskt og enkelt kunne sette opp en geomatikk basert spørreundersøkelse. Denne masteroppgaven tar for seg beslutningene som ble tatt under utviklingen av applikasjonen og den endelige applikasjonsoppbygningen.

Etter produksjon ble det utført en brukervennlighetstest av systemet. Resultatene fra testen konkluderer med at deltakerne syntes applikasjonen var godt integrert og forståelig selv for uerfarne brukere. Det ble likevel konkludert med at det kan være hensiktsmessig å forbedre designet av applikasjonen for å lage den mer brukervennlig og enklere å navigere. Deltakerne satte pris på funksjonaliteten applikasjonen tilbyr og fant den generelt nyttig.
Videre arbeid inkluderer implementering av flere avanserte tilpasningsmuligheter og flere kartverktøy, i tillegg til å utvikle et forbedret system for lagring og visualisering av resultatene fra undersøkelsene.

Preface

This master thesis is written for the Department of Civil and Environmental engineering at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway. It is a part of the study program Engineering and ICT - Geomatics and was written in the spring of 2019.

Firstly, I would like to thank my supervisor Terje Midtbø for his feedback. I would also like to thank everyone who participated in the usability- and pilot test their time and feedback, and my classmates for their continuous comments and advice. At last, I would like to thank my dad and Harry Knowles for their help with finalizing the thesis.

Trondheim, June 2017
Synne Midori Ota Fog

Task description

In many situations, it is interesting to study how visualizations in maps and map applications are interpreted by its users. One method used to collect feedback about the visualization is to create surveys. Usually, these surveys are created using a customized application based on the special needs of the specific survey. This is time consuming and it is therefore desirable with an application that makes it easier and faster to create surveys.

This master thesis will focus on developing a user-friendly application with which users can create a map survey fast and efficient. The thesis builds on the work done in the specialization project "Creating web-based map experiments - methods and technology" written by Fog in the course TBA4560 autumn of 2018 and will be using results from this literature study to integrate tools and functionality specifically aimed at map surveys.

This thesis shall:

- Develop an application that makes it easier to create map surveys
- Implement geomatics specific functionality in the application, based on the findings in the specialization project carried out in TBA4560
- Look at how the application implements the requirements set and what tools are implemented to help the users create a better survey
- Test the usability of the application
- Use the results from the user test to identify issues with the application
- Discuss future work that can be done with the application to improve issues discovered in the user test and to offer users more functionality

Contents

List of Figures

List of Tables

List of Abbreviations

ISO - The International Organization for Standardization

MEAN - Mongodb, Express, Angular, Nodejs

JSON - JaveScript object notation

NPM - Node package manager

SUS - System usability scale

NPS - Net promoter score

1 | Introduction

Surveys are a useful tool for measuring user interpretations of maps and trying out new techniques. Of late, surveys have more frequently been made online. This is because web-based surveys allow for broader participation outreach, quickly and with low effort. It is, however, challenging to find applications that allow for the implementation of a survey combined with visualization of geospatial data.

In the specialization project "Creating web-based experiments - methods and technology", Fog [2018] compared existing map surveys to delineate whether it is possible to create a common base for implementing map surveys. The project revealed that surveys using geospatial data have several similarities regarding what variables they wish to collect and what map functionality they use, as well as possible tools that can be implemented to help the users create better surveys.
This master thesis will develop an application that allows users to create surveys using geospatial data, based on the needs discovered in the specialization project.

Rather than implementing many and advanced functions, the focus of the application will be on basic functionality, alongside making the application as a whole as user-friendly and intuitive as possible.
In addition, some tools for guiding users to create better surveys will be implemented. The following is a list of tools discussed by Fog [2018], that will be considered for implementation in this application:

- Independent variables: registration page with default questions to capture the most commonly used independent variables in map surveys
- Timing: offer users the choice to record the time participants spend on the survey
- Controlled random question order: offer users the opportunity to randomize the order of the survey pages. The page order is controlled by the application, so that there is an even distribution of the page order shown to participants

- Progress bar: users can choose to display a progress bar to the participants in order for them to track their progress

Some aspects that will not be focused on in this iteration of the application include the format of the results collected from the surveys, adapting the application to various screen sizes and devices, creating separate users and adding map animations.

Map animations will not be considered even though it is used in several of the surveys in the literature study. This is because animations are considered more advanced to implement, whereas the focus of this first iteration is to set the groundwork for the application and establish if it is a tool people want to use.
Even though usability is an important part of this iteration, the focus will mainly be on the offered functionality, if the users understand how these work and if they find them useful. These points can be tested without being logged in as a specific user or adjusting the application to different screen sizes or devices, as the tests will be conducted on the same computer as the development. It is therefore decided to implement these requirements at a later point in time.
The results from the surveys will be collected and presented to the user. However, as result analysis and result presentation were not a main part of the literature study, the collection and visualization will not be a primary focus of this thesis either. As it requires a deeper understanding of statistical methods the users might want to perform and what data format will be optimal in order to perform these.

1.1 Paper outline

The first section of this paper will discuss what usability is, and why it is important to have a system that is easily grasped by the users.

Furthermore, the paper will analyse which technologies are used to develop the application, how they work together and why they were chosen, before describing the system requirements and the system design. The system design will describe the database design choices and the front end components used to construct the application. This section will also discuss how the system requirements are fulfilled, as well as presenting other tools that are implemented to subtly help users create better surveys.

Thirdly, the paper will describe usability testing, the purpose and structure of the usability test performed to test the application, as well as the execution of the pilot test and the test itself.

The final section of the paper will present and discuss the results of the usability test, and discuss possible future work.

2 | Usability

Usability is defined by the International Organization for Standardization (ISO) as "the extent to which a system, product or service can be used by specified users to achieve specific goals with effectiveness, efficiency and satisfaction in a specified context of use" [DIS, 2009].

Additionally, ISO states that a user-friendly system increases the productivity of users, as well as the operational efficiency of organizations because the systems are easy to understand and use. This results in reduced training and support costs, increasing usability for people with a wide range of capabilities and therefore increasing accessibility of the system. A user-friendly system will also reduce discomfort and stress as well as improve the user experience. A good system makes the users capable of understanding and using the product without additional assistance.

Nielsen [1994] states that "user interfaces are now a much more important part of computers than they used to be. The revolution in personal computers and falling hardware prices are making computers available to ever broader groups of users, and these users are using computers for a large variety of tasks. When computers were only used by a small number of people who mostly performed very specialized tasks, it made sense to require a high degree of learning and expertise of the user".

With this in mind, the application developed will focus on having an understandable and user-friendly interface and functions. As the target group for the application is not strictly limited to specialists, the application should be made general and comprehensible for everyone. Instead of requiring the users to learn a set of skills, vocabulary or memorize a sequence of commands in order to make use of the desired functions.
To make up for differences in understanding of geospatial terms and technological intuition, the application will distribute information to the users by using tooltips (a message which appears when a cursor is positioned over an icon, image, hyperlink, or another element in an application) and informative texts.

After development, usability tests will be performed to determine whether the information in the application itself is sufficient for independent understanding.

3 | Technology

This chapter gives a short description of the technology used to create the application and discusses the reasoning behind these particular choices.

3.1 The MEAN stack

The application is created using a MEAN stack (MongoDB, Express, Angular, Node) and is hosted on Heroku.com.

As discussed by Fog [2018], it was discovered that it would be desirable to use a noSQL database and a JavaScript frontend framework to create the application. Based on this the noSQL database MongoDB and the frontend framework Angular were chosen for the development. To make communication between these two technologies easier, it was decided to use the MEAN stack for the project development.
The MEAN stack is a full stack that covers the database, frontend- and backend framework needed in the project. The communication flow is as visualized in figure 3.1.

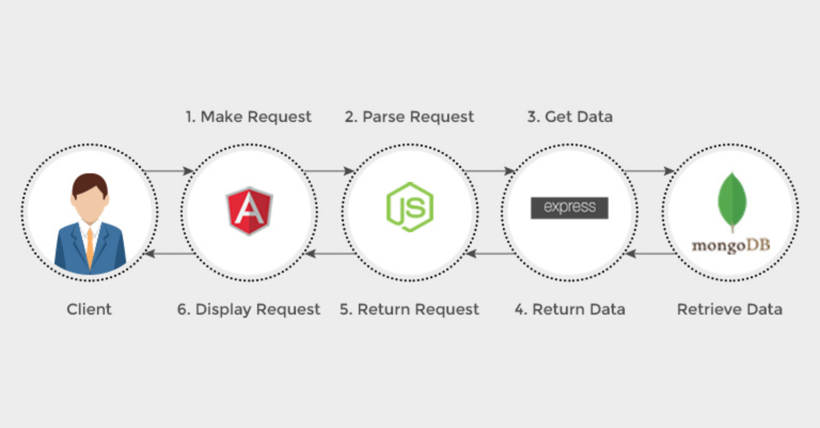


Figure 3.1: The MEAN stack communication flow

Angular.js is an open-source framework for building applications inside the browser [Dickey, 2014], and is a good alternative for highly interactive applications. The framework allows for reuse of components created in the application. In addition, it comes with many ready built components that can be accessed using packages like Angular material or bootstrap. In the MEAN stack, most of the applications logic is executed in Angular.

Node.js is a server-side framework. In the MEAN stack, node receives database requests from Angular in the form of Ajax requests. Node then uses the built-in node package manager (NPM) **Express** to simplify and send the request to the database, MongoDB, as a JSON object (JavaScript object notation). The data is then returned to Angular through the stack, as a JSON object.

MongoDB is a NoSQL database. These can store data that does not have to follow a strict schema and are therefore more flexible than the SQL databases that require strict templates for the data they contain [Buckler, 2015]. In addition, NoSQL databases allow for changing the accepted data inputs at any point in time, without making changes to already existing data file. This, and the fact that NoSQl databases can store data in multiple formats, like GeoJSON and images, were the main reasons why a NoSQL database was chosen for the project.

3.2 Leaflet

Leaflet JS is an open-source JavaScript library that will be used for the interactive map in the application. The core library is lightweight, created to be easy to understand and easy to implement basic functionality.
The library was selected because of its simplicity, and at the same time being capable of implementing all the map functionality listed in the requirements, section 4.2.

4 | Requirements

This chapter presents the requirements for the application.
They are presented in three groups; "must have", "should have" and "may have". "Must have" contains the survey functionality that is essential for a regular working survey, as well as the minimum map functionality needed for the application to handle basic map surveys. "Should have" contains features that are not crucial for the application to function, however, these should be implemented to offer a better user experience. "Can have" contains features that would be preferable to have but are not prioritized, for example, the tools described in chapter 1.

4.1 Basic survey functionality

Must have

- For survey, page, question, map and map data: Be able to create a new element and delete elements
- The information in the survey elements must be kept synchronized with each other
- The user should be allowed to add multiple types of questions (freetext, multiple choice, checkbox)
- The user should be able to add multiple kinds of pages to the survey (questions, map and question, map where the participants can place a marker)
- The user should only be able to edit the survey if coming from the main page
- The application should be able to save results from the surveys
- The user should be able to view the survey results
- Easy navigation between pages, both for users and participants
- Users should be able to navigate back to the survey overview at all time

- Users should be able to get a shareable participant link
- There should be a participant view that hides all editing options and saves the results

Should have

- For survey, page, question, map and map data: The opportunity to edit information and have automatic validation
- The user should be able to delete all the results of a survey from the database
- The survey should save independent variables in a way such that they easily can be used for future analysis

May have

- Users should be able to change the order of pages, questions and alternatives

4.2 Map functionality

Must have

- The user should be able to set start location, zoom level, min and max zoom
- The user should be able to create a static map (no zooming or dragging of the map)
- The user should be able to change the basemap
- The user should be able to upload their own map data and save this in separate map layers
- The user should be able to remove map layers
- The user should be able to add different types of map layers (polygon, line, circle, marker)

Should have

- The user should be able to decide if participants can click in the map, and save that location to the results file
- Users should be able to set the default map values in the map itself, not in the editing form

- Users should have the opportunity to choose what layers to view

May have

- Users should have the opportunity to use their own basemaps. Created in tools like cartodb or mapbox
- Users should be able to customize layers
- Users should be able to customize markers
- Users should be able to draw in the map and save the data to a layer

4.3 Tools

Should have

- The surveys should be created with a default registration page with independent variables specific to map surveys
- The application should have a confirmation pop-up before deleting any content
- The user should be able to choose to display a progress bar to the participants
- The user should be able to choose to time how long the participants spend on the survey
- The user should be able to choose to use controlled random order of the questions in order to vary the sequence the pages are shown in

May have

- The user should be able to decide the background color of the survey

5 | The application

This chapter will look at the design decisions made when creating the application, describe the application and its functionality. The chapter will also discuss shortcomings of the application with regards to the system requirements and problems that resulted in large changes compared to the original design.

Target group
The application is designed for anyone with the need for creating surveys with geospatial data. It should not be a requirement that the users are professionals within the geomatics field, that they have specific technical skills or that they are within a certain age group.

The application can be found here: http://geo-tests.herokuapp.com/survey
And the source code is available at: https://github.com/smfo/geoApp

5.1 The database

The database chosen for the application is MongoDB. This is a document-oriented database, which means that instead of dividing the data into tables as in SQL databases, the data is saved in documents [Buckler, 2015]. The architecture of a document-oriented database is as visualized in figure 5.1. The database consists of several collections, which group the data. When requests are made to the database, they are sent to a specific collection. Therefore, the data in a collection is often related and can be compared to a table in SQL.
Each collection contains documents. These are the files where the actual data is stored. They can be compared to a tuple, or a row, in a SQL table. The difference is that documents do not have to contain the same attributes, they can be widely differently structured and still belong to the same collection.

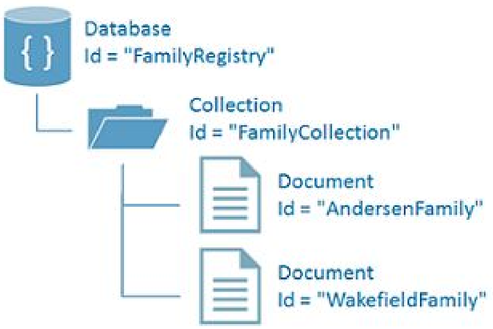
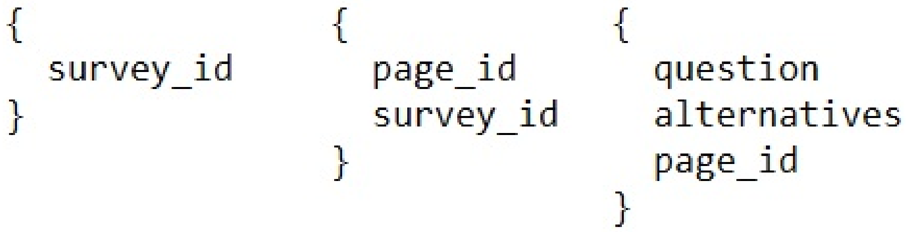


Figure 5.1: Architecture of a document-oriented database

5.1.1 Possible approaches to database design

There are many possible approaches to take regarding how to design the database for the application. This section will explore the two extremes: to split up the survey data as much as possible or to save all the data belonging to one survey in the same document.

The first possible approach would be to design the database with a normalized design, where the different types of data are divided into multiple collections and connected by the child data holding the id of the parent data. A possible design can be seen in figure 5.2.



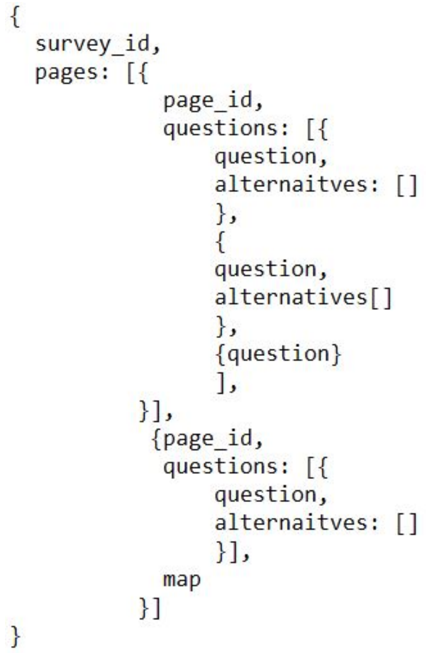
```
{                  {                {
  survey_id          page_id          question
}                    survey_id        alternatives
                   }                  page_id
                                    }
```

Figure 5.2: Example of a normalized design approach, where all the information is split up into separate collections

Because all the data is saved separately, the information in the different collections would have to be retrieved every time a change occurs. For example, when changing to a new page, the information about this page, it's questions and possible map information has to be retrieved from the database. This approach will lead to a lot of database requests that could make the application slow. It would also require some logic to collect new documents from the database using the right id's from the parent data.
Advantages of using this approach are that it would be easier to edit, delete

and add objects, as only one part of the data needs to be modified. This is as opposed to an entire survey object, as seen in the next approach.

The next possible approach is to denormalize the data and keep it all in one document, where the child data is saved in arrays. A possible design for this approach can be seen in figure 5.3.



```
{
  survey_id,
  pages: [{
            page_id,
            questions: [{
                question,
                alternaitves: []
                },
                {
                question,
                alternatives[]
                },
                {question}
                ],
         }],
          {page_id,
           questions: [{
                question,
                alternaitves: []
                }],
           map
         }]
}
```

Figure 5.3: Example of a denormalized design approach, where all the information is saved in one document

This approach will only require one database request to collect all the survey information, as all the needed information is saved in the same document. On the other hand, all data needs to be kept in memory as long as the survey is accessed. If there is a lot of map information in the survey, these files can be large and make the application slow.

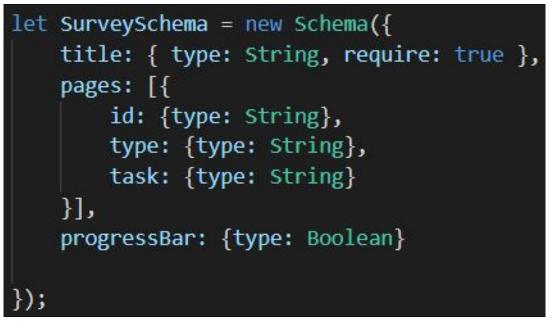
As there is no point in sending an update request to the database of the entire document every time a new change is made, this approach would require to hold an edited version of the survey object in memory. All the changes would be saved to this object, and the object would be posted to the database with all the new changes after editing of the survey is finished. The risk here is that if something goes wrong, all the changes done during the session will be lost.

This approach requires more complicated logic in frontend than the first one, as there is more data to keep track of and because there will be a lot of nested arrays containing objects.

5.1.2 The application database design

The desired design for the database is one that results in few requests, however, still maintains a reasonable complexity to the frontend logic.
An important factor to simplify the logic is to reduce the number of arrays needed. This is because arrays require the system to keep track of positions of each item at all times, as well as making forms and database requests more complex. It would be possible to get rid of all arrays by choosing the normalized approach discussed in 5.1.1. However, as noted there, this would lead to a lot of requests, as well as a larger number of collections in the database and more logic linked to the database communication.

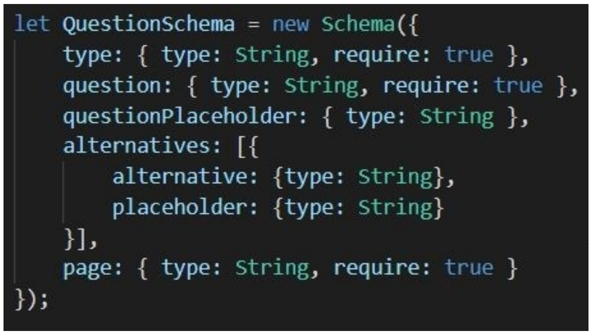
Based on this, the database design in the application is a mix between the two methods discussed in 5.1.1.
The schemas used have a larger survey document containing all the information about the survey object and all its pages. There are separate collections containing all the question documents, the map documents and the map layer documents.



```
let SurveySchema = new Schema({
    title: { type: String, require: true },
    pages: [{
        id: {type: String},
        type: {type: String},
        task: {type: String}
    }],
    progressBar: {type: Boolean}

});
```

Figure 5.4: The schema for the Survey model

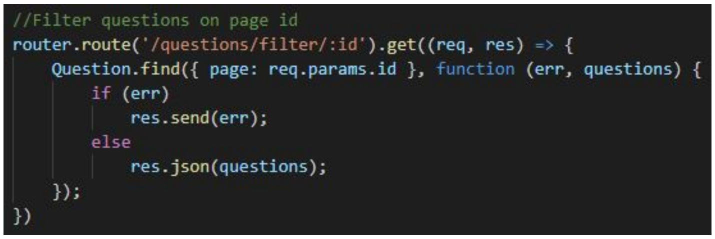


```
let QuestionSchema = new Schema({
    type: { type: String, require: true },
    question: { type: String, require: true },
    questionPlaceholder: { type: String },
    alternatives: [{
        alternative: {type: String},
        placeholder: {type: String}
    }],
    page: { type: String, require: true }
});
```

Figure 5.5: The schema for the Question model. The variable "page" contains the id of the survey page the question belongs to

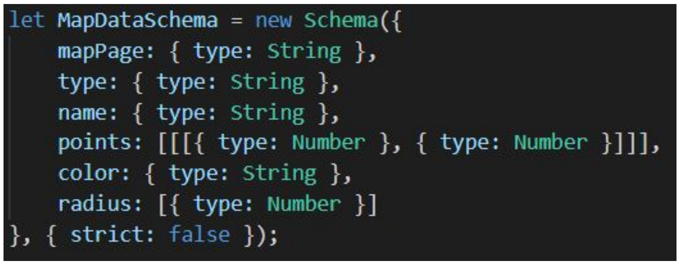
This design was mainly chosen to minimize the amount of logic in frontend. It was decided to keep the information that will most likely need to be changed frequently, the questions, the map settings and the map layers, in separate collections. This makes the documents easier to access and simplifies the logic when working with them in the frontend application, as there is no need for arrays.
The pages remain part of the larger survey object because these contain less information and will therefore not be edited as much. This considered, it was decided that editing the entire survey object when a page is edited, added or deleted, is manageable. Both in regards to the logic complexity and the amount of data kept in memory.

The different collections are linked together using the page id's. When a question or map document is created, the id of the page it belongs to is saved as the value "page", as seen in figure 5.5. When a page is opened, the relevant collections are filtered to get all the information relevant to this page.



```
//Filter questions on page id
router.route('/questions/filter/:id').get((req, res) => {
    Question.find({ page: req.params.id }, function (err, questions) {
        if (err)
            res.send(err);
        else
            res.json(questions);
    });
})
```

Figure 5.6: The query used to get all questions with a certain page value



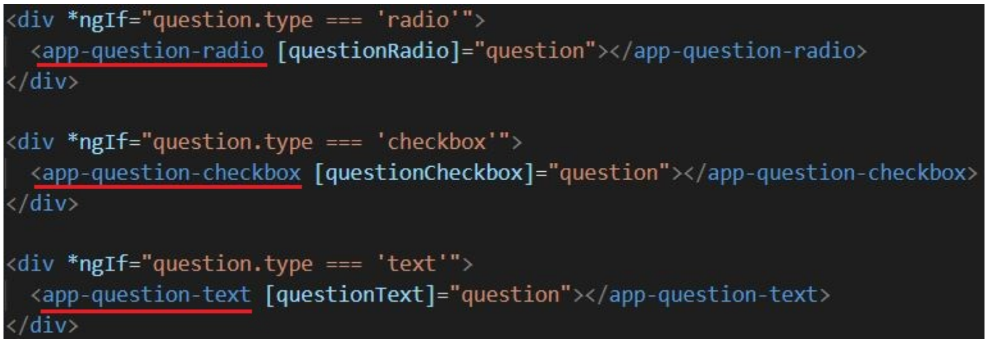
```
let MapDataSchema = new Schema({
    mapPage: { type: String },
    type: { type: String },
    name: { type: String },
    points: [[[{ type: Number }, { type: Number }]]],
    color: { type: String },
    radius: [{ type: Number }]
}, { strict: false });
```

Figure 5.7: The schema for the map layers. The "mapPage" field holds the id of the map the layer belongs to, and works the same way as the "page" field in the Question schema, figure 5.5

5.2 The application components

This section will talk about how the application fulfills the set requirements by describing the structure of the Angular application and the different components created.

The application is built using angular components. An Angular component is a class that contains data, logic, a HTML template and CSS [Ang]. The components can be placed anywhere in the application by using the component name in the same way as a regular HTML tag, as demonstrated in figure 5.8. The result of this is that reuse of components is easy, as only one line of code needs to be edited in order to add or delete a component.



```
<div *ngIf="question.type === 'radio'">
  <app-question-radio [questionRadio]="question"></app-question-radio>
</div>

<div *ngIf="question.type === 'checkbox'">
  <app-question-checkbox [questionCheckbox]="question"></app-question-checkbox>
</div>

<div *ngIf="question.type === 'text'">
  <app-question-text [questionText]="question"></app-question-text>
</div>
```

Figure 5.8: An example of how the components "question-text, "question-radio" and "question-checkbox" is inserted in the application by using HTML

An overview of the components used in the application, with a brief explanation of what they do is displayed in table 5.1.

Component	**Description**
Header	Displays the title of the current survey
Main-Survey	Displays a list of all the surveys created. Also contains a "Create new survey" button
Navigation	Navigates between the pages in the survey
Survey-container	Works as a container for all the components used in the survey. Also contains options to navigate back to the main-survey component and open the editing components
Create/edit survey	Contains the form where the survey variables are set
Create/edit page	Contains the form where the page variables are set
Question-container	Works as a container for all the question components.
Create-question	Contains the form that allows the user to create questions
Question-text	This component is used to display the questions that allows the participants to reply with freetext
Question-radio	This component is used to display the questions that allows the participants to pick at most one answer alternative
Question-checkbox	This component is used to display the questions that allows the participants to pick multiple answer alternatives
Edit-question	Contains the form that allows the user to edit questions
Map-container	Works as a container for all the map components. Also holds the interactive Leaflet map
Edit-map	Contains the forms where the map variables are set
Map-data	Contains the forms where the map layer variables are set
Confirm	Contains the pop-up displayed when the application wants to verify a request made by the user. Ex. when deleting elements
Sharable-link	Contains the link the user can use to share their survey with the participants

Table 5.1: The components used in the application

5.2.1 Survey components

The survey components are the components that are needed for the basic structure of the application. These are the parts of the application that ensures that it functions as a proper, normal survey creating tool.

The main component in this part of the application is the survey-container component. This component is the access point for all the sub-components as visualized in figure 5.9.

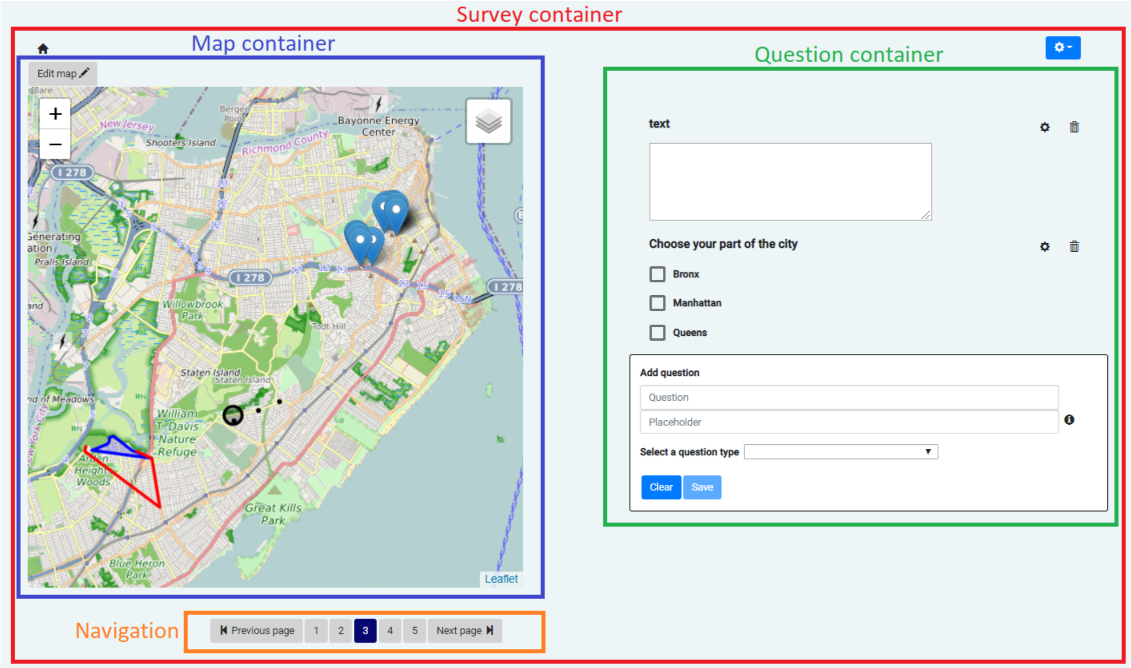
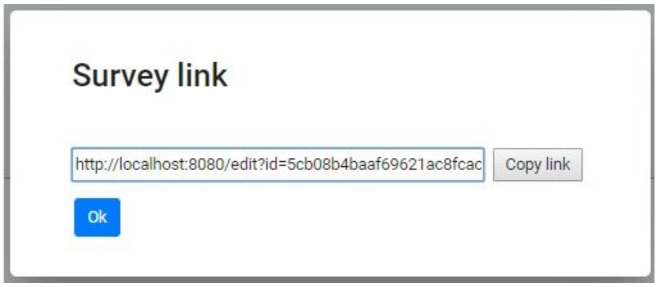


Figure 5.9: Main components in the survey component

Modals

The modals are responsible for all data that is saved, edited and deleted. In the application, these are viewed as popups containing information. The basic survey section of the application has four modals intended for editing of the following data: survey, page and question, as well as the modal provided to get a sharable link of the application.



Survey link

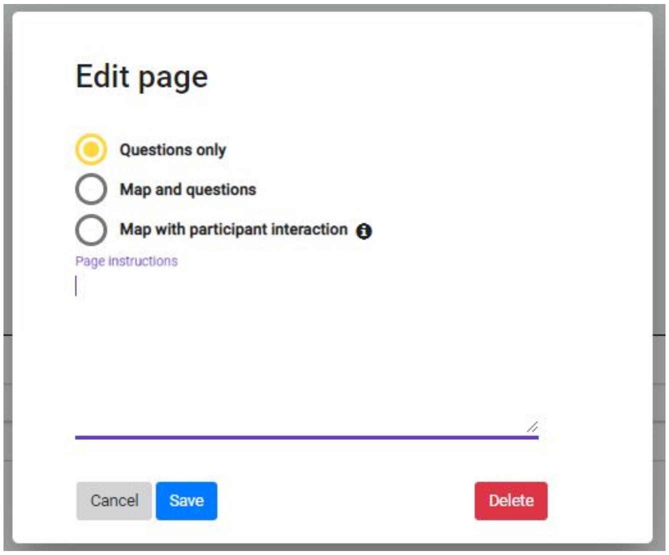
http://localhost:8080/edit?id=5cb08b4baaf69621ac8fcac

Figure 5.10: Sharable link modal

The first three modals all contain a form, where the data related to their component is displayed and can be edited by the user. The modals are responsible for validating the information submitted through their forms. This is to make sure there are no errors while saving this information to the database, or when uploading the information from the database to the application. Table 5.2 shows an overview over the different validators used in the modals.

Modal	**Validator type**	**Fields**
Survey	Required	Title
		Info
		Thank you message
Page	Required	Page type
Question	Required	Question text
		Question type

Table 5.2: Validators used in the basic survey components



Edit page

● Questions only

○ Map and questions

○ Map with participant interaction

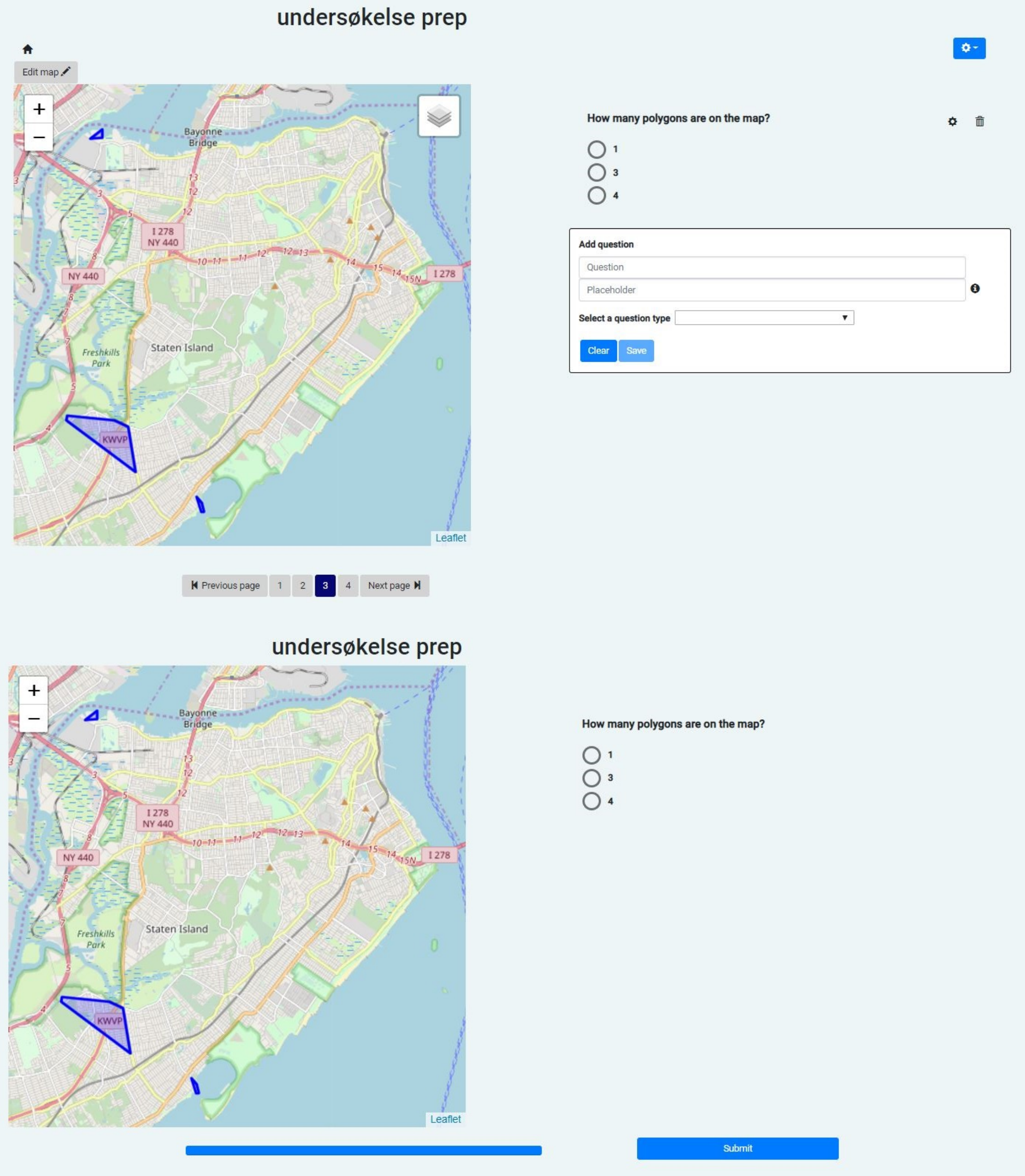
Page instructions

Figure 5.11: Edit page modal

Participant view

Most of the components in the application have two slightly different views. One meant for the user of the application and one meant for the participants answering the survey. This is done to hide editing possibilities from the participants, while still being able to use the same components throughout the application. Mostly the changes involve hiding buttons that gives access to modals, while some components, like create-question are completely hidden.

The navigation component is the only one that has a completely different visual component. The reason why there are not two separate components for navigation is that a lot of the logic is the same in both views, as well as being a quite small component, it was more beneficial to keep all the code in one place.



undersøkelse prep

How many polygons are on the map?

○ 1
○ 3
○ 4

Add question

Question

Placeholder

Select a question type

undersøkelse prep

How many polygons are on the map?

○ 1
○ 3
○ 4

Figure 5.12: Edit view (top) and participation view with progress bar (bottom)

Results

It is the participant view that saves the results after the survey is completed. Only questions that are answered will be saved, this prevents the database from containing a lot of empty replies.
The user can view the survey results by accessing "Display results" in settings. Here, they also have the option to delete all the saved results.

5.2.2 Map components

There are three map components in the application. These display the Leaflet map, holds all the data about the map, the data layers and the two modals that gather map data from the users.

The main component is the map-container. This is the component that displays the map, hold the information about the map settings and the data to be displayed. Visually, the component consists of two items, the edit map button, and the map itself, see figure 5.13. The map will display a chosen basemap, as well as any layers the user desires to add. At default, a zoom tool will be visible. This will be removed if the user choose to create a static map. After adding layers to the map, a layer controller will be visible as well. This allows the user to choose which layers to display in the map at any given time. The controller can be removed from the participant view if desired, it will however always be displayed in the regular view.

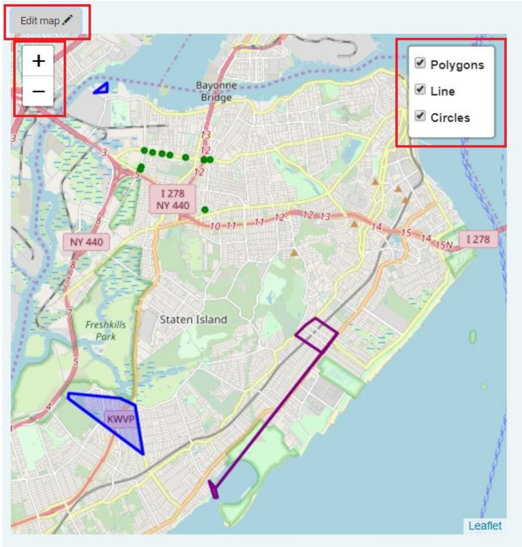
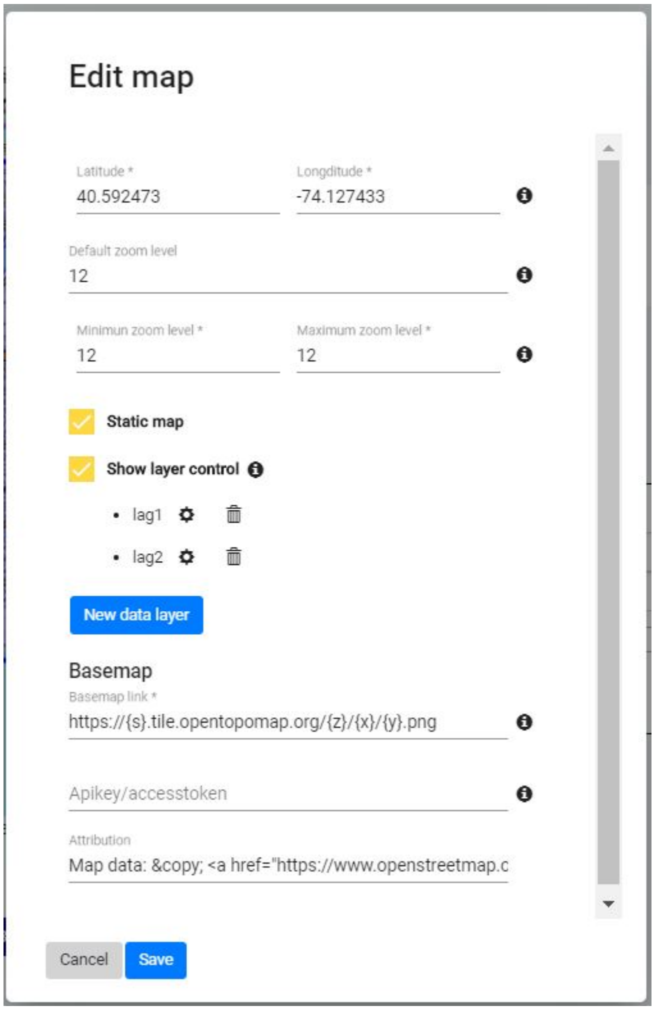


Figure 5.13: The map container contains the "edit map" button and the map itself. Here displayed with the zoom tool (top left corner) and a open layer controller (top right corner). Figure 5.12 displays a closed layer controller

The edit map button, takes the user to the edit map modal. This registers all the information related to the map settings, the basemap and what map tools to display to the user. It also holds a list over the map layers available, and the opportunity to delete these.



Edit map

Latitude * 40.592473

Longditude * -74.127433

Default zoom level 12

Minimun zoom level * 12

Maximum zoom level * 12

☑ Static map

☑ Show layer control

- lag1
- lag2

Basemap

Basemap link * https://{s}.tile.opentopomap.org/{z}/{x}/{y}.png

Apikey/accesstoken

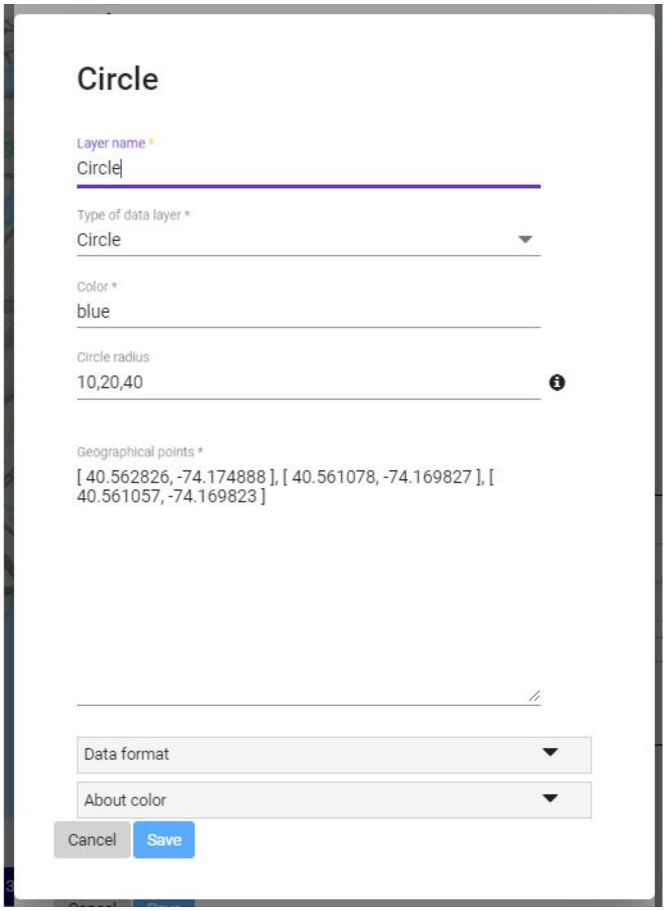
Attribution Map data: &copy; <a href="https://www.openstreetmap.c

Figure 5.14: The edit map modal

The data displayed in the map is added in the mapdata modal. The reason for its design is explained in section 5.6.1.

The component records the desired data to be saved to the map layers as well as customized properties affecting the visual display of these. There are four types of layers available: points, polygons, lines and circles. All types except points can be customized using color. Polygons and circles give the opportunity

to add multiple objects in the same layer, however the color will be the same for all objects within the same layer. Circle also gives the opportunity to set the circle radius. The radius can be the same for all the circles within the layer or specified for each one.
This modal require the user to know more about the allowed formats for color and which data format will be accepted for the different layer types. Because this require too much text for a information box, there are two expandable text-boxes at the bottom of the modal that contain the necessary information to record the data properly.



Circle

Layer name *
Circle

Type of data layer *
Circle

Color *
blue

Circle radius
10,20,40

Geographical points *
[40.562826, -74.174888], [40.561078, -74.169827], [40.561057, -74.169823]

Figure 5.15: Data layer modal containing a circle layer, where the circles have different radiuses

Table 5.3 shows an overview over the different validators used in the map modals.

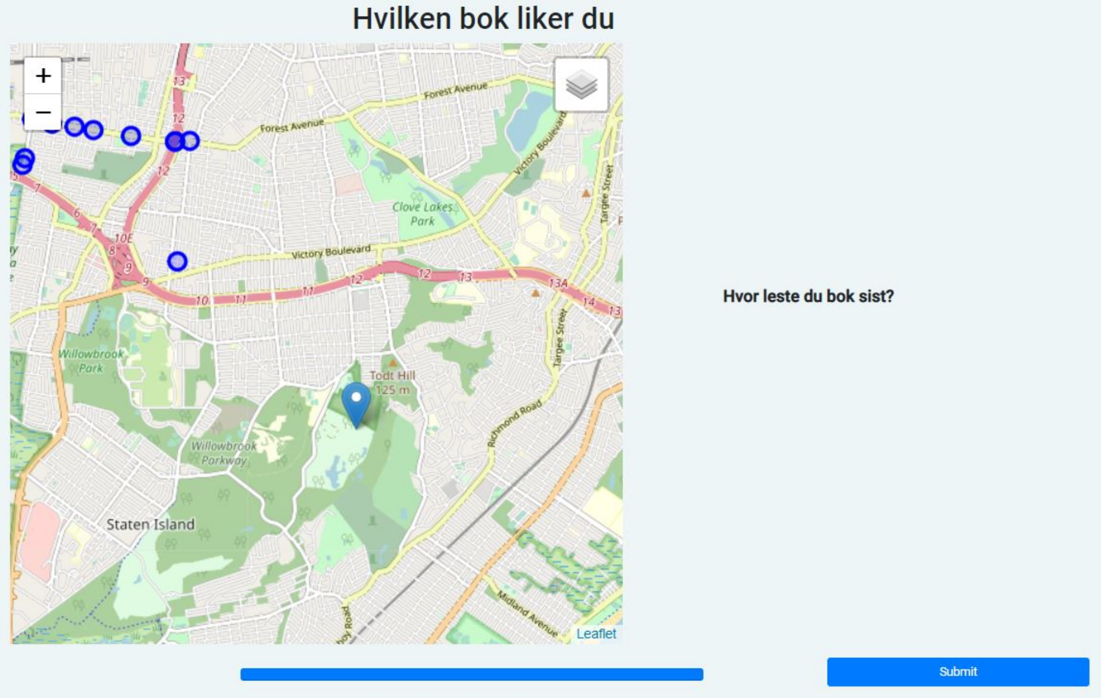
Modal	**Validator type**	**Fields**
Edit map	Required	Latitude and longitude
		Zoom
		Minimum and maximum zoom
		Basemap
	Check that full stop, not comma is used	Latitude and longitude
Data layer	Required	Name
		Data layer type
		Data points
		Color
		Radius
	Count equals 1 or the same as number of points	Radius

Table 5.3: Validators used in the map components

Map with participant interaction

There are two slightly different map pages to choose from: "map and questions" and "map with participant interaction". The "map and question" page displays the map-container on the left side and the question-container on the right side, as shown in figure 5.9. Using this page type will allow the participants to move in the map and change what layers to display, if the creator of the survey allows them to, however they are not able to add any geospatial data. This is because the map components and the question components does not communicate directly with one another.

The "map with participant interaction" allows the user to ask a question that requests the participants to mark a position in the map, and saves the question and the marked location to the results. This is possible because the map component in this page type takes up the entire page, and the question asked is a part of this component. Therefore, the question and the reply, consisting of a latitude longitude object, can be easily saved together.



Hvilken bok liker du

Hvor leste du bok sist?

Figure 5.16: A participant view of a "map with participation interaction" page

5.2.3 Tools

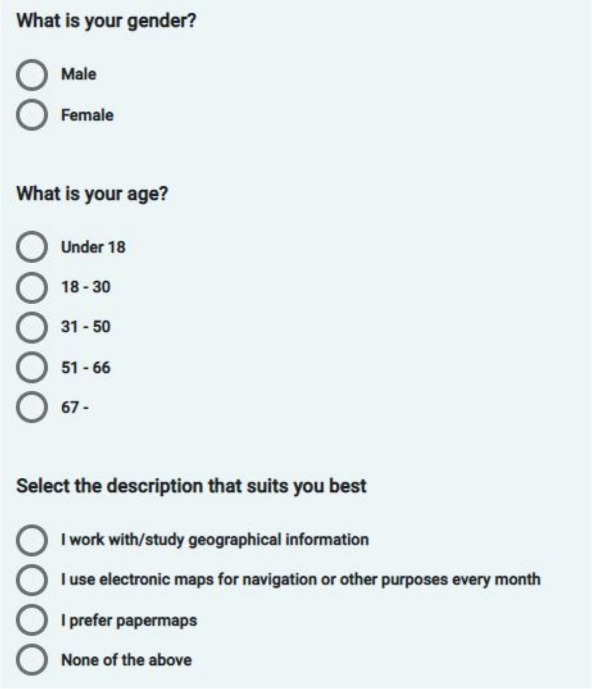
The tools implemented are functionality and information that is added throughout the application, in order to help the users create a better survey. Some of these are entire components, while others are part of a component or information that generate automatically.

Registration questions

Whenever a new survey is created, a registration page is automatically generated. This page contains questions to capture the independent variables found common to map surveys; age, gender and experience with geospatial data [Fog, 2018]. In the surveys used in the literature study, a common way to decide the experience level with geospatial data would be to ask the participants "are you experienced with map data". As this can be interpreted differently, the question generated in the application attempt to capture a more measurable value.

The alternatives in the age question were decided on the following premises: "Under 18" represents participants who are still in mandatory education. "18 - 30" represents participants that are of student age [Keute]. The two alternatives "31 - 50" and "51 - 66" represents participants who are working. And "67 - " represents participants who are retired pen. The registration questions are shown in figure 5.17.

The questions are automatically generated, however, they can be edited and deleted in the same way as all other questions should the user wish to.



What is your gender?

○ Male
○ Female

What is your age?

○ Under 18
○ 18 - 30
○ 31 - 50
○ 51 - 66
○ 67 -

Select the description that suits you best

○ I work with/study geographical information
○ I use electronic maps for navigation or other purposes every month
○ I prefer papermaps
○ None of the above

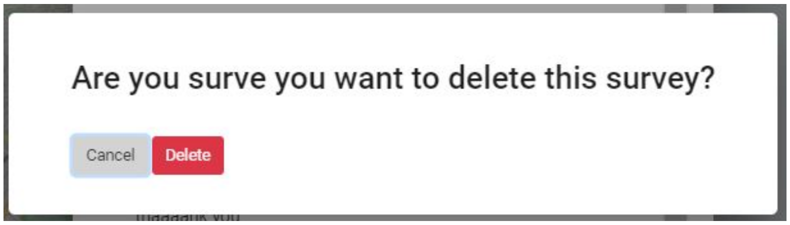
Figure 5.17: Registration questions

Progress bar

It is shown that letting the participant know where they are in the survey, might decrease the likelihood of them not completing it [Dillman et al., 1998]. Therefore, the application gives the user the possibility of displaying a progress bar to the participants. This can be decided in the survey modal. The progress bar is only visible in the participant view and is displayed in figure 5.12.

Confirmation modal

To prevent the users from accidentally deleting parts of the survey, a confirmation modal will be activated every time the user tries to delete saved data; a survey, page, question, map data or survey results. As well as when changing the page type leads to loss of data. For example, changing from a map and question page to just a question page, will delete all the saved map information.



Are you surve you want to delete this survey?

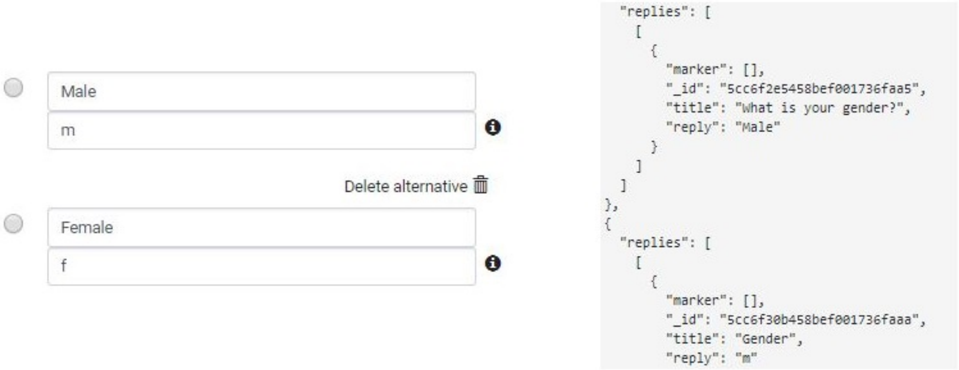
Figure 5.18: Delete modal displayed when deleting survey

Start- and end page

The survey has a mandatory start and end page. The start page is meant to give the participants a short introduction to what the survey is about and why it has been created. The end page is the page the participants are directed to after they have submitted their reply and is meant to contain a message to thank them for their participation.

Placeholder

The placeholder is a field available for simplifying the question and alternative text that is saved to the database. By default, the entire text visible to the participants will be saved as part of the results. However, if a placeholder is added this text will be saved instead. This will make the results easier to navigate.



○ Male
m

Delete alternative

○ Female
f

```
  "replies": [
    [
      {
        "marker": [],
        "_id": "5cc6f2e5458bef001736faa5",
        "title": "What is your gender?",
        "reply": "Male"
      }
    ]
  ]
},
{
  "replies": [
    [
      {
        "marker": [],
        "_id": "5cc6f30b458bef001736faaa",
        "title": "Gender",
        "reply": "m"
```

Figure 5.19: Illustrates question alternatives with a placeholder (left) and saved replies with and without placeholders (right)

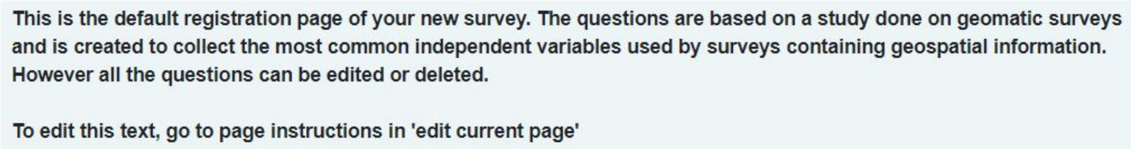
Information boxes and tooltips

Each input field and checkbox in the application has a tooltip with a short explanation of what its functionality is. The tooltip, a brief text description, appears when the mouse is placed above the element it belongs to.

Sometimes a field requires a longer explanation that is not fit to place in a tooltip. In these cases, the field also has an information box containing a more thorough explanation. The information boxes are accessed by clicking on the info-icon displayed next to the field.

Page instructions
The page instructions that can be added on each survey page allow the users to write short explanations of what they want the participants to do on this specific page. The result of this is that the user does not have to provide all the information necessary in the same place, at the start page. If all instructions are provided before the survey starts it is likely that the participants will forget them. It is more effective to provide instructions at the point where the participants may need to use them [Dillman et al., 1998].

Instructions to the users
In the same fashion the page instructions encourage the users to provide information to the participants at the point this information will be useful, the application provides instructions for the user when new functionality is introduced.
This is the case whenever a "map with participant interaction" is created and in the registration page. The texts explain parts of the functionality of the page, as well as informing the users where they can navigate to edit the text provided.



This is the default registration page of your new survey. The questions are based on a study done on geomatic surveys and is created to collect the most common independent variables used by surveys containing geospatial information. However all the questions can be edited or deleted.

To edit this text, go to page instructions in 'edit current page'

Figure 5.20: The user instructions on the registration page

5.3 Keeping components synchronized

An important aspect of the application is the ability to keep all the components up to date with one another. An example of when the components need to be kept synchronized is when one must certify that all the components hold information from the same survey page. This means that the questions, map settings and map layers viewed on the current page belong together. Another example is when a page or survey is deleted. In this case, all the information related to this element should also be deleted. This is to make sure that the database is not overflowing with unused information.

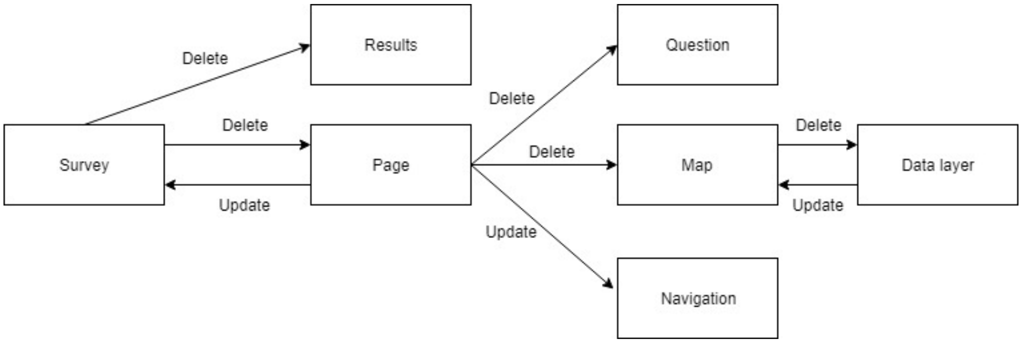


Figure 5.21: Illustrates how other components are affected is one component is deleted

One of the measures taken to make it easier to keep components synchronized was to create a global object of type GlobalVariables.
This is an object all the components can be given access to. It keeps information about the current survey, the logged in state and the current page. This is information that is relevant to multiple components. Whenever the object is updated, all affected components are made aware of the changes at the same time. This ensures that they always have access to the same information. After being notified that there is a change, each component makes the updates required of them. An example is when the current page is changed. When this happens the questions, the map and the map layers need to be updated. The navigation component also needs to change which page number to highlight.

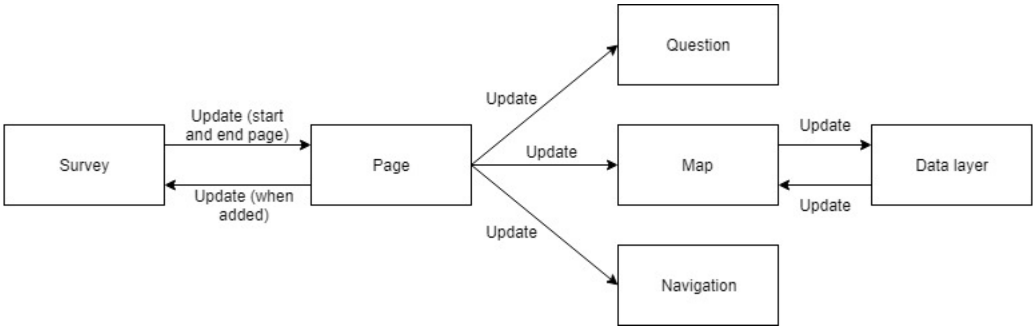


Figure 5.22: Illustrates how other components are affected is one component is changed

5.4 Design decisions

Access to components that will be edited frequently
As stated in section 5.1.2, the data that is most likely to change frequently are

the questions, the map settings and the map layers. The user will have the opportunity to add multiple questions and map layers at each page. Therefore, it was decided to design these components such that adding, editing or deleting them would be as efficient as possible.
To make it more efficient to delete multiple elements, the delete option is placed outside of the editing modal. This prevents the user from having to open the editing modal before deleting a layer or question.
For the same reason, to prevent the user from having to open a modal, the create question component is implemented directly in the question-container. This approach has not been taken regarding the map layer modal. The reason being that the design was not desirable given the limited space in the 'edit map' modal.

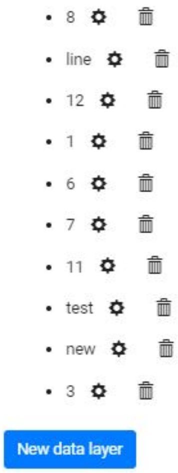


Figure 5.23: Illustrates the list of map layers listed inside the edit map component, and how it gives easy access to delete multiple layers with few clicks

Consistency: modals
All editing and adding options, except for the create question component, are displayed using modals. This is to keep the design of the application consistent and so the users are not required to figure out a new design every time they want to create a new component. To emphasize this, the modals are also kept as similar as possible. They have the same button placement, button design and button colors, and the input fields have the same design.

Simplicity
The application was designed to look as clean and simplistic as possible. This decision was made in order to not overwhelm the users and to make it easy for them to find the available functionality. To emphasize this there are few buttons and hidden menus available. Apart from the question components and

navigation, there are at most two buttons available at any time, the settings drop-down menu and the "edit map" button.
Most functionality either distinctly belongs to the map component or the overall setting of the page making it easy for the user to determine which menu they are placed in. There are some exceptions however, these are described in section 5.2.3 - "Instructions to the users".

Pages
By looking at the research papers used in the literature study by Fog, we see that one of the common traits in these map surveys is that the participants perform a set of similar tasks. To prevent the participants from learning from previous tasks and hence get a misleading result that suggests the later tasks are easier than the previous ones, the order of the tasks shown is rotated [Erichsen, 2017]. If all the tasks in the survey were to be displayed on the same page, accessible to the participants through scrolling, there would be no option to control if the participants navigate back to previous tasks to correct their answer once they are more used to how the survey works. Therefore, to be able to prevent correction of previous answers, it was decided to implement pages in the application.

5.5 Requirements that were not implemented

The following is a list of requirements from section 4 that were not implemented in the application.

- The test should save independent variables in a way such that they easily can be used for future analysis
- Users should be able to change the order of pages, questions and alternatives
- Users should be able to set the default map values in the map itself, not in the editing form
- Users should have the opportunity to use their own basemaps. Created in tools like cartodb or mapbox
- Users should be able to customize markers
- Users should be able to draw in the map and save the data to a layer
- The user should be able to choose to time how long the participants spend on the survey
- The user should be able to choose to use controlled random order in order to vary the sequence the pages are shown in
- The user should be able to decide the background color of the survey

These are all requirements that are classified as "Should have" or "May have", in other words, they are not essential for the application to function.
All the "should have" requirements hold varying weight towards the overall efficacy of the system. As such, the requirements in the list above have been neglected.
Some of them have not been prioritized because they could, to a degree, be fulfilled by using tools already implemented. For example, points can to a degree be customized by using circles instead of markers, which allow for customization through colors.

5.6 Problems/complications

5.6.1 Upload file to database

Saving files from the users computer to the database

The idea for adding customized data to the application was to let the user be able to upload files directly from their computer. While implementing the application, it turned out that saving a file from the computer to the database was difficult. Instead, it was decided to create a text area where the user could post the file content as text, like in any other part of the application. This text will then be saved to the database in GeoJSON format.

The reason this approach was chosen is that it was simple to implement and similar to how other information in the application is saved.
Being able to upload customized data is, as stated in section 4, one of the "must have" features of the application. It was also necessary to have this function working in order to implement many of the other "must have" features regarding map functionality, making it essential for the application. It is less convenient for the user, as they must open the file, copy its content and post it in a form in the required format. However, because of the time limitations of the project, it was viewed as more important to make this functionality work, rather than to implement it in the most user-friendly way possible.

The consequences of this for the application as a system is the need for another modal component. In addition, the edit-map component needed to be tied to this modal.
The modal for adding map layers is described more thoroughly in section 5.2.2.

6 | Testing

This section will look at how to prepare a usability test for the application. The goals of the test will be discussed, as well as what functionality is tested and why. Furthermore, the execution of the test will be described, as well as a summary of the pilot test and the test itself.

6.1 Usability test

The application will be evaluated under controlled settings, as this is a typical method to use for evaluation interfaces [Rogers et al., 2011b]. These tests are often conducted in a laboratory, as these conditions provide the necessary control to reduce outside influences and distractions, as well as systematically investigate whether all the set requirements are met. Usability tests in controlled settings can be conducted at any point in the development, what is tested vary from low-tech prototypes to complete systems.

The primary goal of usability tests are typically to determine whether the interface is usable by the intended target group to carry out the tasks which the application was designed for. This is customarily done by investigating how users perform on typical tasks, by comparing the number and kinds of errors the users make. It is also normal to record the amount of time the users spend performing the given tasks.
After the test is completed, user satisfaction questionnaires or interviews can be used to acquire a better picture of the user's opinions, with regard to their experience with the system.

6.2 The test

The reason the application is tested by participants as well as the developers, is to make sure that the application is understandable for the intended target group, and to make sure that the system works as a whole.

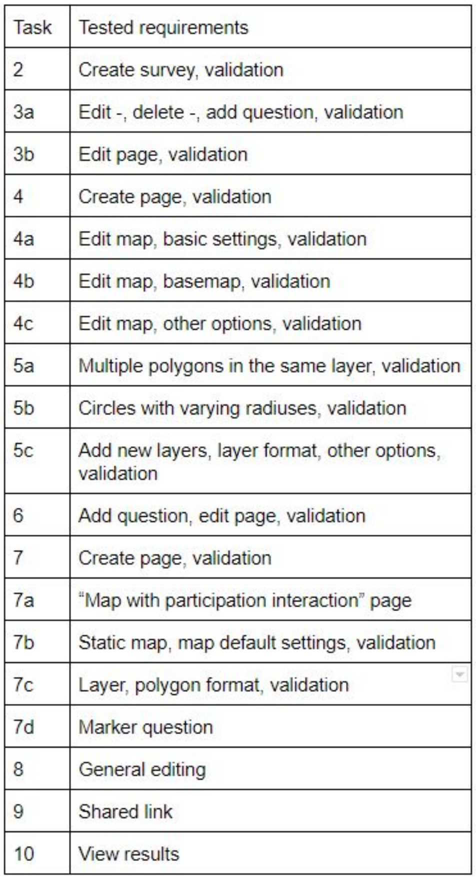
6.2.1 Goals

The goals of the test are

- To see whether the users find the application understandable and intuitive
- In the case that they do not find the application intuitive, are the users able to find enough information within the application to learn how to perform the tasks anyway
- Identify problems the users have with the application. Both design and functionality wise
- Determine if the design helps the users understand and navigate in the application
- The System usability scale score of the system must be "Acceptable", see 7.1
- It is preferable if the System usability scale score of the system is "Promoter", see 7.1

6.2.2 Test design

The primary focus of the test is the use of the application. The participants are given the task of creating a survey using the application. They have a large amount of freedom as to what they want to create. There are, however, a set of instructions listing features they must try as well as data types they have to include. These requirements were designed to touch on as many functionality requirements as possible, with emphasis on the requirements that tests the users understanding of the application. For example "The user should only be able to access the edit page via the main page", was not purposely tested.
Table 6.1 gives an overview of which functionalities were tested in each task.



Task	Tested requirements
2	Create survey, validation
3a	Edit -, delete -, add question, validation
3b	Edit page, validation
4	Create page, validation
4a	Edit map, basic settings, validation
4b	Edit map, basemap, validation
4c	Edit map, other options, validation
5a	Multiple polygons in the same layer, validation
5b	Circles with varying radiuses, validation
5c	Add new layers, layer format, other options, validation
6	Add question, edit page, validation
7	Create page, validation
7a	"Map with participation interaction" page
7b	Static map, map default settings, validation
7c	Layer, polygon format, validation
7d	Marker question
8	General editing
9	Shared link
10	View results

Figure 6.1: Overview of what application functionality is being tested

It was assumed before starting testing, that participants might have difficulty adding multiple polygons in the same data layer or adding a circle layer where the circles have different radiuses. Therefore, all the participants were specifically asked to execute these tasks.

After creating their own survey using the application, the participants were asked to take a System usability scale (SUS) test. This aims to return a general, overall usability score of the application.
The participants were then asked to answer a feedback survey specifically regarding the application. Determining which parts of the application they had difficulty with and if multiple participants had the same difficulties.

There are five independent variables collected in the feedback survey. These are: age, gender, experience with survey creation, experience with geospatial

data and experience with technology.
The ‘experience with survey creation variable’ is used alongside the feedback to determine whether the application and the functionality it offers are regarded as adequate even for experienced survey creators.
The ‘experience with geospatial data variable’ is chosen to determine if the application is accessible to all users, regardless of experience and knowledge of geomatics terms. It is expected that participants with less experience require more help while using the application and it will be noted if the tooltips and information boxes give them sufficient information.
The ‘experience with technology variable’ is used to determine how quickly the participants are expected to learn to use the application, and if the application is understandable for inexperienced users. Participants with a high level of expertise in this field are expected to learn the application quickly and to have insight into the design and implementation of user-friendly systems.

The instructions given to the participants and the results from the feedback survey are available in appendix A and C.
The full feedback survey can be accessed at:
http://geo-tests.herokuapp.com/edit?id=5cc406af556ad00017976696

6.2.3 Execution

The test was executed on a Lenovo Yoga laptop with a 13.3" screen. The participant completed the test on this, while watched by an evaluator. The evaluator was to observe what the participants did during the test, what they struggled with and explain confusion regarding formulation of the questions in the test instructions.

Before starting the test, each participant was given a short introduction to the purpose of the application. Some of the participants were not familiar with geomatics vocabulary, however, because one of the elements being tested was if the participants were able to understand how different application functionality worked without help from another user or a manual, these terms were not explained.

The participants were told to think out loud, and first look for explanations within the applications if they struggled to execute a task. If they struggled to understand what the task was asking, this was explained by the evaluator. However, if they struggled with how to perform what the task was asking, they did not get any help. If the participant, after reading all the available tooltips and helptext in the application still did not understand how to execute the task, it was pointed out which information boxes they should read. If, after reading these, they did still not understand how to execute a task, the task was skipped.

6.3 Pilot test

"A pilot test is a small trial run of the main study, to make sure the proposed method is viable before embarking on the real study" Rogers et al. [2011a].
The pilot test is supposed to make sure the equipment works as it is supposed to, if instructions for the participants are understandable, as well as confirm that the chosen procedure is viable. Executing a pilot test gives the opportunity to identify problems and correct them in advance of the main study.

Results from the pilot test for this application were used to improve the usability test instructions, discovering bugs in the application, as well as making changes to the application based on the suggestions of the participants of the pilot test. The main changes were clarifying content in the information boxes, adding new information boxes in the application and modifying the usability test instructions. Minor design changes were also made.

6.3.1 Execution

The execution of the pilot test was the same as the usability test described in section 6.2.3. The participants first used the developed application to create their own survey, before answering the SUS test and the specialized feedback survey for the application. The participants were asked to take particular notice of unclear formulations or information in both the test instructions and throughout the application.

The pilot test was conducted with four participants, all inexperienced with geospatial data and with little experience regarding survey creation. Two of the participants were in the age range 18 - 30, while the other two were in the range 51 - 67. The number of participants was chosen based on Rubin [1994] (p. 128) who states that "the latest research indicated that testing four or five participants will expose the vast majority of usability problems."
It is acknowledged that four participants is towards the lower side of the scale. For that reason, participants were chosen so that they would be part of the less experienced target group. As they had little previous experience with creating surveys and geospatial data, they would have to use the application itself to learn how to operate it. Instead of relying on previous experience.

6.3.2 Results and takeaways

Three of the SUS scores for the system were in the range 72.5 - 82.5. While the lowest was 55. Giving an average score of 72.5 out of 100. This places the application just over the average SUS score of 68 and just classifies it as "acceptable". A classification given to systems with a score of 71.1 or higher. This gives the indication that the application is usable, however, it can be greatly improved.

The participant with the lowest score stated that a large reason for the confusion was that the instructions and the application were in English. Without taking this score into consideration, the average is 78.33.

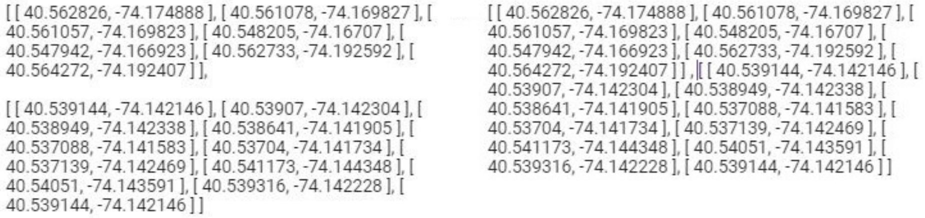
The pilot test revealed that some of the test instructions, as well as some of the information boxes in the application, were unclear. These were edited continuously while testing, to see if the other participants found them more understandable. Some additional information boxes were also added in the application, where the participants struggled to understand the purpose of different input fields.

The participants wanted more context about what they were doing for the different parts of the test. Therefore, some simple descriptive sentences were added to the instructions. For example "In this section, you will try to visualize geographical information in the map".
There were also some words that needed further explanation. These were mostly due to the language used and therefore a few words in the instructions have Norwegian translations.

It was discovered that the participants had trouble understanding the web page they were asked to go to in order to change the basemap. Therefore, a 30-second explanation of this was given to the participants of the finalized usability test. They were told that the web page contained basemaps that were available for everyone to use in their applications. They were also told that the right-hand side of the web page displayed these and that they were to pick one that covered the entire world. They were also shown where the information about the basemap was, but not what information to use to change the basemap in the application, as this was one of the tasks.
It was considered acceptable to give these instructions because this web page is not part of the application that was to be tested. However, it was still desirable to use it, to create a more realistic scenario than having a list of potential basemaps, telling the participants to choose one.

The tasks multiple participants struggled to understand were task 5a (adding one polygon layer containing multiple polygons) and task 5b (adding a circle layer containing circles with different radiuses).
Few of the participants understood that the intention was for them to use multiple data sets to create separate polygons within the same layer. They, therefore, tended to add one polygon and move on to the next task. They were then told what the actual task was and asked to try again.
It was observed that multiple participants used a break-line instead of a comma to separate the polygons. This format was not originally supported, however after seeing this approach performed multiple times, the application was changed to accept this format as well. It was also decided to add a line-break between each polygon when displaying already saved files, to make the data more readable. Before this, the next polygon started where the last one ended and it was

hard for the participants to separate them. The instruction for this task was changed to make it clear that the goal was to display multiple polygons in the same layer.



```
[ [ 40.562826, -74.174888 ], [ 40.561078, -74.169827 ], [
40.561057, -74.169823 ], [ 40.548205, -74.16707 ], [
40.547942, -74.166923 ], [ 40.562733, -74.192592 ], [
40.564272, -74.192407 ] ],

[ [ 40.539144, -74.142146 ], [ 40.53907, -74.142304 ], [
40.538949, -74.142338 ], [ 40.538641, -74.141905 ], [
40.537088, -74.141583 ], [ 40.53704, -74.141734 ], [
40.537139, -74.142469 ], [ 40.541173, -74.144348 ], [
40.54051, -74.143591 ], [ 40.539316, -74.142228 ], [
40.539144, -74.142146 ] ]
```

```
[ [ 40.562826, -74.174888 ], [ 40.561078, -74.169827 ], [
40.561057, -74.169823 ], [ 40.548205, -74.16707 ], [
40.547942, -74.166923 ], [ 40.562733, -74.192592 ], [
40.564272, -74.192407 ] ] , [ [ 40.539144, -74.142146 ], [
40.53907, -74.142304 ], [ 40.538949, -74.142338 ], [
40.538641, -74.141905 ], [ 40.537088, -74.141583 ], [
40.53704, -74.141734 ], [ 40.537139, -74.142469 ], [
40.541173, -74.144348 ], [ 40.54051, -74.143591 ], [
40.539316, -74.142228 ], [ 40.539144, -74.142146 ] ]
```

Figure 6.2: The new format for multiple polygons (left) supports line-break, whereas the old format (right) does not. Both formats are still accepted as input, and files retrieved from the database will be displayed as the file on the left

The most common difficulty regarding task 5b, was that the participants expected each data set to produce one circle instead of one circle per data point, as this was the case with the polygons. Because of this, an information box was added explaining how the different shapes are drawn when using different data types.

7 | Results

This chapter will present the results from the usability test. The SUS score of the test will be commented on, as well as the written feedback from the participants and the evaluator's observations during the tests.

The usability test was conducted with 11 participants, all in the age group 18-30 and students with varying majors at NTNU. They spent between 18 and 43 minutes completing the practical part of the test, wherein they used the application to create a survey. Two participants used respectively 40 and 43 minutes, while the average of the other 9 participants were 24 minutes. The feedback process was not timed.

The test was conducted with 2 participants who study or work with geospatial data and 9 participants who do not. 5 of the latter stated that they use electronic maps every month.
As the application is meant to be available not only to professionals but everyone, it was not seen as a major weakness that few of the participants are experienced with geospatial data. It was rather regarded as a good opportunity to see if the information integrated within the application was sufficient for all users to understand.
Table 7.1 show the independent variables collected in the test.

Variable	**Alternative**	**Count**
Gender	Male	3
	Female	8
Age	Under 18	
	18 - 30	11
	31 - 50	
	51 - 67	
	67 -	
Map experience	I work with/study geographical information	2
	I use electronic maps for navigation or other purposes every month	5
	I prefer papermaps	
	None of the above	4
Technology experience	I develop technology for work/study	5
	I use advanced programs for editing/simulation	3
	I mainly use technology for every day purposes	3
	I prefer times before the computer	
Survey experience	I have created several surveys for research purposes	
	I have created simple surveys related to school or other activities	8
	I use surveys in informal settings, like on Facebook	1
	I don't have any experience with creating surveys	2

Table 7.1: Usability test, independent variables

7.1 System usability scale

The System usability scale is a 10-item questionnaire that provides a measure of the user's perception of the usability of a system. They do this by giving each question a value between 1 and 5. These values are then used to calculate the score given by each participant, see appendix B for the calculation method and the SUS questionnaire.
Aftethoer using the application to create a survey, each participant was asked to fill out a SUS test. The collected results will be discussed in section 7.1.2.

7.1.1 Analysing SUS results

Sauro states that there, over 30 years, has been developed 5 ways to interpret SUS scores. The analysis methods are based on collected scores of hundreds of systems.

Percentiles compares your system's SUS score with the scores of the systems in the database, and tells how well your system is received compared to these. The average score is 68. Hence if your system has a score of 68 it has been better received than 50 percent of the systems used to develop this interpretation.

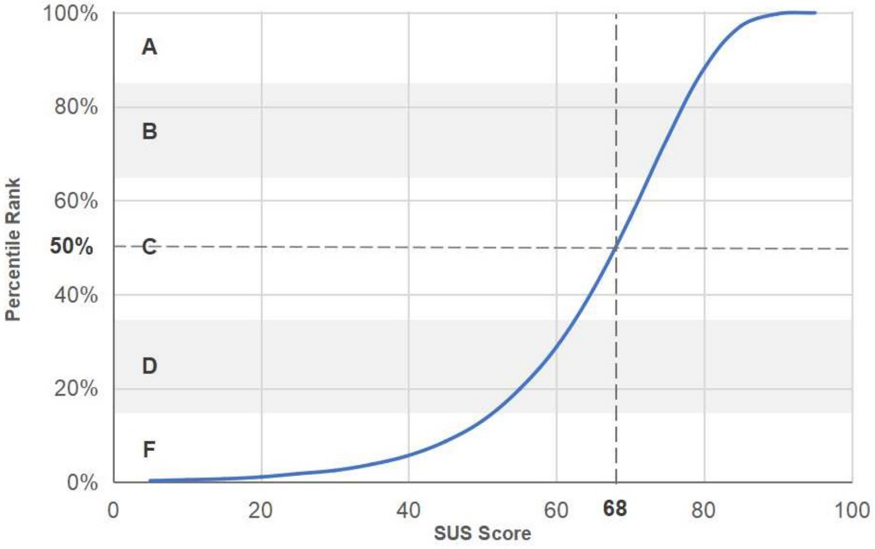


Figure 7.1: SUS percentiles and grade scale

Acceptability is a comparison method suggested by Bangor et al. [2008], and states that "products that are at least passable have SUS scores above 70. (...) Products with scores of less than 70 should be considered candidates for increased scrutiny and continued improvement and should be judged to be marginal at best."

The strictest analysis scale is **promoters and detractors** and is based on what score a system should have for users to recommend it to others. Sauro have used the SUS and Net promoter score data of 4 664 respondents to find a correlation between the SUS score and the users' likelihood of recommending a system. The results showed that the system needs to have a SUS score of at least 81 to be classified as "promoter". While a score of 53 and below classifies the system as "detractor". A score in between classifies the system as "passive". A "promoter" system is likely to be recommended to others, while a "detractor system" means that users will be likely to advise others to not use the system.

The groupings that are classified as "Acceptable" have the following values:

SUS score	Percentile range	Acceptable	NPS
84.1 - 100	96 - 100	Acceptable	Promoter
80.8 - 84.0	90 - 95	Acceptable	Promoter
78.9 - 80.7	85 - 89	Acceptable	Promoter
77.2 - 78.8	80 - 84	Acceptable	Promoter
74.1 - 77.1	70 - 79	Acceptable	Passive
72.6 - 74.0	65 - 69	Acceptable	Passive
71.1 - 72.5	60 - 64	Acceptable	Passive

Table 7.2: Analysis of SUS score

Other analysis scales are grades, visualized in figure 7.1 and adjectives, visualized in figure 7.2.

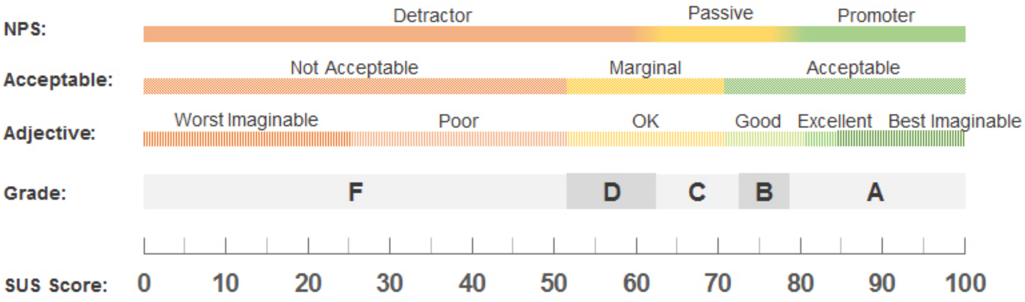


Figure 7.2: SUS analysis scales

7.1.2 Usability test SUS results

The average SUS score from the user test was 80.0 out of 100. This places the application in the 78,9 - 80,7 SUS score group, see table 7.2. Which ranks the application higher than 85 - 89% of systems in the SUS database, and classifies it as "Acceptable" and "Promoter". This is a big improvement from the pilot test score of 72,5 and accomplishes both goals for the SUS score set in section 6.2.1.

Three of the participants gave scores in the 60-70 bracket, three participants gave scores in the 90-100 bracket, whilst the rest were distributed between 77,5 and 87,5. There are no patterns in independent variables regarding the first six scores.

The average SUS score of the participants who develop technology for work/study is 77,5. While the average score for the rest of the participants is 82,1. The participants with technological background tended to give a lower score on question

9, "I felt very confident using the system".

The groupings are not evenly enough divided to comment on the average of the other independent variables.

7.2 Written feedback

This section contains a summary of the comments and opinions from the participants, given in the written feedback section of the user test. The complete list of answers can be found in appendix C.

Formulation
A couple of the participants had difficulty understanding the formulation of the questions. However, most participants state that once they read the instructions again, more carefully, they understood what to do.

Difficult tasks
The participants' feedback indicates that they found task 5a (polygon layer with multiple polygons), 5b (circle layer with multiple radiuses) and 4b (changing the basemap) most difficult.

Appreciated tools
The participants especially appreciated the layer tools, the fact that they could choose what type of visualization to use and what color the layer would be, the participant interaction with the map and the map in general.

Missing tools
The functionality most participants thought were lacking were the ability to draw in the map, both to save data layers and to offer more participant interaction, as well as the ability to pair the questions not only with a map but also an image.
Other functionalities that were mentioned include the ability to upload layers by using GeoJSON files, as well as a more visual presentation of the results.

Information boxes and tooltips
The participants found the information boxes and tooltips helpful while conducting the survey.
However, feedback suggests that some were too long and could have been split up in more specific boxes and that the information boxes belonging to the data format and circle radius were a bit hard to understand. A more common opinion was that the text was too small.
Some of the participants drew attention to features that lacked a description, and may have benefit from the addition of a tooltip or an information box. However, none of these features were mentioned more than once.

Design
Overall the participants thought the design was clean and simple. Whereas most participants appreciated this, some would have preferred to have more specific buttons directly on the page instead of inside settings or "edit map". This was because they thought it was not always clear where to find the different functions, especially in settings, and because they thought the "edit map" modal was too crowded.

General impression
The overall feedback from the participants was positive. They stated that they could understand the need for an application like this.
Several of the participants stated that they thought it was fun to test the application and that, even though it would not be relevant for them currently, they would like to use it if they ever need to create a survey containing maps.

7.3 Observations

This sections will state the repeating observations made by the evaluator. Some of the observations will serve as comments to the written feedback by the participants summarized in section 7.2, while some will comment on other observations.

Time
As mentioned, the participants spent between 18 and 43 minutes completing the practical part of the survey. As the participants were encouraged to explore the application if they desired, instead of following the instructions meticulously, some participants spent longer on the survey because they wanted to experiment. While others spent some time making sure their survey had a theme they enjoyed. Some of the participants did spend a relatively longer time because they were struggling with how to perform tasks.

Text size
Even though only a couple of participants commented on the text size, it was observed that the majority had difficulty reading the information boxes due to the text size.

Information and navigation
It was observed that when beginning a new task, the participants would often immediately open the correct location for executing the task. Whether this would be if they were to go via settings or use features in "edit map". However, sometimes they would navigate away from this, thinking that the desired information was not in this location.
The same was the case for the information boxes. The participants would open the information box they needed to read to execute the task, then close it again after a few seconds. Sometimes they would state "I can't be bothered reading

that". Other times they believed the information box did not contain the information they were looking for.
The information boxes that the participants spent the longest amount of time on were "Data format" and "Circle radius". These were also the information boxes most frequently ignored by participants who were looking for the information they contained.

Difficult tasks
Co-responding with the participants own feedback, the evaluator observed that the most difficult tasks were to change the basemap, to add a map layer with multiple polygons and to add a layer containing circles with different radiuses. When changing the basemap, some of the participants were confused by the "apikey/accesstoken" field, while most were confused by the webpage that distributed the basemaps, not knowing exactly where to get the information they needed.
Regarding circles, it was a common misunderstanding that one dataset would visualize one circle, as supposed to one circle per data point. A few of the participants stated that this was because the dataset in the previous task had visualized one polygon. When understanding the concept, all participants managed to execute the task.
Even though the feedback indicates that most participants found the basemap task to be the most difficult, the evaluator experienced that more participants struggled with the polygon layer in 5a. Even though the task instructions were modified based on the feedback from the pilot test, some participants did not understand that they were desired to visualize more than one polygon in the same layer. After realizing this, the thing most participants struggled with was using the right format to save the data points. As stated in the previous section, participants would often open the "data format" information box and close it without reading the content. When they did discover that the content was what they were looking for, they would manage to execute the task and the information there would be used a lot more frequently to confirm the data format of layers that were added in later tasks.

8 | Discussion

This chapter will discuss the feedback and the results collected in the usability test with regards to the goals set in 6.2.1.

8.1 SUS

Regarding SUS the goal for the usability test was that the application had to get a score that classified the system as "Acceptable", while the additional preferred goal was to get a score classifying the system as "Promoter". Promoter meaning that users of the system will be likely to recommend the system to others.
As mentioned in 7.1.2, the overall average SUS score is 80,0. which fulfills both of these goals. This means that the participants regard the application as user-friendly and something they could see themselves and others use. However, there are improvements that are possible to make in order to create an even more user-friendly application.

8.2 Intuitive and information boxes

Two of the goals of the usability test were to determine if the participants found the application intuitive and understandable. In the case that they did not, the aim was to discover whether the application helped them find the appropriate information in order to be able to use the desired functionality after all.
Judging by the feedback given by the participants themselves and the observations made by the evaluator, the participants found most of the application to be self-explanatory. It was not observed that any of the users had any difficulties with the questions, creating a survey or new page, the shareable link or the results.
Some of the participants did not immediately understand what the different values in the "edit map" modal did, however this was mostly made clear after reading the information boxes available.

8.3 Identifying problems

Another goal for the usability test was to identify what problems the participants had with the application.
Both the feedback from the participants and observations find the main problems to be the participants' interaction with the mapdata modal and the basemap.

Mapdata modal
There were a few reoccurring difficulties with the mapdata modal. One was how to input the correct data format for the geospatial data and the radiuses for the circles. Another evolved around the understanding of the geospatial information and how the different data types were visualized.
Many participants had trouble adding the polygon layer the first time they interacted with the mapdata modal, as stated in chapter 7. However, both the participants and the evaluator noted that once they got used to the idea of what the data layers were, knew where to find the data format information and how to interpret the geospatial data, they greatly appreciated this functionality.
The same is true for the use of different radiuses, once the participants understood that one data point represents one circle instead of one layer representing one circle. The information the participants used in these tasks was always available and was sufficient once discovered. However, in these cases, the information was not visible enough in the application. The solutions discussed regarding the mapdata modal will, therefore, evolve around making the information in the modal more visible and more structured.

One option that was considered when implementing the 'data format' information box was to, by default, add the required data format as a placeholder in the 'geospatial points' input field when the datatype of the layer was selected. A placeholder is the weak gray text sometimes found in empty input fields. This idea was dismissed, as the placeholder disappears when anything is written in the field and it was desirable for the users to be able to view the format while editing the geospatial data. However, it is an option to implement the placeholder as well as keeping the information box the application has today, in order to make it more visible to the users that there is a required input format and that this changes with the data type.
Another recurrent information section that participants struggled with was how the different data types visualized the geospatial data. As mentioned, this was most evident with the circles. There is an information box belonging to this field that explains how the types work, however, most participants did not read this. A solution could be to get this information across to the user while they are choosing what data type to use. In the same fashion that the difference between radio buttons and checkboxes is explained in the create question component, see figure 8.1.

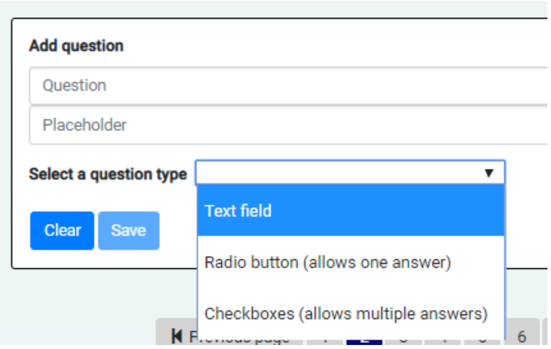


Figure 8.1: How the create question component informs the users of the difference between radio buttons and checkboxes

A different approach could be to restructure the information given in the modal. Instead of splitting it according to theme, it is possible to split it according to what data type they belong to and only display it when this data type is selected. The result of this would be that the user only has access to immediate useful information and would be less likely to be overwhelmed by all the information available. It will also be easier for the users to grasp that adding different radiuses to different circles is an option.
This approach will still require a solution to the same basic problem mentioned earlier, which is how to get the users to read the information provided.

Basemap
Whereas the participants got the hang of how to add and customize the data layers after a few tries, this was not the case regarding how to change the basemap. All participants managed to execute the task, however, they did not fully understand how the functionality worked.

It was a conscious decision to make the information box belonging to the basemap field as simple as possible. Both because it is hard to explain in full how it works as different providers use different attributes in their basemap links and because an attempt to explain this would require more text than what should be placed in an information box. However, the user test shows that a more detailed description of the field is needed if it should be regarded as simple enough for even inexperienced users to utilise.
This can either be done by writing a longer explanation to be displayed in a pdf or separate modal, or achieved in the same regards as the 'data format' and 'color' information in the mapdata modal. Neither of these are optimal. The pdf approach requires the user to leave the application, the modal approach does not allow the user access to the basemap field at the same time as the information is visible and the latter approach adds more complexity to the 'edit map' modal which several participants stated is already too complex.

As one of the participants suggested in their feedback, one option to make the 'edit map' modal less complex is to split the functionality there into multiple modals. For example one modal and button for 'Basic settings' and one for 'Basemap'. Adding the basemap information in its own modal would allow for including more information about how the functionality works without overcrowding the modal itself, as the basemap only has three simple input fields related to it. This could also be implemented with fairly little change to the current code.
The design perspective of this solution will be discussed in the next section.

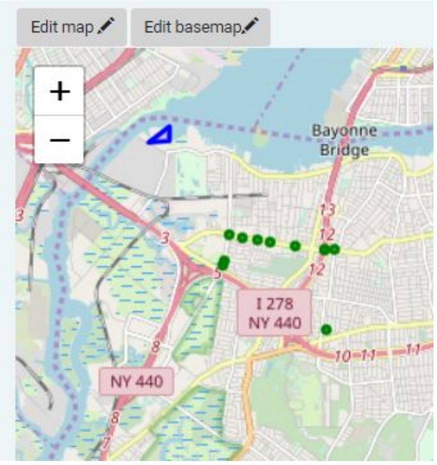


Figure 8.2: Adding more editing buttons can be a way to simplify the map modals and adding more information in the application

8.4 Design

The test also wished to determine if the design of the application helps the users navigate in the application.
The participants gave some mixed feedback on this point. It was observed that most participants found the tools they were meant to interact with quickly, some found them after a bit of time, while others found the right tools quickly but did not realize they were the tools they required. The written feedback suggests that the participants shared these impressions. Even though most of the participants state, in feedback question 9, "What did you think of the design of the application", that they found it simple and clean, it is suggested by several participants throughout the rest of the feedback survey that they wish the buttons in the application were different. Some state that a portion of the options in settings could have been more visible and that the editing tools were hard to find at first. However, because there were only two buttons, they found the desired tools in a short amount of time. Others wished for multiple buttons

to display more specific groupings of the available tools, while other participants expressed that they appreciated everything being in the same place. These design options should be tested further to find the optimal solution.

In the written feedback the participants were asked what tools they missed in the application. Whereas most participants stated that they did not miss any tools, some had suggestions for future work. Several of these, as well as several of the planned 'may have' requirements for the application, are related to the map and the 'edit map' modal. As discussed, some participants found this modal to be too crowded even with the current design and additional functionality would make the modal even more complex. One option to keep the modal simple while being able to offer more functionality, is to do as one participant suggested and add more, separate editing buttons. Another approach is to create an 'advanced editing' choice. Wherein the user, either in the basic settings of the survey or in each modal, can choose to get access to additional editing tools. This approach still has the challenge of not making the modals too large and complex. A combination of the two might be most favorable.

The usability test showed that a design change that should be made to improve the overall usability and user experience of the application, is to enlarge the text size of the information boxes and tooltips.
Some of the participants found certain tool names and the purpose of some fields unclear, therefore functionality naming throughout the application should also be evaluated.

8.5 Other takeaways

Independent variables and SUS
The results chapter states that participants that work or study technology development scored the system lower than other participants. A reason for this could be that these users have other standards for usability, design or functionality implementation than the other participants. Coupled with having their own opinions on how a system should be presented to the user that does not correspond with this application.
The "technology experience" variable was intended to capture how well the participants were expected to understand the application based on previous experience. In hindsight, the "develop technology" alternative might not be as relevant as first thought. The idea was that these participants would know how to use technology better than the other participants so they would easily pick up on how to use the application and would have experience with user design. This was the prediction, however, it is not necessarily true for this type of application. The application is meant to be a simple editing tool and because of this, is somewhat more complex than other webpages, as the user has more options to manipulate the page. If standard webpages are the only ones the participant uses and develops, they are still not relevantly experienced with applications

that offer a lot of options,despite having more experience with technology than average users. This can be supported by the fact that several of these participants gave a lower than average score to the question "I felt confident using this system".

With the variable intention in mind, the most experienced users should be the participants who answered, "I use advanced programs for editing/simulation". These participants have an average SUS score of 82,5 compared to the 77,5 of the technology developing participants. Implying that participants with experience with similar tools found the system easier to use. However, there were only three participants who used advanced editing or simulation tools, which is too few to draw a final conclusion.

A solution to this problem could have been to split the variable into two more specific parts. One to map the participants experience with editing tools, instead of technology in general. As well as another mapping the participants experience with user design.

Participant variety

The participant variety in the usability test is not optimal, especially regarding the age group and the survey experience the participants had. As the test was executed in a university, all the participants are relatively young and fall into the same age group. It would have been preferable to test with a more varied age group as they might experience the application differently.

There were also no participants who stated that they have created surveys for research purposes. It can be argued that this is not negative in regards to testing the application, as the participants can not rely heavily on previous experience with survey creation. It can, however, impact the feedback regarding faults of the application or areas in which it is lacking. Close to all the feedback received about tools that can be added to the application relates to offering more variety in what surveys are possible to create. Suggestions include surveys that can also display images, surveys with more user and participant interaction etc. There are no suggestions regarding analysis or monitoring of the participants while they execute the tasks in the survey. This could be timing the participants or rotating the order of the questions shown, both tools the literature study [Fog, 2018] shows are commonly used when conducting research surveys. Therefore the application should be presented to multiple users that fall into this category for feedback if it is to be developed further.

9 | Conclusion and future work

This thesis looks at an application developed to make it easier to create map surveys. The application is implemented using the MEAN stack and the Leaflet.js library.
The system requirements are set according to what is needed to implement the essentials for a working survey application, as well as the tools and functionality discovered in the literature study in TBA4560 performed in fall 2018 by Fog.

The application allows the user to create surveys containing several types of questions and to visualise map data as points, polygons, lines or circles . One survey page can contain multiple questions, questions as well as a map or a map that allows the survey participants to interact with it.
Each survey has a shareable link that can be distributed to give participants access to the survey. The results are saved and available to the survey creator.

The thesis discusses the results from the usability test conducted on the application. The results from the test show that the participants find the application to be intuitive and simple to understand, even though there still are measures that can be taken to improve the usability.

The test also identified some issues with the application. These were mainly in design, usability and formulation of the information related to the geospatial points input field, the circle radius input field and the 'change basemap' functionality.
The thesis discusses how different design changes can contribute to making these tools more user-friendly, and contribute to the users understanding of the application.

Even though the participants found the map layer functionality difficult to get familiar with, the feedback shows that they greatly appreciated this component and the customization options.
The participants were generally positive to the application and thought it was a useful tool to have access to.

9.1 Future work

Future work with the application should include implementing the remaining requirements set in this thesis. Several of these relate to the map, adding to its functionality and customization options, which participants feedback indicates as a primary user desire.
Feedback from the users also shows that communication and presentation of the information provided in the application can be further improved.

The themes mentioned in the introduction, that were not focused on in this thesis, are also a good starting point for future development. Of these, presentation of results has been mentioned by the usability test participants as a point of improvement.

It should be noted that before future implementation the application should be evaluated by more users belonging to different age groups than the ones in the user test, as well as users who are familiar with how to construct surveys for research purposes. Collecting and adjusting the application based on feedback from several categories of the target group will make the application more appealing to a wider range of users with different needs regarding functionality and analysis tools. Keeping a easily navigable design that points users in the right direction as well as implementing more complex tools, will keep the application approachable for inexperienced users while still being able to offer advanced users the functionality they desire.

Bibliography

Angular architecture. Available at: https://angular.io/guide/architecture [Accessed 24 March 2019].

Pensjonsalder. Available at: https://www.spk.no/Ord-og-uttrykk-om-pensjon/Pensjonsalder/ [Accessed 27 April 2019].

Aaron Bangor, Philip T Kortum, and James T Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594, 2008.

Craig Buckler. Sql vs nosql: The differences. *Sitepoint*, 2015. Available at: http://www. sitepoint. com/sql-vs-nosql-differences/ [Accessed 13 March 2019].

Jeff Dickey. *Write modern web apps with the MEAN stack: Mongo, Express, AngularJS, and Node. js.* Pearson Education, 2014.

Don A Dillman, Robert D Tortora, and Dennis Bowker. Principles for constructing web surveys. In *Joint Meetings of the American Statistical Association*, pages 1–16, 1998.

ISO DIS. 9241-210: 2010. ergonomics of human system interaction-part 210: Human-centred design for interactive systems. *International Standardization Organization (ISO). Switzerland*, 2009.

Anne Sofie Strand Erichsen. *Optimizing the Micro-Tasking Workflow and Exploring its Usage Potential Within Geospatial Data.* 2017. Master thesis at the Norwegian university of science and technology, faculty of engineering.

Synne Fog. *Creating web-based map experiments - methods and technology.* 2018. Specialization project at the Norwegian university of science and technology, faculty of engineering.

Anna-Lena Keute. Norske studenter blandt de eldste i europa. Available at: https://www.ssb.no/utdanning/artikler-og-publikasjoner/norske-studenter-blant-de-eldste-i-europa [Accessed 27 April 2019].

Jakob Nielsen. *Usability engineering*, pages 1–67. Elsevier, 1994.

Yvonne Rogers, Helen Sharp, and Jenny Preece. *Interaction design: beyond human-computer interaction*, pages 225 – 226. John Wiley & Sons, 2011a.

Yvonne Rogers, Helen Sharp, and Jenny Preece. *Interaction design: beyond human-computer interaction*, pages 433 – 452. John Wiley & Sons, 2011b.

Jeffrey Rubin. *Handbook of usability testing : how to plan, design, and conduct effective tests.* Wiley technical communication library. Wiley, New York, 1994.

Jeff Sauro. 5 ways to interpret a sus score. Available at: https://measuringu.com/interpret-sus-score/ [Accessed 3 May 2019].

Appendix A

This appendix contains the test instructions given to the participants in the user test and the document "Data til test".
The document "Data til test" contains the data available to the participants during the test.

Geo-test application - user test

1. Go to the application: http://geo-tests.herokuapp.com/survey

General functionality

2. Create a new survey. Name it and insert the other required information as you please
3. Navigate to the second page and edit the age interval options
    a. Delete, edit or add any other questions you see fit for a registration page (options: change the question text, change the type of question, add or delete alternatives)
    b. Change the page instructions

Map page and map edition

4. Create a new "map and question" page
    a. Set the default settings of the map to
        i. Default zoom 12
        ii. Latitude: 40.592473
        iii. Longitude: -74.127433
    b. Change the basemap (background map).
    You can get alternatives here:
    https://leaflet-extras.github.io/leaflet-providers/preview/
    The first alternative is the basemap already used in the application, so please pick a different one
    c. Set static map and layer control according to preference

Map layers and questions

One of the things the application allows for, is visualize geographical information in the map. That's what you will do in this part.

5. The "Data til test" document contains 10 datasets consisting of geographical data points. One points consists of two number [latitude, longitude]. Use these datasets to add data layers to the map
    a. Use at least two of the datasets to create **one** polygon (mangekant/figur) layer, that contains multiple polygons. The result should be displayed as multiple, separate figures
    b. Add a circle layer, containing circles with **different** radiuses
    c. Choose some of the datasets, at least two, and add them to the map as new layers. Choose type and other options that might be available according to own preferences
    d. Save and return to map
6. Add two questions of any type and add page instructions

Marker map

7. Add a new page of type “map with participation interaction”
    a. What do you think the difference between this and the regular map is?
    b. Create a static map with default settings
        i. Default zoom 12
        ii. Latitude: 40.592473
        iii. Longitude: -74.127433
    c. Add any dataset as a polygon layer
    d. And ask the user to click in the center of the polygon

General functionality

8. Go through the survey and change, add or delete anything you please
9. Get the shared link that allows participants to take your survey. Access it and submit a reply
10. Go back to the administrator page and view the registered results

Feedback

11. Take the SUS test
12. Reply to this survey
    http://geo-tests.herokuapp.com/edit?id=5cc406af556ad00017976696

Data til test

This document contains 10 datasets. Each dataset contains several geographical points
One geographical point is written on this format: [40.642307,-74.183728]

[40.642307,-74.183728],[40.642240,-74.183651],[40.642261,-
74.183633],[40.642280,-74.183622],[40.642301,-74.183615],[40.642323,-
74.183613],[40.642342,-74.183617],[40.644970,-74.179340],[40.644560,-
74.179299],[40.644583,-74.179298],[40.642342,-74.179472],[40.642690,-
74.179340]

[40.562826,-74.174888],[40.561078,-74.169827],[40.561057,-
74.169823],[40.548205,-74.167070],[40.547942,-74.166923], [40.562733,-
74.192592],[40.564272,-74.192407]

[40.596584,-74.085593],[40.596781,-74.085513],[40.597312,-
74.084944],[40.597438,-74.084809],[40.597462,-74.084742],[40.598563,-
74.081710],[40.598390,-74.077164],[40.597637,-74.073872],[40.597210,-
74.073623],[40.600183,-74.064638],[40.596306,-74.061829],[40.595236,-
74.062829],[40.593188,-74.060780]

[40.553807,-74.184640],[40.554240,-74.185508],[40.554549,-
74.186093],[40.553275,-74.188265],[40.552424,-74.187074],[40.550714,-
74.188576],[40.551133,-74.189318],[40.553754,-74.192417],[40.562079,-
74.194076],[40.562733,-74.192592],[40.564272,-74.192407],[40.564392,-
74.190794]

[40.563049,-74.190335],[40.564549,-74.185254],[40.565163,-
74.184371],[40.566527,-74.184311],[40.566651,-74.182846],[40.563983,-
74.177945],[40.563241,-74.176594],[40.561375,-74.171463],[40.561078,-
74.169827]

[40.539144,-74.142146],[40.539070,-74.142304],[40.538949,-
74.142338],[40.538641,-74.141905],[40.537088,-74.141583],[40.537040,-
74.141734],[40.537139,-74.142469],[40.541173,-74.144348],[40.540510,-
74.143591],[40.539316,-74.142228],[40.539144,-74.142146]

[40.624931,-74.146120],[40.624992,-74.144025],[40.611932,-
74.145790],[40.622401,-74.167934],[40.623125,-74.167600],[40.627367,-
74.166559],[40.626874,-74.163733],[40.626530,-74.160500],[40.626172,-
74.157800],[40.625536,-74.152423],[40.624931,-74.146127],[40.624931,-
74.146120]

[40.579728,-74.100345],[40.578623,-74.101670],[40.574559,-
74.105646],[40.574527,-74.105678],[40.579181,-74.114826],[40.579738,-
74.114235],[40.581792,-74.111276],[40.582679,-74.109248],[40.583715,-
74.107974],[40.579728,-74.100345]

[40.619909,-74.085899],[40.619146,-74.085998],[40.618515,-
74.086080],[40.610806,-74.094270],[40.610742,-74.098180],[40.612015,-
74.099068],[40.612419,-74.099350],[40.612635,-74.099623],[40.619409, -
74.090224],[40.620307,-74.089420],[40.620114, -74.087745],[40.620501,-
74.087011],[40.620080,-74.087071]

[40.575770,-74.125934],[40.573479,-74.133159],[40.572233,-
74.141883],[40.570783,-74.141536]

Appendix B

This appendix includes the SUS form the user test participants answered after performing the test, as well as information about how to use and calculate the score of the test.

System usability test

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome (tungvint) to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lost of things before I could get going with this system	1	2	3	4	5

Using SUS

The SU scale is generally used after the respondent has had an opportunity to use the system being evaluated, but before any debriefing or discussion takes place. Respondents should be asked to record their immediate response to each item, rather than thinking about items for a long time.

All items should be checked. If a respondent feels that they cannot respond to a particular item, they should mark the centre point of the scale.

Scoring SUS

SUS yields a single number representing a composite measure of the overall usability of the system being studied. Note that scores for individual items are not meaningful on their own.

To calculate the SUS score, first sum the score contributions from each item. Each item's score contribution will range from 0 to 4. For items 1, 3, 5, 7 and 9, the score contribution is the scale position minus 1. For items 2, 4, 6, 8 and 10, the contribution is 5 minus the scale position. Multiply the sum of the scores by 2,5 to obtain the overall value of SU.

SUS scores have a range of 0 to 100.

Appendix C

This appendix contains the written feedback from the participants of the user test.
Repetitive replies were removed.

The feedback survey can be found at:
http://geo-tests.herokuapp.com/edit?id=5cc406af556ad00017976696

User test – written feedback

1. Do you see the need for a tool like this application? Why/why not?

Yes surveys are often used to inform people of various thing, like what most people prefer out of water brand...

Jeg er ukjent med i hvilken grad noen har behov for å lage kartbaserte undersøkelser, men hvis de har, så likte jeg mulighetene som finnes i applikasjonen.

Yes. It offers a more geographically inclined option for the creation of surveys. Full integration of maps into the survey makes it a more linear process.

Probably, yes. I would not need it in my studies, but understand that it is relevant for some subjects. It simplifies the use of maps in surveys.

Ja. Finnes ikke et godt system fra før.

Ja! Det er veldig mye lettere å grafisk visualisere steder enn å måtte forklare.

I do not know of another survey tool that allows users to easily ntegrate maps in their surveys like this one does. Therefore, I do see the need for this application.

Yes, surely useful for Geomatics-related surveys

Kan være kjekt å bruke når deltakerne bør knyttes til steder

Yes. Interactive surveys are fun, and more compelling than regular ones.

Kan se at folk med behov for å lære kart og kartnavigering kan trives godt med dette verktøyet. Lett å lag enkle undersøkelser basert på forskjellige kart

It is a nice tool to introduce something, maps in this case, that is harder to comunicate with word. Take for example you want to know someones favorite spot for grabbing a pint on a hot summer day. Giving both the name of the place and the position on a map could be nice in such a senario. Especially if either the person recommending or the one getting the recommondation do not have an intimate knowlegde of the place.

2. Did you have trouble understanding the formulation of any of the questions? Which?

No not really

No, not really. Problem with understanding was more related to my little knowledge of maps. But I understood it when I read it again more carefully.

5c

Nei, det var godt formulert.

Ja, jeg ikke skjønte hva "Data til test" gikk ut på i pkt. 5. Hva dataene var og skulle brukes til.

Yes, but only due to my lack of technical competence in the subject

Mesteparten av misforståelsen var og grunnet jeg ikke leste spørsmålene godt nok. Samt en følelse av å bli testet som alltid gjør meg noe nervøs

Some questions point to places or buttons that are not visable on the screen you start on. Like number 4 "Create a new "map and question" page". First you have to click on create a new page option, then you can chose your type of page from the menu presented. It took some seconds to realise, so it was not a major toruble.

3. What task was the hardest to understand how to perform (the hardest to figure out how to do, not the hardest question to interpret)?

The polygon layer because of the comma that needed to be in between the polygons. It was a bit hard to see

Changing the basemap

The multiple polygraphs question. The addition of square brackets and commas was not immediately evident, potentially this is due to my lack of experience with geographical data points.

Maybe understanding how to change the page instructions the first time. I assumed it would be right next to the description, like it is right next to the questions.

Creating data layers in task 5.

Change the basemap (background map)

To add the circle layer with only a point for each circle, not a dataset

Det var ganske greit det meste. Kanskje å finne utav hva som var "basemap link"

It was a little hard to find out how to create a new page. I was looking for it around the page navigation part of the application (at the bottom). I had similar trouble when trying to view results. Here i could have been intuitive to have a button on the home admin page directly next to the form. EG MySurvey - Edit -View Results. On the other hand there was only one settings button so on the page which was quite visible so it did not take too long to find this functionality

Not quite sure, maybe task 4.b? Wasn't quite sure how the API-keys worked for basemaps and such. Especially when the Leaflet-extra page did not give an api-key for my selected Base map?

Kanskje spm. 4b. Hvilken link man skulle kopiere og sette inn

question 5 a and b. And than I got the hang of it

Var ingenting som var spesielt vanskelig å gjøre. kanskje bytte kart, fordi jeg selv surret litt med uthentingen av linken. Men alt på nettsiden var godt forklart hvis man lurte på noe

The hardest was making one circle layer with circles of different layers as instructed in 5 b

4. Did you think any of the tools/options were unnecessary?

No I imagine they all can be used to various different surveys.

No, not as far as I know.

Den nederste infoboksen inni edit map ga jeg ikke så mye oppmerksomhet til, og skjønte egt helt hva det var. Men vil tro at den er nyttig :)

Ikke som jeg rakk å tenke på under testen

5. What tools did you appreciate?

The layer tool for the map, once I got the hang of it

basemap option, zooming, layering information and layer control, question types, generally everything?

The map with participation interaction.

The many editing options, to tailor the form.

Kartet

Adding maps

At man står veldig fritt til å bestemme både farge/form osv. på figurene man satte inn, som gjør det veldig lett å visualisere og skille mellom de forskjellige figurene.

The drop a marker tool and the add layer tool.

Creation of layers, polygons, circles etc

Likte godt "Hvilken type" som i sirkel, polygon osv, den var enkel å forstå og visuell:) Likte også fargevalg-verktøyet, og satte stor pris på infoboksen om hvordan å separere de ulike koordinatene!

Layers and participant interaction

Likte godt at man kunne interacte selv som bruker

Begin able to chose your own map, nice and quick way to experiment.

6. What tools/editing options did you miss?

Diagrams, where one maybe had to guess how big parts of a cake diagram is. The option to insert images next to questions.

Can't think of any. Maybe the function of conditional parts of the survey - "if female, go to this page" and "if male, go to this other page" as an example.

At man kan klikke på et sted for å legge inn et punkt. Litt vanskelig å jobbe med latitude/longitude når man ikke har et forhold til det. Det kan være fint med en visualisering av resultatene på testen etter den har blitt sendt ut til deltakere

Ability to draw on the map would have been useful both for the admin to draw layers and for the person answering to maybe answer by drawing polygons or lines (not just markers). It would also have been useful to be able to upload layers with geojson files of wms links.

Vanskelig å si når jeg ikke pleier å lage undersøkelse. Men kanskje muligheten for å sette inn bilder slik at man kan assosiere(?). Og hadde likt det hvis det var litt større skrift på alt, større buttons, og informerende tekst til hva man skulle gjøre/hva denne delen gjorde/hva som var neste steg for en perfekt undersøkelse. Litt som en som er med og hjelper på veien. Evt kunne man kanskje velge å huke av "Help during creation" eller noe, så man ikke må ha det med når man er skilled :P

Drawin polygons and paths

7. Were the tooltips and helptexts throughout the application helpful?

Yes

Yeah, probably. Some were quite long. I had problems concentrating to the end, wanted to go editing before I had read them all.

Yes, could be slightly more concise.

Yes! But the text was really small, hard to read.

ja, men skulle gjerne hatt større knapper/mer tekst, og kanskje flere knapper relatert til kartet, siden det var mye å gjøre der

Yes, but i strugled to understand circles. Might just be me

The ones with the figures

8. Is there anywhere that did not have helping text that could have needed it?

No

New layer

For the "geographical points" box when creating a new layer

Jeg tror ikke det. Evt en tekst som er overordna på hver side; "her skjer det og det".

Kanskje for kartuthenting. Var noe sånt at man kunne trykke på der det stod "basemap link", men ingenting skjedde. Trodde kanskje man fikk mer forklaringer etc når man gjorde det.

How the create page worked. Not a major deal though

9. What did you think of the design of the application?

Simple to understand and navigate

At first I think the editing could be easier to spot than the "Innstillinger"-wheel. But of course, when you have first found it, you know where it is.

It's immediately clear and intuitive. The font size was not universally easy to read.

Clean, easy to use. But the font could be larger - use the whole page.

Enkelt og ryddig

Veldig bra, oversiktlig og enkel

Fint design

Clean and neat.

Fin! Likte fargene og enkelheten :)

Simple and easy to understand

Ålreit design. Men kan gjerne jobbes litt mer med for både brukervennligheten og for visualiseringen

It is nice, works and is in my experience easy to understand. At least with some technical experience.

10. Were any of the tools hard to find?

No they were all in the same place, so they were easy to find

New layer

New page and view results could have been more visible

Jeg rota ganske mye inn på edit map, synes det var litt smått og knotete, og hadde kanskje likt det bedre hvis det var flere "Edit maps"-knapper til ulike formål

11. Other comments

Synes den var enkel når man først visste hva man skulle gjøre :) Men vanskelig i starten hvis man ikke er kjent med kart og det var litt forvirrende da det vinduet med kode og nettsider kom opp.

It was fun

Selecting circles leads to a new field popping up, I didn't notice it at first, maybe have some way to alert the user it is there

NTNU
Norwegian University of
Science and Technology