

UNIVERSITY OF TARTU
Institute of Computer Science
Software Engineering Curriculum

Partha Sarathi Kar

SpatialData IO: A web-based spatial data customization tool for collection of area of interest information

Master's Thesis (30 ECTS)

Supervisor(s): Juhan-Peep Ernits, Ph.D

Dietmar Pfahl, Ph.D

Andreas Kiik, MSc

Tartu 2019

SpatialData IO: A web-based spatial data customization tool for collection of area of interest information

Abstract:

To get services from some spatial data analysis service provider, a user often needs to specify the area of interest (AOI) to the provider in some spatial data format. The user has multiple ways to express their wish to collect AOI information. One is writing all information in a file according to the spatial data format or download datasets from the world wide web. Due to the technically detailed structure of spatial data, first one is too complicated and time-consuming for a non-specialist to write manually; for example, writing location information for 1000 buildings, roads or rail line of a city. On the other hand, a full dataset may include unnecessary location data, few of them may be small, big or over-lap with other objects. Additionally, data can be in different data formats, e.g. geojson, kml, shp etc. So it is quite challenging for the user to quickly understand and modify the spatial data according to their need. The goal of this thesis is to develop a free and open-source user-friendly web solution for non GIS specialist end-users so that they can solve above-mentioned problems graphically with the help of performing different spatial operations in the background. After processing those data end-users will be able to visualize and check the processed data on the map and it will be available for download in different spatial data formats as well. The expected outcome of this application is to provide a graphical solution for the user to perform different geospatial operation for collecting information for the AOI. That will help users to send clean and specific information regarding AOI to their desired providers.

Keywords:

geographic information system, web-based user interface, area of interest description

CERCS: P170, Computer science, numerical analysis, systems, control

SpatialData IO: veebipõhine ruumiandmete haldamise töövahend huvipiirkondade andmete kogumiseks

Lühikokkuvõte:

Ruumiandmete pärimiseks teenusepakkujalt tuleb kasutajal tihti kirjeldada huvipiirkond (Area of Interest, AOI) mingis ruumiandmete vormingus. Kasutajal on mitmeid viise, kuidas selliseid andmeid esitada. Üheks võimaluseks on ruumiandmete salvestamine faili mõnes ruumiandmete vormingus või siis laadida veebist alla terved asjakohased andmestikud. Kuna ruumiandmete vorming on tehniliselt detailiderohke, siis on kirjeldatud lähenemine ebapraktiline. Näiteks pole mõistlik käsitsi kirjeldada suure hulga hoonete või taristuobjektide paiknemist. Teisalt võib terviklikus andmestikus olla liiga palju teavet, mida pole käsil oleva ülesande lahendamiseks vaja. Lisaks võivad andmed olla erinevates vormingutes, näiteks geojson, kml, shp jt. Seega on ruumiandmete mugav kirjeldamine lõppkasutajale suure lisandväärtusega. Käesoleva magistritöö eesmärk on arendada vabavaraline kasutajasõbralik veebipõhine lahendus, mis aitab kasutajatel, kes pole geoinfosüsteemide spetsialistid, kirjeldada huvipiirkondi graafiliselt nii, et mitmed abistavad toimingud tehakse ära taustal. Pärast vastavate andmete töötlemist saavad kasutajad tulemust kaardil visualiseerimise abil kontrollida. Tulemus on alla laetav erinevates ruumivormingutes. Töö tulemuseks on testitud rakendus huvipiirkondade kirjeldamiseks.

Võtmesõnad:

geoinfosüsteem, veebipõhine kasutajaliides, huvipiirkonna kirjeldamine

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Table of Contents

1	Introduction.....	6
1.1	Motivation.....	6
1.2	Problem statement.....	8
1.3	Contributions	9
1.4	Organization of the Thesis.....	9
2	Software Requirements Specification.....	11
2.1	External interface requirements.....	11
2.1.1	User Interface.....	11
2.1.2	Communication Interface.....	12
2.2	Functional requirements.....	12
2.3	Logical data requirements.....	19
2.3.1	File format.....	19
2.4	Non-functional requirements	19
3	Background and Related Work.....	21
3.1	Area of Interest	21
3.2	Related work	21
4	Methodology.....	23
4.1	Used techniques	23
4.1.1	Visualization	23
4.1.2	Analysis.....	24
4.2	Used technologies	27
5	Software Design Description.....	30
5.1	Software perspective.....	30
5.2	Software functions	30
5.3	User characteristic.....	31
5.4	Constraints	31
5.5	Use Case diagram	31
5.6	Sequence diagram	31
6	Software Implementation.....	36
6.1	Choice of technologies for development	36
6.1.1	Choice of technology for Front-end development.....	36
6.1.2	Choice of technology for Back-end development	36
6.2	Software development	37
6.2.1	Front-end.....	37

6.2.2	Back-end	38
6.2.3	Source code of the projects	38
7	Software Evaluation	39
7.1	Acceptance test for functional requirements.....	39
7.2	Automated tests.....	47
7.3	Non-functional requirements test.....	53
7.4	Tool Evaluation by User	55
8	Summary	58
9	References.....	59
	Appendix.....	61
I.	List of abbreviations and terms.....	61
II.	License	62

1 Introduction

1.1 Motivation

The development of specialized software for spatial data analysis has facilitated GIS (geographic information system) [1] specialists to offer different services based on GIS technology for various purposes, e.g., infrastructure monitoring, crime mapping, transportation planning etc [2]. Although getting services from any spatial analysis service provider is simple: the end-user will send the area of interest (AOI) and specialists will work on it. But sometimes collecting, managing or customizing the AOI data in spatial data formats (geojson¹, kml², shp³ etc) is a bit complicated to understand and time-consuming for end-users who are not GIS specialists.

For example, Mr. X from XYZ real estate company is looking for a solution that measures deformations of a real estate area. So now if he needs help from ABC company who are doing research on it, ABC will ask Mr. X to send all geographical position or area of all buildings. So if the total number of buildings is 300 it's enough time-consuming to draw all the building locations on map or find out all 300 buildings' locations from the spatial dataset of thousands of buildings. In these cases, this spatial data customizing tool could be a solution for Mr. X.

Generally, spatial data are stored and transferred over the web in two very distinct data formats: as vector and raster data[3].

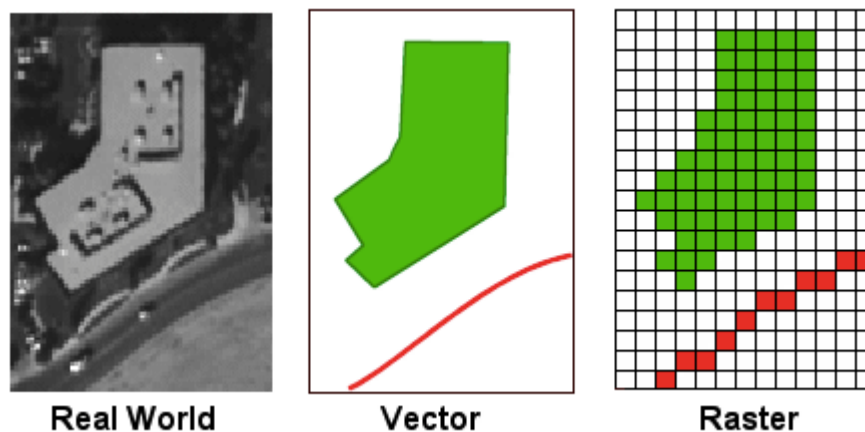


Figure 1. Real-World data representation in Vector and Raster format⁴

In vector data, (Figure 1) each real-world objects i.e., bridge, line, building, etc. are described by a geometry (points, lines, and polygons), a set of attributes that characterize the real-world object and often styling information for presentation of the information. Table 1 presents a typical vector data types for specifying geographic areas of interest in geojson and kml formats.

¹ <https://geojson.org/>

² <https://developers.google.com/kml/>

³ <https://wiki.openstreetmap.org/wiki/Shapefiles>

⁴ <http://www.geography.hunter.cuny.edu/~jochen/GTECH361/lectures/lecture05/concepts/03%20-%20Geographic%20data%20models.html>

Each object (such as a bridge) information represented as Feature in spatial format and FeatureCollection contains all features in an array. Usually, it is complex in nature to read, write or modify.

Table 1. Point information in two different spatial data format

kml	geojson
<pre><?xml version="1.0" encoding="UTF-8"?> <kml xmlns="http://www.open- gis.net/kml/2.2"> <Placemark> <name>Simple place- mark</name> <description>Attached to the ground. Intelligently places itself at the height of the underlying terrain.</description> <Point> <coordinates>- 122.0822035425683,37.422289901 40251,0</coordinates> </Point> </Placemark> </kml></pre>	<pre>{ "type": "FeatureCollection", "features": [{ "type": "Feature", "geometry": { "type": "Point", "coordinates": [-122.0822035425683, 37.42228990140251, 0] }, "properties": { "name": "Simple placemark", "description": "Attached to the ground. Intelligently places it- self\n at the height of the un- derlying terrain." } }] }</pre>

Raster data represents the world as a surface divided into a regular grid of cells or pixel, and each cell has an associated value, such as temperature. Raster data models are useful for storing data that varies continuously, as in an aerial photograph, a satellite image or an elevation surface. It is very easy to modify or program due to a simple data structure.

So, when end-user selects AOI there may be different objects such as buildings, roads, etc and in spatial data, those objects are actually polygons, lines or points. Additionally, the service provider expects AOI information in a vector data format as a bounded box object (in the form of polygons), so that later they will not need to regenerate AOI information in vector format for system input. On the other hand, end-users are also not interested in learning this complex data format for their simple work. Although there are some tools to solve this problem, some of them are more expensive (such as ArcGIS⁵) and free ones (such as geojson.io⁶) are not good enough to cover all the features. So we need a free user-friendly solution to solve the above-mentioned problems.

⁵ <https://www.arcgis.com/index.html>

⁶ <http://geojson.io>

1.2 Problem statement

Users have multiple ways to express their wish to collect AOI information. One is writing all information in a file according to the vector data format or draw on the map using Openlayer [5] API tools and another one is to download those datasets from the world wide web.

First one is too complicated and time-consuming for a non-specialist to write manually; for example, writing location information for 1000 buildings, roads or rail line of a city.

On the other hand, a dataset may include:

- Data for the AOI and other locations as well, so removal of unnecessary location data is needed,
- A collected dataset may be in different data format, e.g. geojson, kml, shp etc. Where end-user is unaware of the data structure of those file format.
- The dataset can include multiple real-world objects in spatial object types (Points, Polygons, Lines, etc.) and these objects can be small, big, overlap and so on.
- Co-ordinates projection for a spatial object may differ in the dataset, e.g. the location at 2754634.938102, 8275407.094152 in Tallinn, Estonia (EPSG:3857) coordinates is 24.77975, 59.3974219 in WGS 84 (EPSG:4326) long/lat. Sometimes co-ordinates need conversion to visualize the correct location of that object on a map [6].

Furthermore, few object information may need to be updated, e.g., road information which one is given in the dataset should be updated because of its recent extension from road administration.

Moreover, due to the highly complex and dynamic nature of GIS systems, vector data may include MultiPoint, MultiLineString, MultiPolygon, Geometry Collection as well.



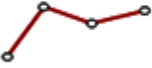

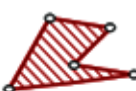


POINT		MULTIPOINT		Tree, Pole, Hydrant, Valve
LINESTRING		MULTILINESTRING		Road, River, Railway, Pipeline
POLYGON		MULTIPOLYGON		Cadastrre, Park, Administrative Boundary
COLLECTION				Graphics, Markups

Figure 2. Vector data types used for specifying Areas of Interest⁷

From this data nature,

MultiPolygon type feature may contain 4-5 polygon but a user needs only 1 or 2 polygon information of those, in this case, I didn't find any existing solution that can help a user

⁷ https://sqlmentalist.files.wordpress.com/2012/06/image_thumb7.png?w=600&h=223

collect information separately. Same for MultiLineString or MultiPoint type feature as well.

GeometryCollection type feature is more complex than the previous one. It contains an array of Point, Polygon, LineString, MultiPolygon, etc. Figure 2 describes different types of geometric type in vector data.

So application needs a custom solution to reformat and convert all MultiPolygons type feature to several Polygons, MultiLineStrings to several LineStrings and MultiPoints to several Points type feature as well.

End-users are not aware of these problems which usually occur in datasets and it is very time-consuming to clean the dataset manually for client and service provider as well.

1.3 Contributions

The goal of this thesis is to develop a free and open source user-friendly web solution for non GIS specialist end-users for performing different geospatial operations [7](e.g., buffer operation [8] for increasing the size of a point or line or polygon in both sides; envelope, concave or convex hull operation for merging nearby point or polygon) in background of the application.

Additionally, the system will take spatial data as input in several widely used spatial data format [9] (e.g., kml, geojson, etc.), so that the end-users can solve above-mentioned problems graphically without any knowledge of the GIS data complexity [10]. For example, the end-user can draw AOI on the map and be able to generate AOI information in GIS format. If the dataset needs cleaning, user can be detecting and correcting (or removing) corrupt or inaccurate data graphically through user interface of that application. Even if the location is too small so that clients need to increase the size of the area, they can do it just by sliding a slide range to increase or decrease the area size around that object.

After processing those data end-users will be able to visualize and check the processed data on the map and it will be available for download in different spatial data formats as well. Additionally, the application will provide the end-user or an external system to access those data from anywhere through a web service interface based on REST[11].

The expected outcome of this application is to provide a graphical solution for the user to perform a different geospatial operation for collecting information for the AOI. That will help users to send clean and specific information regarding AOI to their desired providers.

All geospatial operations of this web-based tool will be performed by an open sourced modular geospatial engine written in JavaScript called turfJs[12] and another open-source JavaScript library called Openlayer for displaying spatial data in web browsers as slippy maps. Besides, an open-source web application framework named Angular[13] will be used to design and develop interactive user interface for this tool.

Validation of spatial operations will be done by checking the spatial data structure and visualization on the map. Evaluation of the application will be done by user feedback and unit testing.

1.4 Organization of the Thesis

- **Software Requirements Specification:** User interfaces, software interfaces, functional and non-functional requirements are specified.
- **Background and Related work:** related application for spatial operation and development technologies are described here.

- **Methodology:** a short overview of different methods of spatial operation and technologies which are used to solve the above mentioned are given.
- **Software Design Description:** Software perspective, user characteristics, use case diagram, and sequence diagrams are displayed.
- **Software Implementation:** reasoning for the choice of technology, dependencies, and development process is described in detail.
- **Software Evaluation:** experimental results and validations of outputs of the developed application are visualized and explained. Also, performance measurements are displayed.
- **Summary**

2 Software Requirements Specification

This part of the thesis describes an in-depth description of spatialdata.io⁸ application. It covers user interface, communication interface, functional and non-functional requirements, technical requirements and logical data requirements of this application.

2.1 External interface requirements

This section describes the input output functionalities for application. Likes, how the user will interact with the application through the User Interface and how the UI will interact with other service providers.

2.1.1 User Interface

User will be able to use this application through any modern web browser and it will be a single page application. If it works with a mobile that would be more user-friendly.

Since it is a single page application⁹, so there will be few options for a user to interact with different feature. When the user will browse the application after putting the URL on the browser, the home page will appear. There should be a map view and upload option for the user to upload spatial data. Also, there should be few more options to navigate to another page for creating new spatial for point of interest or to check user's created and saved data in the list. Now, if a user wants to find a point of interest from existing spatial data they will upload data on application throw upload option. When uploading spatial data is complete, it will visible on the map view. The map view will show different identification for different geometry type. For example, Point feature will not show on the map as like Polygon feature. That will help the user to identify which feature is Point geometry type or Polygon geometry type.

When all data are visible on the map view, user can click on the feature which one wants to edit or save for future and at the same time clicked feature will appear with feature details as text format in a list somewhere on the application. On that list, there would be two options, one to edit that feature immediately and another one to save for the future.

If the user wants to edit that feature immediately then after clicking the Edit button a pop-up window will open with that feature details.

Now there will be an option to add a buffer for that feature, like there would be an option for a user like a slider to increase or decrease the buffer size. Buffer adding means when user will increase or decrease the size that time according to the size (could be 100 meters or 900 meters) Point area will expand from every side to increase its area. This change will appear on the map in real time.

Other option to edit a feature would click on the feature then the user will be allowed to change co-ordinate of that feature. After completing those operation users will be able to download that changed feature details or save for future use.

Additionally, there will be an option for the user so that they can create a new point of interest data after drawing that on Map.

Another option should be there to check the list of what the user saved for future use. Those data should be appearing as a list view so that user can sort by name or search on that list to find easily. Edit, delete or share options should be there to share or edit data.

⁸ <http://spatialdata.io/>

⁹ https://en.wikipedia.org/wiki/Single-page_application

For each map view, there will option to change different layer for map view. That will help the user to find out the precise location.

2.1.2 Communication Interface

There will be few backend service to keep the user data saved, so that they can use it later if they want. So the frontend will interact will backend using internet and exchange data format could be JSON¹⁰ format [14].

2.2 Functional requirements

This section describes (Table 2) all the functional requirements of the application, i.e. how the system should react to particular inputs and how the system should behave in particular situations.

Table 2. Showing use case tables for functional requirements

2.2.1-1	
Use Case ID:	UC01
Use Case Name:	Upload Data to the application
Summary:	The User needs to upload spatial data to edit.
Actor	User
Pre-Conditions:	The system should be online and able to be accessed.
Description: -User selects "Upload Data" option from the Home page. -System will allow users to browse data file. -User uploads data. -System will show that data on map. Network or Internet Connection Error: The application will not be able to access. No or incorrect data: User uploads incorrect and shows an error message	
Post Conditions:	Spatial data will appear on map visually.
2.2.1-2:	
Use Case ID:	UC02
Use Case Name:	Change map view layer

¹⁰ <https://www.json.org/>

Summary:	The user needs to switch map layer between different layer
Actor	User
Pre-Conditions:	The system should be online and able to be accessed.
<p>Description:</p> <ul style="list-style-type: none"> -User browse home page -After loading the home page there will be option to change layer <p>Network or Internet Connection Error: The application will not be able to access. No or incorrect data: N/A</p>	
Post Conditions:	Different layer style look will be appeared on map view.

2.2.1-3:	
Use Case ID:	UC03
Use Case Name:	On click on map feature
Summary:	The user needs to click on their interested feature when multiple features will appear on map
Actor	User
Pre-Conditions:	The system should be online and data already uploaded.
<p>Description:</p> <ul style="list-style-type: none"> -User will click on the visualized feature. -That feature details will be in the list -And Edit and Save icon will be on the List. <p>Network or Internet Connection Error: The application will not be able to access. No or incorrect data: If click on the map there is no feature visible, then it will not add on the List.</p>	
Post Conditions:	List of all the selected feature will appear.

2.2.1-4:

Use Case ID:	UC04
Use Case Name:	Feature Edit Option
Summary:	The user needs to click on Edit icon from feature List item.
Actor	User
Pre-Conditions:	The system should be online and clicked on feature from map view.
<p>Description:</p> <ul style="list-style-type: none"> -User will click on the Edit icon. -A new editing page will open in modal window. <p>Network or Internet Connection Error: The application will not be able to access. No or incorrect data: If click on the map there is no feature visible, then it will not add on the List.</p>	
Post Conditions:	A modal window will open with edit view

2.2.1-5:	
Use Case ID:	UC05
Use Case Name:	Feature Save Option
Summary:	The user needs to click on Save icon from feature List item.
Actor	User
Pre-Conditions:	The system should be online and clicked on feature from map view.
<p>Description:</p> <ul style="list-style-type: none"> -User will click on the Save icon. -That will save that feature details on database <p>Network or Internet Connection Error: The application will not be able to access. No or incorrect data: If click on the map there is no feature visible, then it will not add on the List.</p>	
Post Conditions:	Will show a successful message

2.2.1-6:	
Use Case ID:	UC06

Use Case Name:	Add buffer on feature
Summary:	The user needs to slide buffer slider to add on feature
Actor	User
Pre-Conditions:	The system should be online and clicked on Edit feature icon from list.
<p>Description:</p> <ul style="list-style-type: none"> -User will slide between several range. -That will be added on feature. -User will be able to change on map <p>Network or Internet Connection Error: The application will not be able to access. No or incorrect data: N/A</p>	
Post Conditions:	User can save or download new edited feature

2.2.1-7:	
Use Case ID:	UC07
Use Case Name:	Download feature details
Summary:	The user needs to click on Download button
Actor	User
Pre-Conditions:	The system should be online and clicked on Edit feature icon from list.
<p>Description:</p> <ul style="list-style-type: none"> -User will click on Download button -That will show different option -After clicking on option it will allow user to download on local device <p>Network or Internet Connection Error: The application will not be able to access. No or incorrect data: N/A</p>	
Post Conditions:	User can save feature details data

2.2.1-8:	
Use Case ID:	UC08
Use Case Name:	Save object details

Summary:	The user needs to click on SAVE OBJECT FOR FUTURE button
Actor	User
Pre-Conditions:	The system should be online and edit feature should be opened
<p>Description:</p> <ul style="list-style-type: none"> -User will click on SAVE OBJECT FOR FUTURE button -That will show a save object details on database <p>Network or Internet Connection Error: The application will not be able to access. No or incorrect data: N/A</p>	
Post Conditions:	User can get a confirmation message

2.2.1-9:	
Use Case ID:	UC09
Use Case Name:	Create new object details
Summary:	The user needs to click on CREATE OBJECT from top menu
Actor	User
Pre-Conditions:	The system should be online and browser navigate to /new object page view
<p>Description:</p> <ul style="list-style-type: none"> -User will draw interested area on map -After drawing object details will appear on left side of that page -Download button appear -Save and Clear button should active as well <p>Network or Internet Connection Error: The application will not be able to access. No or incorrect data: If there application didn't get place name on that location Object place name may blank.</p>	
Post Conditions:	User will be able to download, save or clear draw

2.2.1-10:	
Use Case ID:	UC10
Use Case Name:	Search place
Summary:	The user needs to write place name on search text area

Actor	User
Pre-Conditions:	The system should be online and new object page should open
<p>Description:</p> <ul style="list-style-type: none"> -User will write place name on Search option text area -That will show a list of related with that name -User can select their desire location -Map view will zoom in to that place <p>Network or Internet Connection Error: The application will not be able to access. No or incorrect data: Search list will be empty</p>	
Post Conditions:	Map view will zoom in on that place

2.2.1-11:	
Use Case ID:	UC011
Use Case Name:	Object List
Summary:	The user needs to click on Object List menu item from top menu
Actor	User
Pre-Conditions:	The system should be online
<p>Description:</p> <ul style="list-style-type: none"> -User will click on OBJECT LIST menu. -That will navigate to /objectlist page -List of all object should be in a list <p>Network or Internet Connection Error: The application will not be able to access. No or incorrect data: N/A</p>	
Post Conditions:	User will be able to see the list of the object details

2.2.1-12:	
Use Case ID:	UC12
Use Case Name:	Object search
Summary:	The user needs to write name object or id name on search area
Actor	User
Pre-Conditions:	The system should be online and /objectlist page should be opened

<p>Description:</p> <ul style="list-style-type: none"> -User will write object name or id on that area -That will sort object list in table <p>Network or Internet Connection Error: The application will not be able to access. No or incorrect data: List will not sort if does not match anything with that list text.</p>	
Post Conditions:	User will get a sorted list

2.2.1-13:	
Use Case ID:	UC13
Use Case Name:	Update object details
Summary:	The user needs to click on EDIT OBJECT icon from list
Actor	User
Pre-Conditions:	The system should be online and /objectlist page should be opened
<p>Description:</p> <ul style="list-style-type: none"> -User will click on EDIT OBJECT icon -That object details will open in Modal window -From there user can modify object co-ordinates -Then will be able to update information after clicking on UPDATE DETAILS button <p>Network or Internet Connection Error: The application will not be able to access. No or incorrect data: N/A</p>	
Post Conditions:	User can get a confirmation message after saving this information to database

2.2.1-14:	
Use Case ID:	UC14
Use Case Name:	Delete object details
Summary:	The user needs to click on DELETE OBJECT icon from list
Actor	User
Pre-Conditions:	The system should be online and /objectlist page should be opened

Description: -User will click on DELETE OBJECT icon -That object details will delete permanently from list -Object list will update soon. Network or Internet Connection Error: The application will not be able to access. No or incorrect data: N/A	
Post Conditions:	User will get new updated list

2.2.1-15:	
Use Case ID:	UC15
Use Case Name:	Share object details
Summary:	The user needs to click on SHARE button
Actor	User
Pre-Conditions:	The system should be online and /objectlist page should be opened
Description: -User will click on SHARE button -That will show multiple option to user -Download option to download that data -API link to share with company or provider -Raw object details to share Network or Internet Connection Error: The application will not be able to access. No or incorrect data: N/A	
Post Conditions:	User will get a new view with different share option

2.3 Logical data requirements

2.3.1 File format

This application should work with different vector data formats as an input. Also user will be able to download data as vector data formats.

2.4 Non-functional requirements

Non-functional requirements are described on Table 3.

Table 3. Non-functional Requirements

3.4.1: NFR01: User-Friendly	The application ought to have the capacity to give a refined structure that is, a simple to discover and to investigate interface, permitting the user to complete the planned action with no trouble.
3.4.2: NFR02: Accessibility	The application ought to be usable by as many people as possible from any part of the world.
3.4.3: NFR03: Performance	When a request for content is made it should response as soon as possible.
3.4.4: NFR04: Cross Browser Compatibility	The application should behave in the same way across various browsers.
3.4.5: NFR05: Access permission	No Access permission needed for general user
3.4.6: NFR06: Extensibility	The system should be flexible for any further modifications.
3.4.7: NFR07: Compatibility	The application should be compatible to be hosted on any kinds Web Server.
3.4.8: NFR08: Operability	The application should behave in the same way in various operating systems.
3.4.9: NFR09: Error Handling	The application should provide meaningful error messages when errors occur.

3 Background and Related Work

3.1 Area of Interest

Generally, The AOI represents the specific area of a large domain that someone may find useful or interesting for any purpose. This term is widely used in GIS applications, especially in area-based spatial data analysis. User needs to specify AOI to the provider for further analysis.

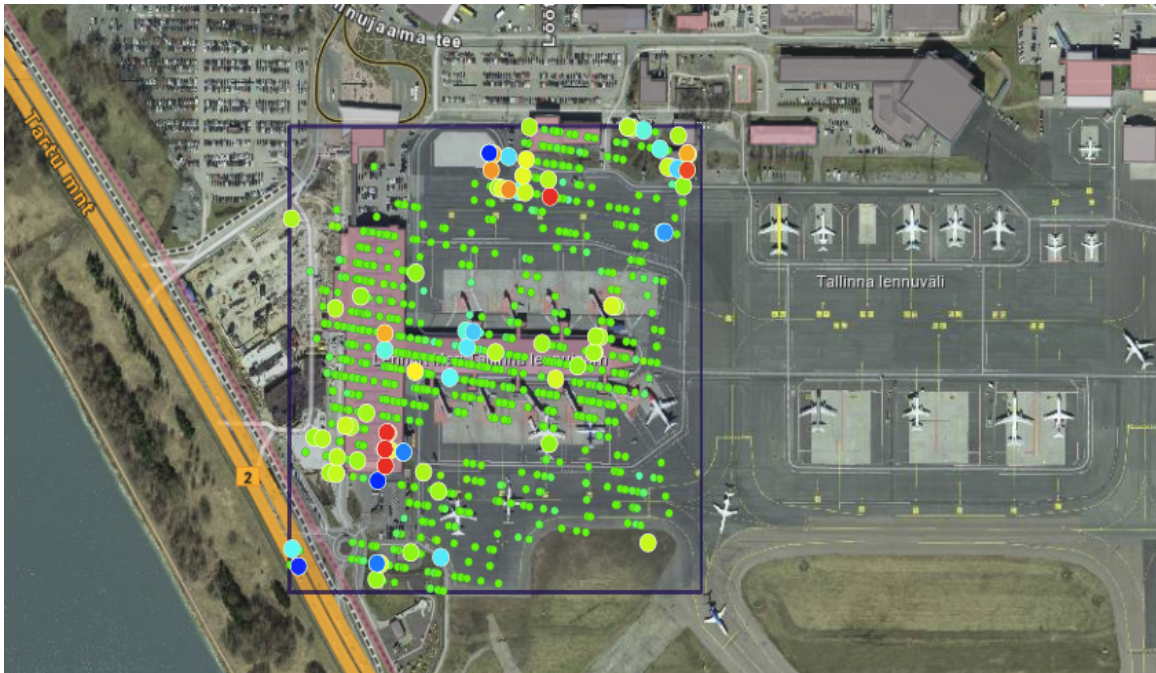


Figure 3. Land deformation analysis result by SILLE¹¹

Figure 3, shows the land deformation information, analysis by SILLE Application of Tallinn Airport¹² area and square bounding box is showing the AOI.

Some examples of types of AOI:

1. Mining area
2. Housing estate area
3. Railway area
4. Nuclear plant area etc.

3.2 Related work

With the exponential growth of world wide web and easy access to satellite data for free, GIS applications have been getting popular. But, it needs more work on those applications to make it user-friendly and useful for the general user. These types of applications are mostly focused on two parts, one is data visualization and the other is spatial analysis after performing the spatial operation based on different geospatial technique.

Various GIS analysis tools have been developed over the years. A very popular GIS application is ArcGIS[15], it is mainly a desktop based application which provides map creation

¹¹ <https://www.sille.space/app/#/map/58/171>

¹² <https://www.tallinn-airport.ee/en/>

and multiple analysis functions. But it needs a good deal of training to work with it. However, it has an online version named ‘ArcGIS Online’ but it mostly focuses on content management and sharing. No online analysis functionalities are available there.

QGIS [16] is another popular application to analyze and edit spatial information, in addition to composing and exporting graphical maps. It is completely desktop-based software, so it is not quite easy to access for a general user.

GIS support has been developed for database management systems as well to support work with spatial data. For example, PostGIS[17] is an open source software that adds support for geographic objects to the PostgreSQL[18] object-relational database. Similarly, SpatiaLite [19] is a spatial extension to SQLite relational database management system. Recently, the most popular database MySQL also added spatial data types in their system to perform the different geospatial operation in database level. Since these functionalities are available for database experts, it is not quite available to the general user until they learn database systems.

On the other hand, few development libraries like Openlayer, leafletjs [20] and turfjs are available, but libraries cannot directly be used by general users, hence the need to develop a free application for them.

So, to visualize and customize spatial data, a general user still needs a free tool with few spatial functionalities, like adding buffer in a line feature, or merging closest point and so on.

4 Methodology

4.1 Used techniques

4.1.1 Visualization

Spatial data visualization is an important task for non GIS specialized person. Representation of objects on the map helps the user to get a clear idea about this object location. As a result of good representation user easily can select their area of interest. OpenLayer is the most popular and completely free, open source library for visualizing spatial data on any dynamic map in any web page. The strengths of Openlayers are many, first of all, in its compliance with the Open Geospatial Consortium (OGC) standards [21], making it capable to work together with all major and most common spatial data servers. Additionally, it supports different tile layers, so user would be able to see his object area in different map tiles like street, aerial etc.

Visualizing the objects in geo-spatial data is as important as the data itself. End user is only concerned about the visualized object. Additionally, this visualization task becomes more challenging with data dimensionality and data structure complexity.

As I mentioned vector data may include different objects as MultiPoint, MultiLineString, MultiPolygon and Geometry Collection as well, so if user needs to work with few lines of MultiLine , so application needs to visualize all lines of Multiline as separate objects



Figure 4. Multi Lines String data representation



Figure 5. Two Lines String data with bounding box representation

```

{ "type": "MultiLineString", "coordinates": [ [ [ [ -105.021443, 39.578057 ], [ -105.021507, 39.577809 ], [ -105.021572, 39.577495 ], [ -105.021572, 39.577164 ], [ -105.021572, 39.577032 ], [ -105.021529, 39.576784 ] ], [ [ -105.019898, 39.574997 ], [ -105.019598, 39.574898 ], [ -105.019061, 39.574782 ] ], [ [ -105.017173, 39.574402 ], [ -105.01698, 39.574385 ], [ -105.016636, 39.574385 ], [ -105.016508, 39.574402 ], [ -105.01595, 39.57427 ] ], [ [ -105.014276, 39.573972 ], [ -105.014126, 39.574038 ], [ -105.013825, 39.57417 ], [ -105.01331, 39.574452 ] ] ] ] }

```

Figure 6. Data representation in spatial data format for MultiLineString type

Figure 4 and Figure 5, present visual representation of MultiLineString type of spatial data, which has four LineString objects and Figure 6 represents as a spatial geojson data format. In this case if user needs to combine 1 (number 1 from Figure 4) and 2 (number 2 from Figure 4) line objects to select for merging, it is not quite possible from this structure.

For this reason, if it is multi-type-object, then restructuring task was one of the challenging parts on this work for visualizing multi-type object in single-type objects. Then the task is done with implementing few customize technique. For spatial data visualization, this system supports multi-point, polygon, line as well as geometry collection.

4.1.2 Analysis

Spatial data analysis is especially used on geographic data. It has many analytic techniques in a different context. Although this application is focused on only for the general use so few general spatial operations are used in this application. This application is using one simple and powerful JavaScript library called turf.js for spatial analysis. It includes traditional spatial operations, helper functions for creating geojson data, and data classification and statistics tools.

Using this application user can perform the following spatial operations

Buffer:

The buffer operation takes two parameters: a buffer distance and the spatial object points, lines or polygons around which the buffer is to be created. It is also referred to as a zone of the specified distance around coverage features. This application is using this feature to extend the size of an object.

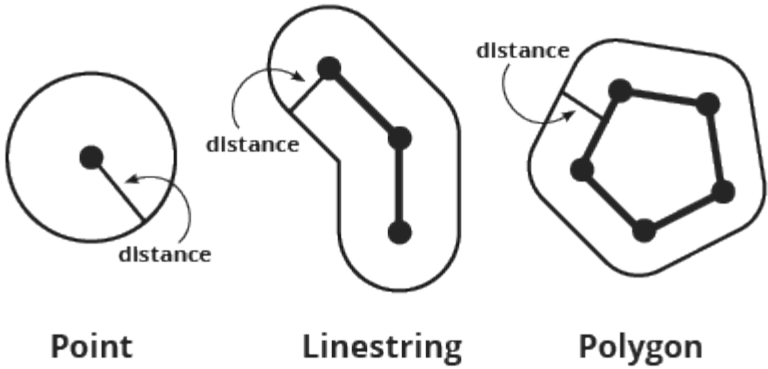


Figure 7. After effects of adding buffer in different object type

Figure 7, shows the result of adding buffer in different object. This application provides user this buffer feature for the following reasons, as for example if the point of interest is too small then the user needs to increase the size of that point or maybe a line of an area is extended over time. Following figure shows the result after performing buffer operation using spatialdata.io application.

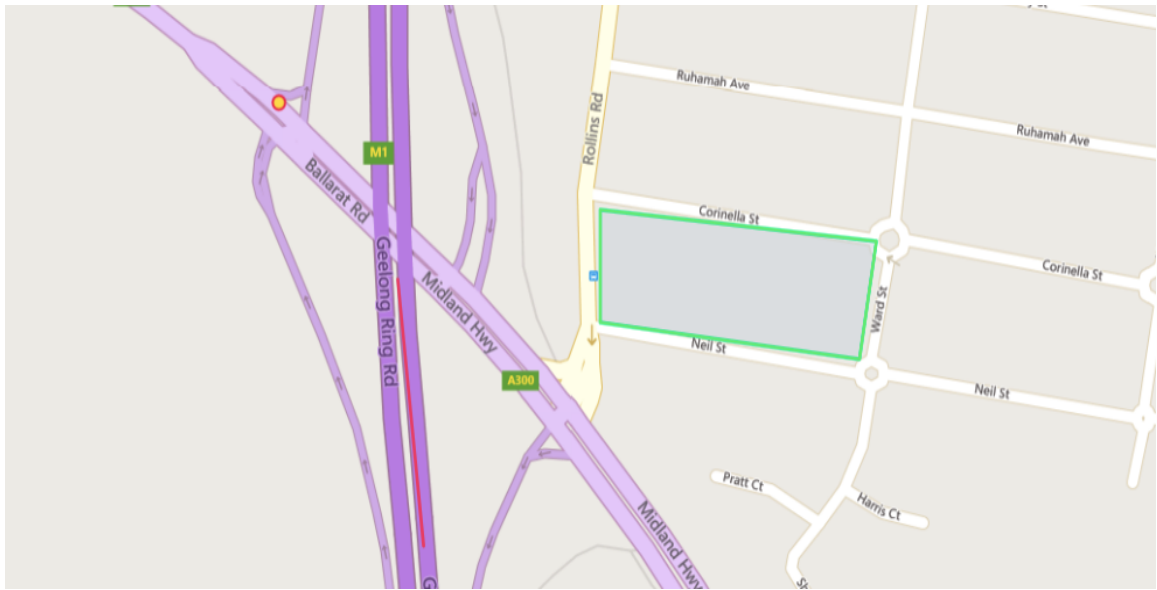


Figure 8. Three different objects from real data



Figure 9. The results representation of after adding buffer using spatialdata.io

Figure 8, shows the objects, and the result of adding the buffer on those objects are shown in Figure 9.

Convex Hull:

Computing a convex hull is one of the first sophisticated geometry algorithms, and it is popular for determining the smallest convex set (called the 'convex hull') containing a discrete set of points. There are numerous applications for convex hulls: collision avoidance, hidden object determination, and shape analysis to name a few. This application uses two techniques to merge closet points or lines or polygons, one of them is Convex Hull based on Monotone chain [22] algorithm.

Andrew's monotone chain convex hull algorithm constructs the convex hull of a set of 2-dimensional points in $O(n \log n)$ time. It does so by first sorting the points lexicographically (first by x-coordinate, and in case of a tie by y-coordinate), and then constructing upper and lower hulls of the points in $O(n)$ time.



Figure 10. Objects merging using convex hull technique

Closest points merge in an area now, output result shown on Figure 10.

Bounding Box:

A Bounding box (usually shortened to bbox) [23] is an area that is around different objects that are being used to analysis purpose. This bbox area is defined by two longitudes and two latitudes, where:

Latitude is a decimal number between -90.0 and 90.0.

Longitude is a decimal number between -180.0 and 180.0.

They usually follow the standard format of:

bbox = left, bottom, right, top

bbox = min Longitude, min Latitude, max Longitude, max Latitude



Figure 11. Objects merging using bounding box technique

Closest points merge in an area now using Bounding box technique, output result shown on Figure 11.

4.2 Used technologies

Bootstrap

Bootstrap [24] is an open-source JavaScript framework developed by the team at Twitter. Mostly used to design mobile-friendly responsive user interface. It has a free collection of several tools for creating a web application. Why I used bootstrap

- Easy to get started e.g. download dependency and start working
- Great grid system e.g. col-md-12, col-lg-6
- Base styling for most HTML elements e.g. Typography, Code, Tables, Forms, Buttons, Images, Icons etc.
- Extensive list of components
- Bundled Javascript plugin
- It is supported by all popular browsers
- Well written Documentation

In this application I used latest bootstrap version 4 and user can use our application through any mobile devices as well.

Node JS

NodeJS [25] is a server-side solution for JavaScript, particularly for handling HTTP (REST Services) requests and responses. It works especially well with Single Page Applications (SPAs) running on a client such as a laptop, mobile phone, or refrigerator. (Yes, I said “refrigerator” because Node.js works great with embedded applications too — the so-called Internet of Things, or IoT¹³.) Node.js is perfectly suited to the growing number of micro-

¹³ https://en.wikipedia.org/wiki/Internet_of_things

services architectures across the web, whereby a client can access multiple services simultaneously running on many different servers. Popular NodeJS framework are ExpressJS¹⁴, MeteorJS¹⁵, SailsJS¹⁶ etc. Why I used NodeJS?

- It provides asynchronous event-driven I/O API's which allows handling thousands of concurrent requests.
- It is written in Javascript and so extremely easy to learn if someone already know Javascript.
- We can use the same programming language in server-side and client side for web application.
- It is well documented and supported by an enthusiastic community. e.g. Google¹⁷, Microsoft¹⁸, Linux¹⁹, PayPal²⁰ etc
- Streaming big files will be extremely easy by using NodeJS. You can load multiple files at the same time
- It is fast comparing to other server-side programming languages like Ruby, Python, PHP etc.

In this application I used this technology because if user needs then I can provide facilities to save their data on database or if external system needs to use processed data throw REST service [27].

Angular

Angular is mainly maintained by Google[13]. Usually, developers use this library to create single page applications (SPA). Every couple of months Google comes up with a new version—from Angular 1 through 7. Key features of angular are given below [28]

- Detailed documentation: Angular boasts detailed documentation where developers can find all necessary information.
- Support by Google: A lot of developers consider Google support another benefit of Angular, making the platform trustworthy.
- Great ecosystem of third-party components: The popularity of Angular has resulted in the appearance of thousands of additional tools and components that can be used in Angular apps. As a result, you can get additional functionality and productivity improvements.
- Component-based architecture: In the second version, Angular shifted from an MVC to a component-based architecture. According to this architecture, an app is divided into independent logical and functional components.
- CLI : It is probably the most beloved feature for the majority of Angular developers. It automates the whole development process making app initialization, configuration, and development as easy as possible.

¹⁴ <https://expressjs.com/>

¹⁵ <https://www.meteor.com/>

¹⁶ <https://sailsjs.com/>

¹⁷ <https://www.google.com/>

¹⁸ <https://www.microsoft.com/et-ee/>

¹⁹ <https://www.linux.org/>

²⁰ <https://www.paypal.com/us/home>

Angular Material. This collection of Material Design elements optimized for Angular lets developers easily integrate UI components.

Turf JS

Turf is a modular geospatial engine written in JavaScript for spatial analysis. It includes traditional spatial operations, helper functions for creating geojson data, and data classification and statistics tools. Turf is supporting to use as a client-side plugin, or server-side with Node.js.

In our application I used this library to process spatial data. e.g. converting point into polygon.

PostgreSQL

PostgreSQL is a general purpose and object-relational database management system, the most advanced open source database system. PostgreSQL is free and open source software. Its source code is available under PostgreSQL license, a liberal open source license. You are free to use, modify and distribute PostgreSQL in any form. PostgreSQL requires very minimum maintained efforts because of its stability. Therefore, if you develop applications based on PostgreSQL[18], the total cost of ownership is low in comparison with other database management systems.

Many companies have built products and solutions using PostgreSQL. Some featured companies are Apple, Fujitsu, Red Hat, Cisco, Juniper Network, etc.

5 Software Design Description

The Software Design Description is an important document for software development lifecycle. This SDD describes the detailed design and structure of the spatialdata.io and the implementation details required to satisfy the requirements as specified in the Software Requirements Specification.

5.1 Software perspective

This application consists of two parts, Frontend, and Backend. Most functionalities are in backend. The backend only works for rest API service for frontend.

The Frontend will be used to upload data and perform all editing functionalities and Backend service will be used only if the user wants to save details in the database. However, the Backend Application doesn't have a user interface and neither user nor administrator interacts with it directly.

When a user wants to save object details in the database then frontend application sends a request to backend through REST API service. Although there is no interface for the backend through rest API Services provided by backend system and interactions between these systems are done through an Internet connection. As few parts of this application are data-centric, it uses the database to store all application's data. The database is located in a database server. The frontend part can't interact or fetch data from the database directly. The database will be connected only to the backend system and all the CRUD operations will happen with the help of the backend system.

The Backend Application is a RESTful API service [27] application that has a connection with the database and whole business logic is done in it. A summary is User interacts with Frontend to perform all operation and frontend will take care to interact with the backend if applications needs. The Backend Application will be running on a web server.

Applications interactions and running environments are shown on Figure 12.

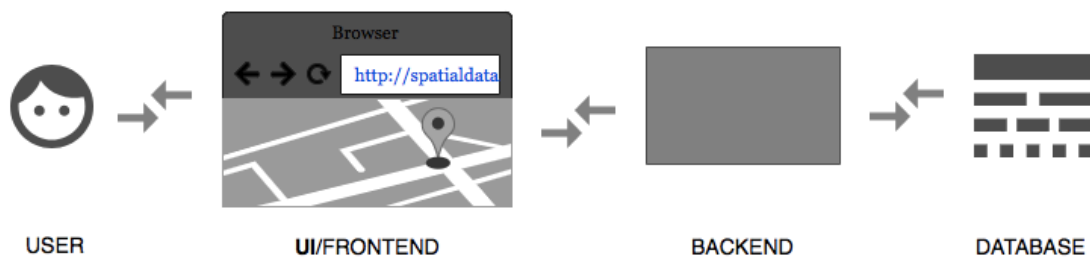


Figure 12. Block diagram of SpatialData.io application

5.2 Software functions

Using this application user will be able to create and edit spatial data features after uploading existing data on that application. Another way they can create new spatial data after drawing an object on map.

Additionally, this application allows the user to save that information for future use and can perform other functionalities like to edit, delete as well.

5.3 User characteristic

Basically for now only one type of user available for this application. They are allowed to perform all operations as described in functional requirements.

5.4 Constraints

Internet connection is also an important part for the system. Since all two parts of the system use the Internet to interact with each other, it is crucial that there is an Internet connection for the system to allow the user to perform all operations.

5.5 Use Case diagram

Figure 13 shows the use case diagram for spatialdata.io.

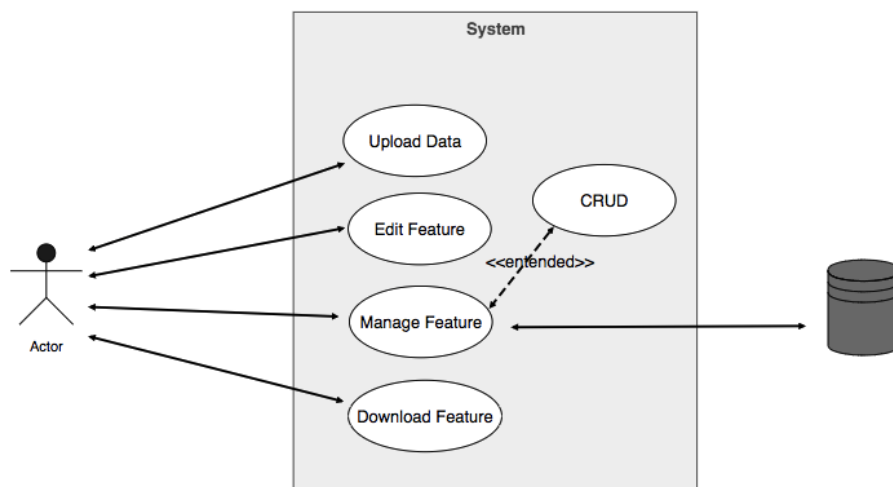


Figure 13. Use Case diagram of spatialdata.io application

5.6 Sequence diagram

Figure 14 shows the sequence diagram for functional requirement UC 01. Similarly, Figure 15 for UC 04, 05,06, Figure 16 for UC 13,14 and Figure 17 for UC 09.

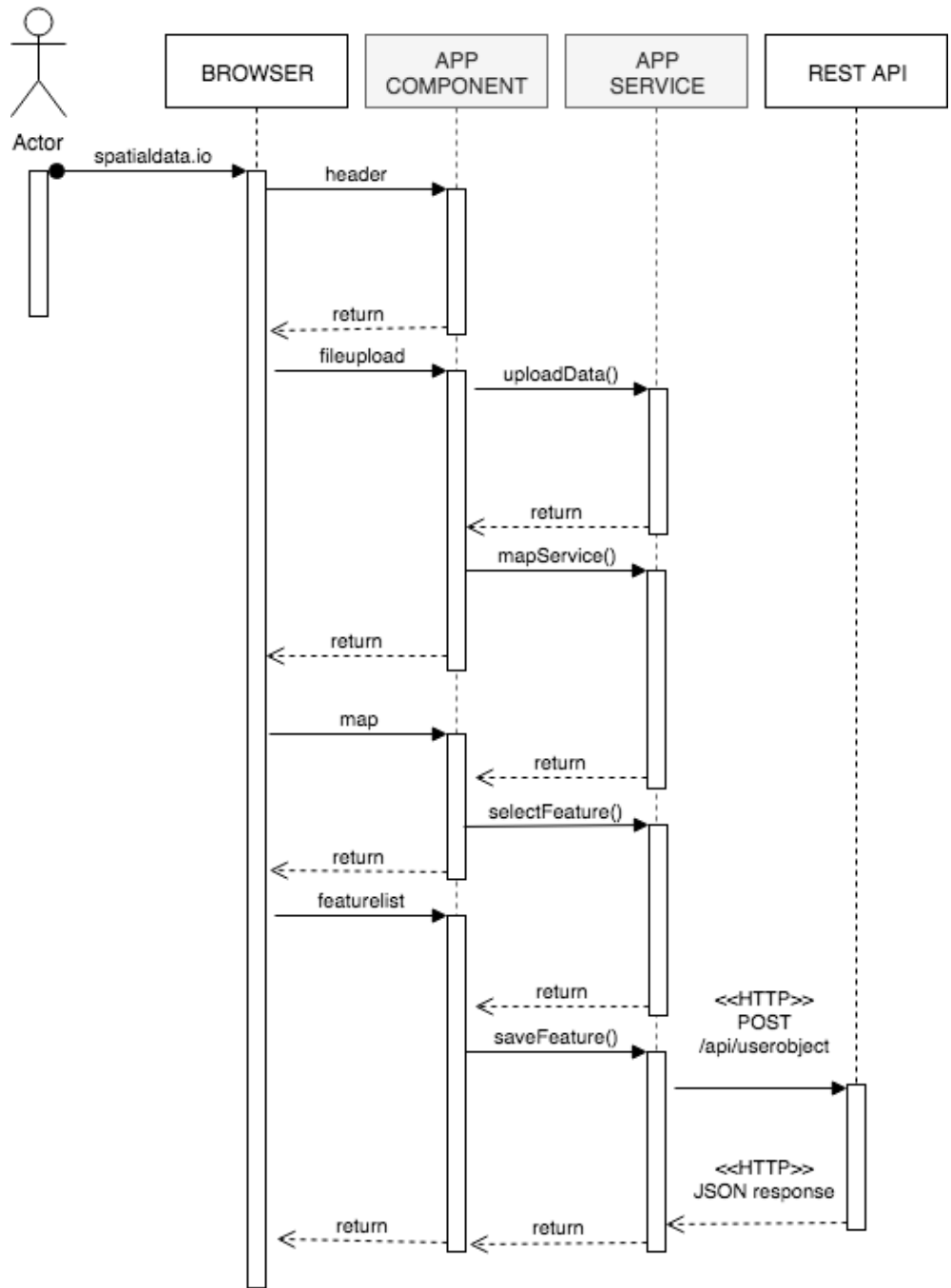


Figure 14. Sequence diagram for uploading data on spatialdata.io

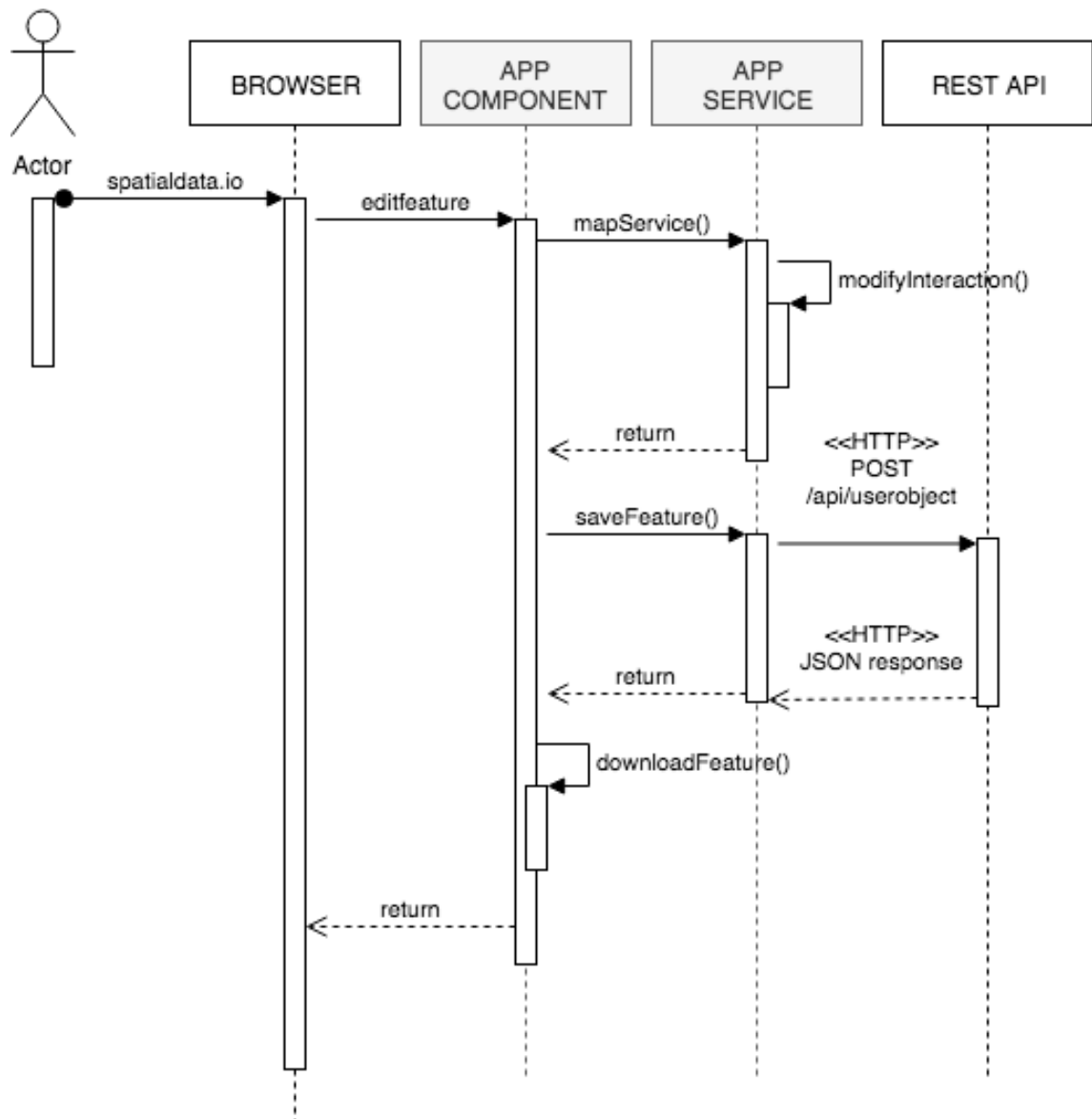


Figure 15. Sequence diagram for Editing feature on spatialdata.io

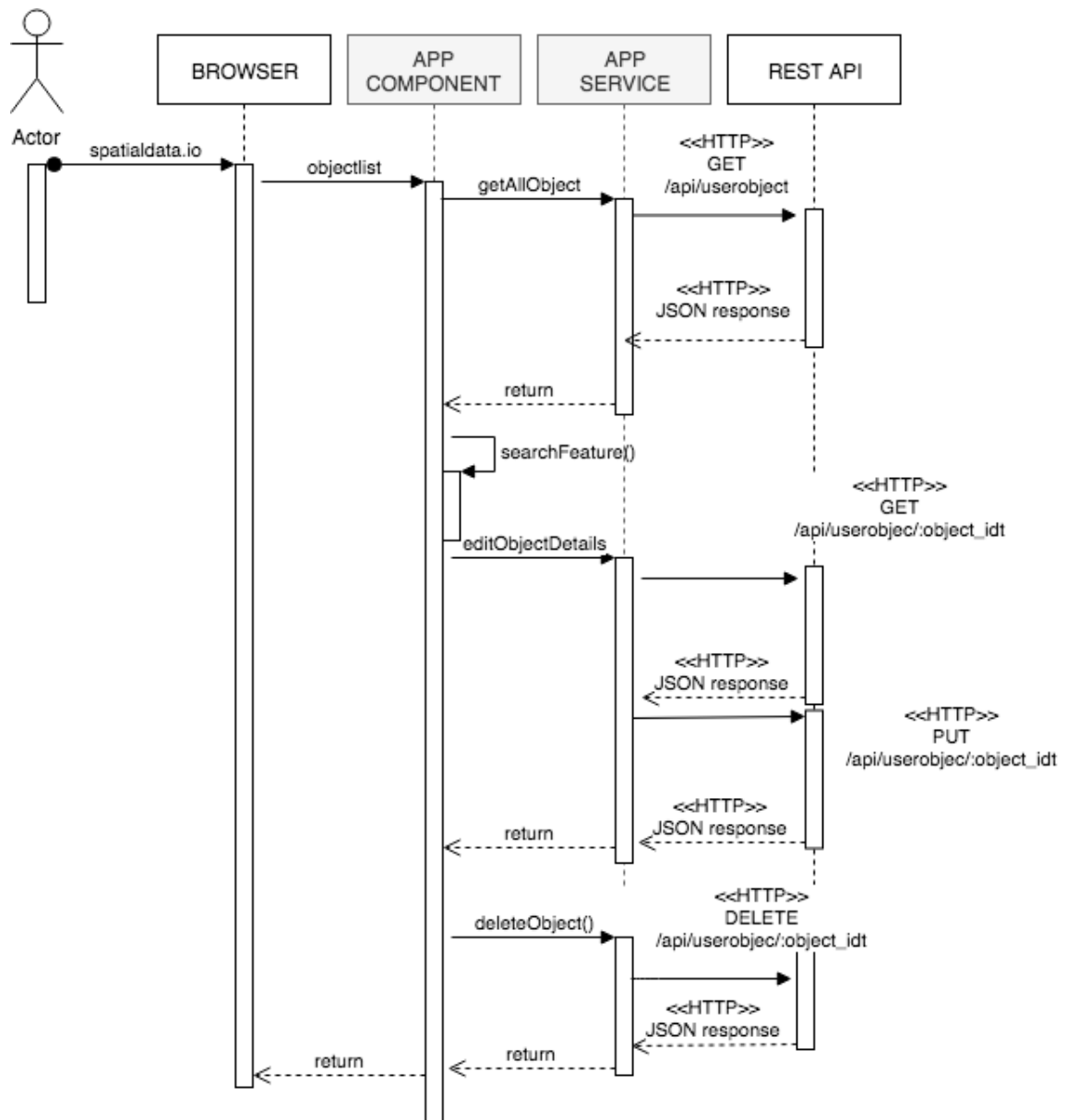


Figure 16. Sequence diagram for update feature on spatialdata.io

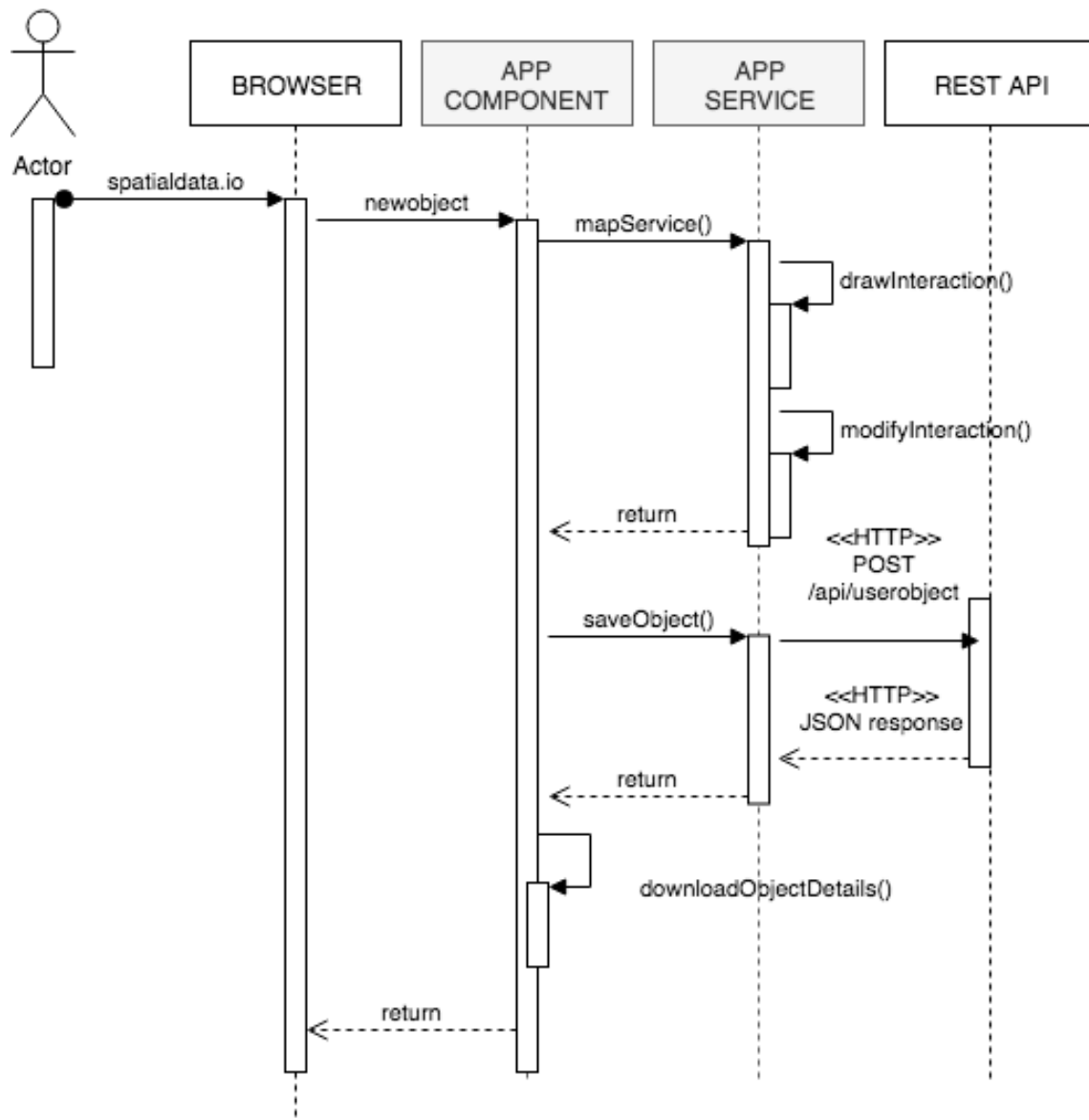


Figure 17. Sequence diagram for Create a new object details on spatialdata.io

6 Software Implementation

6.1 Choice of technologies for development

Because of so many solutions, products and providers, choosing a technology stack, or a combination of software languages, frameworks, tools, used to develop a project is an overwhelming and challenging task. Although the main thing is that the application should perform correctly, and the choice of technology is up to the developer dealing but sometimes we have to think about project cost as well.

This section discusses about the choosing of different technology after taking the all consideration of Non-functional requirements like extensibility, performance, cross-platform etc.

6.1.1 Choice of technology for Front-end development

Front-end, simply the visible parts of a web application. Where different components enable the users to interact with different feature of an application. In the context of web apps, such interaction occurs using browser.

Mostly following three parts are controlling the frontend development.

- HTML (Hypertext Markup Language)²¹ is describing the structure of the information presented in the browser
- CSS (Cascading Style Sheets) is responsible the style of the application content (colours, fonts, and layouts)
- JavaScript empowering the interactivity of the web page

Additionally, there are many HTML and CSS frameworks are available for rapid development. Designing reliable, brief, and efficient CSS can be quite a lot of work. There are so many things to consider like responsiveness, accessibility, and structure. This is exactly why HTML and CSS Frameworks are useful. To mention a few: Bootstrap, Materialize CSS²², Bulma²³, Semantic UI²⁴, Foundation²⁵ etc.

For my project, I choose Bootstrap because it helps to work faster for designing user friendly interface based on our requirements.

Furthermore, as we know there are few JavaScript frameworks, i.e. angular, react²⁶, vue²⁷ as well for single page web application. I took angular because it is an open source framework and it provides a rich variety of resources to build a single page web application based on our requirements. Additionally, it has more flexibility for further modification in future.

6.1.2 Choice of technology for Back-end development

Back-end is the unseen part of a web application. It works like “under the hood” where user can’t access directly but they use from Front-end. Although back-end is invisible to users,

²¹ <https://developer.mozilla.org/en-US/docs/Web/HTML>

²² <https://materializecss.com/>

²³ <https://bulma.io/>

²⁴ <https://semantic-ui.com/>

²⁵ <https://foundation.zurb.com/>

²⁶ <https://reactjs.org/>

²⁷ <https://vuejs.org/>

but it forms the mechanism that drives the application and implements its business logic. The main parts of the backend are

Web server: who are responsible for processing requests from the browser and returning the corresponding data

Database: Used to store the application data

Programming language: used to create the business logic for application

Web development framework: Helped developer for faster and easier development. The web framework directly depends on the choice of the programming language

Since there are so many frameworks based on programming language are available nowadays. Likes, Spring (Java²⁸), .net(c#²⁹), django(python³⁰), node js(js), ruby on rails(ruby³¹) etc

I choose lightweight, efficient node js framework, where I implemented all business logic using expressjs³² framework. While it is a JavaScript framework that helps me to reuse and share the code between the frontend and the backend parts of this application, which speeds up the development process.

At last I choose PostgreSQL because it has better functionalities to work with GIS application. That will help me in future to extend this application with more functionalities.

6.2 Software development

6.2.1 Front-end

Before start development, organizing a development environment is important. Here, I used Visual Studio code as an editor. Since, frontend of this application is based on angular, we have to make sure our development environment includes Node.js and node package manager(npm³³). Node.js is a JavaScript runtime environment includes everything we need to execute a program written in JavaScript.

After installing Node and npm, next job is to install angular cli using npm from Command prompt of PC.

Basic command is like the following command,

- `npm install -g @angular/cli`

Now with the help of angular cli, we can create project and can add other dependencies. Most important package manager along with angular what spatialdata.io used to develop are given below,

- `"@turf/turf": "^5.1.6"` – an advanced geospatial analysis engine written in JavaScript for browsers and Node.js
- `"bootstrap": "^4.3.1"`,
- `"geojson-validation": "^0.2.1"`

²⁸ <https://www.java.com/en/>

²⁹ [https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))

³⁰ <https://www.python.org/>

³¹ <https://www.ruby-lang.org/en/>

³² <https://expressjs.com/>

³³ <https://www.npmjs.com/>

- "jquery": "^3.4.0",
- "ol": "^5.3.2" – package for use openlayer features.
- "togejson": "^0.16.0" - convert kml data to geojson format
- "tokml": "^0.4.0" - convert geojson data to tokml
- "karma-jasmine": "~1.1.1" – for frontend automated test.

6.2.2 Back-end

Since we are using same node js technology for backend, we need to add another framework called express js. Express [30] is a minimal and flexible Node.js web application framework that provides a robust set of features to develop backend logic to support rest api service for frontend.

Simply according to the upwork [28]

Express.js and Node.js gave JavaScript newfound back-end functionality—allowing developers to build software with JavaScript on the server side for the first time. Express.js is used by Fox Sports³⁴, PayPal, Uber³⁵, IBM³⁶ and many more companies.

And what we more packages used to develop backend are given below,

- "bluebird": "^3.3.4" - Bluebird³⁷ is a fully-featured Promise library for JavaScript that allows you to "promisify" other Node modules in order to use them asynchronously.
- "body-parser": "~1.13.2" – Used to extracts the entire body portion of an incoming request stream and exposes it on req.body as something easier to interface with.
- "pg-promise": "^3.2.3" - pg-promise is a Postgres library for Node written on top of the brian/node-postgres library. It's technically a wrapper for node-postgres and converts the call-back interface into a promises based interface.

Following library are used to test automated test for backend service.

- "chai": "^4.2.0",
- "chai-http": "^4.3.0",
- "mocha": "^6.1.4"

6.2.3 Source code of the projects

Spatialdata.io frontend source: <https://github.com/pskpartha/spatialdata.io/>

Spatialdata.io backend source: <https://github.com/pskpartha/upload-back-sille>

³⁴ <https://www.foxsports.com/>

³⁵ <https://www.uber.com/ee/et/>

³⁶ <https://www.ibm.com/ee-en>

³⁷ <http://bluebirdjs.com/docs/getting-started.html>

7 Software Evaluation

This section details all the manually designed and performed test cases result, those are mentioned in functional and non-functional requirements section.

7.1 Acceptance test for functional requirements

Table 4 describes the result of functional requirements.

Table 4. Acceptance test result tables

7.1-1					
Test Case ID:	UC01				
Test Case Name:	Upload Data				
Description:	The User needs to upload spatial data to edit.				
Step	Action	Specific Input/Action	Expected System Response	Received Output	Pass/Fail
1	The users click on Upload button	Upload button clicked	Browse window should be opened	Browse file window opened	pass
2	User browse spatial data file	Clicked on Open button from browser window	Data should on map	Uploaded data displayed on map view	pass
Post Conditions:	User data uploaded successfully and it shows on map				
Test Case Status:	pass				

7.1-2	
Test Case ID:	UC02
Test Case Name:	Change map view layer
Description:	The user needs to switch map layer between different layer

Step	Action	Specific Input/Action	Expected System Response	Received Output	Pass/Fail
1	The users browse Home page	Clicked on ROAD option	Map view layer should change	Map layer changed	pass
2	Clicked on OPEN-STREET option	Clicked on OPEN-STREET	Layer changed	Layer changed to openstreet map view	pass
Post Conditions:	User changed layer option successfully and it shows on map				
Test Case Status:	pass				

7.1-3					
Test Case ID:	UC03				
Test Case Name:	On click on map feature				
Description:	The user needs to click on their interested feature when multiple features will appear on the map				
Step	Action	Specific Input/Action	Expected System Response	Received Output	Pass/Fail
1	The users clicked on the map feature	Featured clicked	Featured clicked listen and add on the list	Feature details added on the list	pass
2	Clicked on Another feature on the map	Another feature clicked	Featured clicked listen and add on the list	One more feature added on the list	pass
Post Conditions:	User clicked and added those feature on the list				

Test Case Status:	pass
-------------------	------

7.1-4					
Test Case ID:	UC04				
Test Case Name:	Feature Edit Option				
Description:	The user needs to click on the Edit icon from feature List item.				
Step	Action	Specific Input/Action	Expected System Response	Received Output	Pass/Fail
1	The users clicked on the Edit icon from feature list	Edit Icon clicked	Modal window opened	Modal window opened with editing options.	pass
Post Conditions:	New modal window opened with editing options				
Test Case Status:	pass				

7.1-5					
Test Case ID:	UC05				
Test Case Name:	Feature Save Option				
Description:	The user needs to click on the Save icon from feature List item to save an item on the database for future use.				
Step	Action	Specific Input/Action	Expected System Response	Received Output	Pass/Fail

1	The users clicked on the Save icon from feature list	Save icon clicked	Snackbar will open with a successful message.	Snackbar opened with a successful message	pass
Post Conditions:	New item add in the database				
Test Case Status:	pass				

7.1-6					
Test Case ID:	UC06				
Test Case Name:	Add buffer on feature				
Description:	The user needs to slide buffer slider to add on feature from edit feature page				
Step	Action	Specific Input/Action	Expected System Response	Received Output	Pass/Fail
1	The users slide on slider to increase or decrease to add buffer area	Slide on the slider	Map feature area will extend or shrink	Map feature extended	pass
Post Conditions:	Feature details updated on map				
Test Case Status:	pass				

7.1-7	
Test Case ID:	UC07
Test Case Name:	Download feature details
Description:	The user needs to click on Download button

Step	Action	Specific Input/Action	Expected System Response	Received Output	Pass/Fail
1	The users clicked on Download button	Download button clicked	Few options should open	Few download options opened	pass
2	Click on the option	Option clicked	Option clicked and download will start with a moment	Open clicked and downloaded	pass
Post Conditions:	Download complete				
Test Case Status:	pass				

7.1-8					
Test Case ID:	UC08				
Test Case Name:	Save object details				
Description:	The user needs to click on SAVE OBJECT FOR FUTURE button				
Step	Action	Specific Input/Action	Expected System Response	Received Output	Pass/Fail
1	The users clicked on the SAVE OBJECT FOR FUTURE from modal window	Button clicked	Snackbar will open with a successful message.	Snackbar opened with a successful message.	pass
Post Conditions:	Snackbar opened with a successful message.				
Test Case Status:	pass				

7.1-9	
Test Case ID:	UC09
Test Case Name:	Create new object details

Description:	The user needs to click on CREATE OBJECT from top menu				
Step	Action	Specific Input/Action	Expected System Response	Received Output	Pass/Fail
1	The users clicked on CREATE OBJECT from top menu	CREATE OBJECT from top menu clicked	New page will open with creating option	New page opened	pass
2	Draw on the map	Draw an object area on the map	Drawn area will be visible on the map	New area drawn by user	pass
3	Click on the Save option to save this object details	Clicked on the save option	A successful message will appear	Successful message appeared	pass
4	Click on the Download option to download new object details	Clicked on the download option	Will start download	Newly created object details downloaded	pass
Post Conditions:	User abled to create new object details				
Test Case Status:	pass				

7.1-10					
Test Case ID:	UC10				
Test Case Name:	Search place				
Description:	The user needs to write place name on search text area				
Step	Action	Specific Input/Action	Expected System Response	Received Output	Pass/Fail
1	User writes place name on	Place name written	Show list of place	Place list showed	pass

	Search option text area				
2	User clicks on the name from the place list	List item listened	Map will zoom in to that place	Map view changed to that place	pass
Post Conditions:	New place area shown on the map				
Test Case Status:	pass				

7.1-11					
Test Case ID:	UC11				
Test Case Name:	Object List				
Description:	The user needs to click on Object List menu item from top menu				
Step	Action	Specific Input/Action	Expected System Response	Received Output	Pass/Fail
1	The users clicked on the Object List from the menu	Clicked on the Object List menu item	New page will open with a list of all saved object	New page opened with a list of saved object	pass
Post Conditions:	New page opened with a list				
Test Case Status:	pass				

7.1-12					
Test Case ID:	UC12				
Test Case Name:	Object search				
Description:	The user needs to write name object or id name on search area				
Step	Action	Specific Input/Action	Expected System Response	Received Output	Pass/Fail

1	The user writes name on the search area	Search area listened	List will sort according with the search input text	List sorted and matched with the name	pass
2	The user write number on the search area	Search area listened	List will sort according with the search input text	List sorted and matched with the id	pass
Post Conditions:	Sorted list will appear				
Test Case Status:	pass				

7.1-13					
Test Case ID:	UC13				
Test Case Name:	Update object details				
Description:	The user needs to click on EDIT OBJECT icon from list				
Step	Action	Specific Input/Action	Expected System Response	Received Output	Pass/Fail
1	The users clicked on the Edit Object icon from object list	Edit Icon clicked	Modal window opened	Modal window opened with editing options.	pass
Post Conditions:	New modal window opened with editing options				
Test Case Status:	pass				
7.1-14					
Test Case ID:	UC14				
Test Case Name:	Delete object details				
Description:	The user needs to click on DELETE OBJECT icon from list				

Step	Action	Specific Input/Action	Expected System Response	Received Output	Pass/Fail
1	The users clicked on the Delete Object icon from object list	Delete Icon clicked	That object details will delete permanently from list	That object details deleted from list	pass
Post Conditions:	Item deleted from the object list				
Test Case Status:	pass				

7.1-15					
Test Case ID:	UC15				
Test Case Name:	Share object details				
Description:	The user needs to click on SHARE button				
Step	Action	Specific Input/Action	Expected System Response	Received Output	Pass/Fail
1	The User clicks on SHARE button	SHARE button clicked	Multiple download option should open	Multiple download options opened for data sharing	pass
Post Conditions:	Download feature options opened				
Test Case Status:	pass				

7.2 Automated tests

Automated tests are integral part of any contemporary software project. It saves time compared to manual testing. But building a successful automated testing strategy is tough and

the approach will vary on a project basis. Here this project is tested by Jasmine³⁸ and Karma³⁹ framework. Where Jasmine works to describe tests in a human readable format so that non-technical people can understand what is being tested. After updating each test, it is boring to refresh the browser, Karma is used to check change and refresh if detect any changes.

Table 5 shows the information about automated test for spatialdata.io. Here I randomly tested few functionalities. For clarification,

- when decision is 'pass', it means function is working fine because it is receiving input as expected.
- when decision is 'fail', it means function is working fine too because it is tested with wrong input data.

Table 5. Automated Service Test tables

7.2.1-1:				
Test Case Name:	Should return object list			
Class Name	ApidataService			
Case	Input	Expected	Actual	Decision
1	expect(data.success).toBe(true)	true	true	pass
2	expect(data.data.length).not.toBe(null)	not null	not null	pass
3	expect(req.request.method).toEqual('GET')	GET	GET	pass
4	expect(req.request.method).toEqual('POST')	GET	POST	fail

7.2.1-2:				
Test Case Name:	Should return single object details			
Class Name	ApidataService			
Case	Input	Expected	Actual	Decision

³⁸ <https://jasmine.github.io/>

³⁹ <https://karma-runner.github.io/4.0/index.html>

1	expect(data.success).toBe(true)	true	true	pass
2	expect(data.data.length).not.toBe(null)	not null	not null	pass
3	expect(req.request.method).toEqual('GET')	GET	GET	pass
4	expect(req.request.method).toEqual('POST')	GET	POST	fail
5	expect(req.request.url).toEqual(endpoint+"/2");	end-point/2	end-point/2	pass

7.2.1-3:				
Test Case Name	Should add object in list			
Class Name	ApidataService			
Case	Input	Ex-pected	Actual	Deci-sion
1	expect(data.success).toBe(true)	true	true	pass
2	expect(req.request.method).toEqual('GET')	POST	GET	fail
3	expect(req.request.method).toEqual('POST')	POST	POST	pass

7.2.1-4:				
Test Case Name	Should update an object details			
Class Name	ApidataService			
Case	Input	Ex-pected	Actual	Deci-sion
1	expect(data.success).toBe(true)	true	true	pass
2	expect(req.request.method).toEqual('GET')	PUT	GET	fail
3	expect(req.request.method).toEqual('PUT')	PUT	PUT	pass

7.2.1-4:				
Test Case Name	Should update an object details			
Class Name	ApidataService			
Case	Input	Ex-pected	Actual	Deci-sion
1	expect(data.success).toBe(true)	true	true	pass
2	expect(req.request.method).toEqual('GET')	PUT	GET	fail
3	expect(req.request.method).toEqual('PUT')	PUT	PUT	pass

7.2.1-5:				
Test Case Name	Should delete an object details			
Class Name	ApidataService			
Case	Input	Ex-pected	Actual	Deci-sion
1	expect(data.success).toBe(true)	true	true	pass
2	expect(req.request.method).toEqual('GET')	DE-LETE	GET	fail
3	expect(req.request.method).toEqual('DELETE')	DE-LETE	DE-LETE	pass

7.2.1-6:				
Test Case Name	convertToPolyGon() should convert Point or LineString to Polygon			
Class Name	DataService			
Case	Input	Expected	Actual	Deci-sion
1	expect(input.geometry.type).not.toEqual(result.geometry.type);	Polygon	Polygon	pass

2	expect(input.geometry.type) .toEqual(result.geometry.type);	Point	Polygon	fail
3	expect(input.geometry.type) .toEqual(result.geometry.type);	Lin- eString	Polygon	fail

7.2.1-7:				
Test Case Name	convertToPolyGon() should convert Polygon to Polygon			
Class Name	DataService			
Case	Input	Expected	Actual	Deci- sion
1	expect(input.geometry.type) .toEqual(result.geometry.type);	Polygon	Polygon	pass

7.2.1-8:				
Test Case Name	multiPointToPoints() should convert MultiPoints->Point, Point ..			
Class Name	DataService			
Case	Input	Expected	Actual	Deci- sion
1	expect(input.geometry.type).not .toEqual(output[0].geometry.type);	Point	Point	pass
2	expect(input.geometry.type) .toEqual(output[0].geometry.type);	Mul- tiPoint	Point	fail

7.2.1-8:				
Test Case Name	multiLineToLines() should convert MultiLineString->LineString, LineString .			
Class Name	DataService			
Case	Input	Expected	Actual	Decision
1	expect(input.geome- try.type).not	LineString	Lin- eString	pass

	.toEqual(output[0].geometry.type);			
2	expect(input.geometry.type).toEqual(output[0].geometry.type);	MultiLineString	LineString	fail

7.2.1-9:				
Test Case Name	multiPolygoTopolygons() should convert MultiPolygon->Polygon,Polygon .			
Class Name	DataService			
Case	Input	Expected	Actual	Decision
1	expect(input.geometry.type).not .toEqual(output[0].geometry.type);	Polygon	Polygon	pass
2	expect(input.geometry.type).toEqual(output[0].geometry.type);	MultiPolygon	Polygon	fail

7.2.1-10:				
Test Case Name	geometryCollectionToSimple() should convert GeometryCollection->Point,Polygon,LineString .			
Class Name	DataService			
Case	Input	Expected	Actual	Decision
1	expect(input.geometry.type).not .toEqual(output[0].geometry.type);	Polygon	Polygon	pass
2	expect(input.geometry.type).toEqual(output[0].geometry.type);	GeometryCollection	Polygon	fail

7.2.1-11:				
-----------	--	--	--	--

Test Case Name	Should return place lat long data successfully			
Class Name	LocationService			
Case	Input	Expected	Actual	Decision
1	expect(data.success).toBe(true);	true	true	pass
2	expect(data.message[0].lat.length).not.toBe(null); expect(data.message[0].lon.length).not.toBe(null);	59.4250354, 24.7438692	59.4250354, 24.7438692	pass

7.2.1-12:				
Test Case Name	Should return place name successfully			
Class Name	LocationService			
Case	Input	Expected	Actual	Decision
1	expect(data.success).toBe(true);	true	true	pass
2	expect(data.message[0].display_name).toBe("67a, Pärnu mnt, Kesklinn, Tallinn, Kesklinna linnaosa, Tallinn, Harju maakond, 10131, Estonia");	"67a, Pärnu mnt, Kesklinn, Tallinn, Kesklinna linnaosa, Tallinn, Harju maakond, 10131, Estonia");	"67a, Pärnu mnt, Kesklinn, Tallinn, Kesklinna linnaosa, Tallinn, Harju maakond, 10131, Estonia");	pass

7.3 Non-functional requirements test

Most of the non-functional requirements tested by manually and works fine. Few of tested by different tools are described below.

NFR 03: Website performance test results using several testing tools are shown on Figure 18, Figure 19 and Figure 20.

Gtmetrix⁴⁰ :

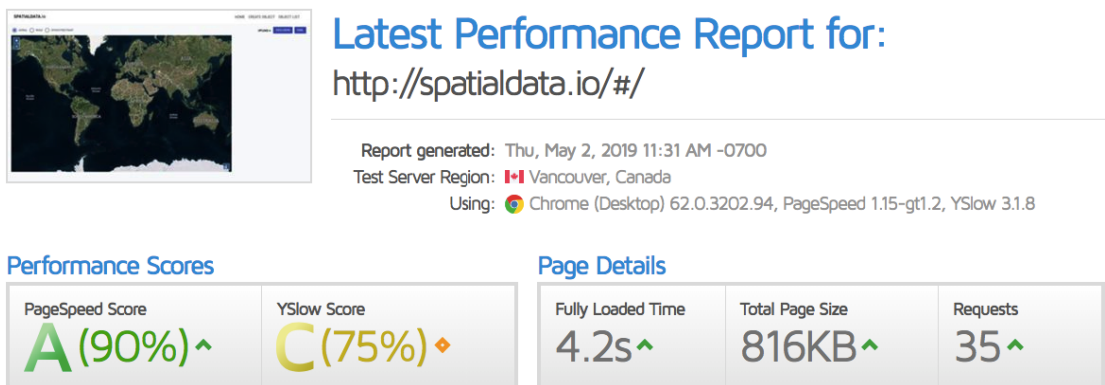


Figure 18. Performance test result by Gtmetrix

Pingdom⁴¹:

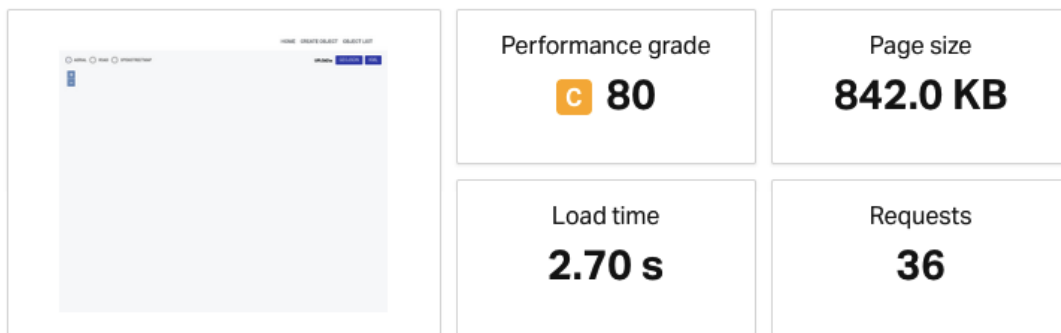


Figure 19. Performance test result by Pingdom

Google Page Speed⁴²:

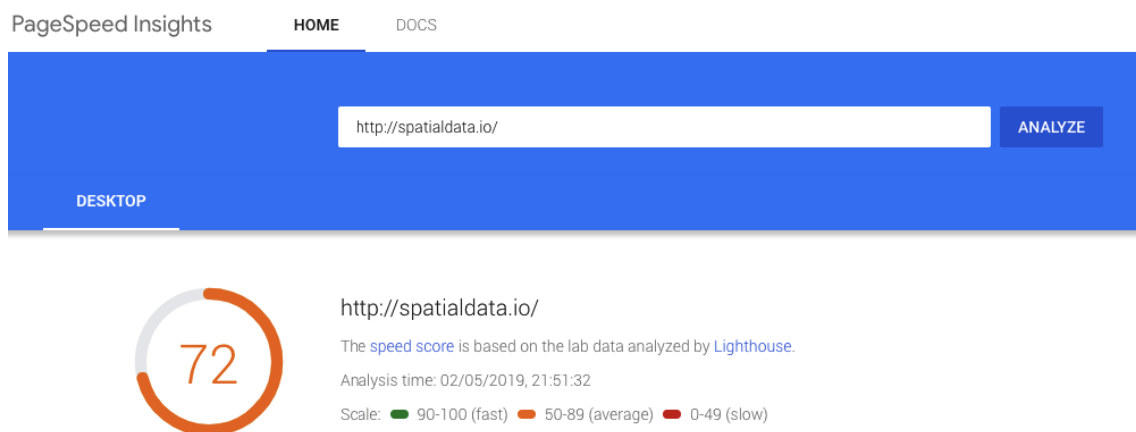


Figure 20. Performance test result by Google page speed

⁴⁰ <https://gtmetrix.com/>

⁴¹ <https://tools.pingdom.com/>

⁴² <https://tools.pingdom.comdevelopers.google.com/speed/pagespeed/insights/>

7.4 Tool Evaluation by User

To evaluate the spatialdata.io tool, I conducted a meeting with the relevant team members of the project of SILLE. Participants had more than 2 years of experience in the field of GIS application development and analysis.

First, the tool was presented to the team members. During the presentation, we had an exploration session where participants could get used to with the application. After the presentation, I asked the team members a few questions following the non-functional requirements.

After that, we identified the correctness of visualized data and customized data, after performing the following random test.

Random test 1: User took 3 random points from the uploaded dataset for merging in an area.



Figure 21. Merging points in an area using convex hull functionality

Random test 2: User took a line from dataset to extend rail line size using buffer add functionality.

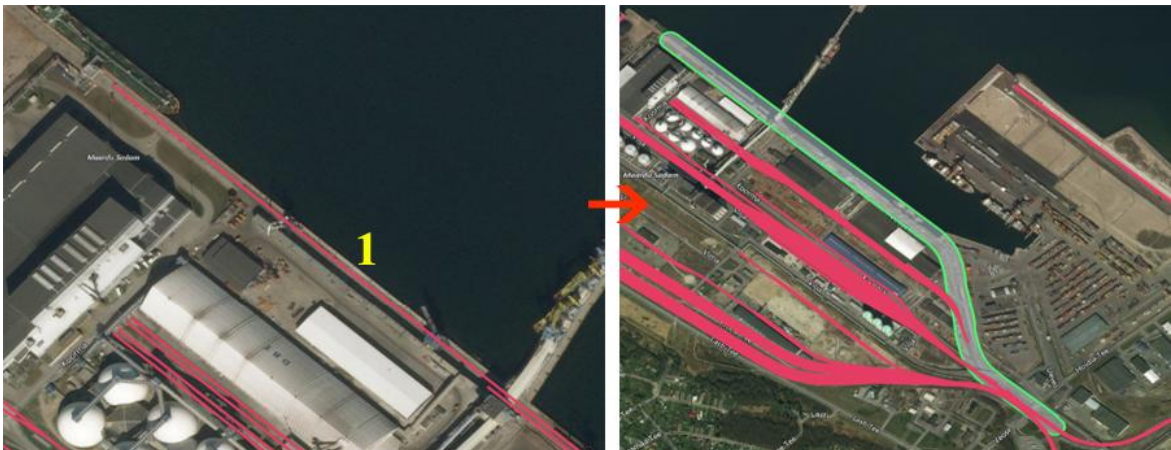


Figure 22. Adding buffer to the rail line

Random test 3: User took 2 buildings' area from dataset to merge it in an area using bounding box functionality.



Figure 23. Merge two buildings' area into a single area

Figure 21, Figure 22 and Figure 23 are representing the random test results. Overall feedback from team members are given below in Table 6.

Table 6. Shows the feedback from SILLE team

Requirement	Overall Result	Feedback
NFR01: User-Friendly	Ok	It would be better if UI acts as a well-defined workflow that guides the user through the application
NFR02: Accessibility	Satisfied	
NFR03: Performance	Satisfied	
NFR04: Cross browser compatibility	Satisfied	
NFR05: Access permission	Satisfied	It would be better if there is a registration functionality so that the user can save their own data in their profile for further modification.
NFR06: Extensibility	Satisfied	The extensibility of this tool was good enough - we had a discussion on how we can integrate it with SILLE application.
NFR07: Compatibility	Satisfied	

NFR08: Operability	Satisfied	
NFR09: Error Handling	Ok	

Additionally, we had a discussion on how we can integrate with SILLE application for collection of the area of interest information so that it be useful for the GIS application user, as well as for a service provider.

The current workflow in this application for a user to order service showing on Figure 25.

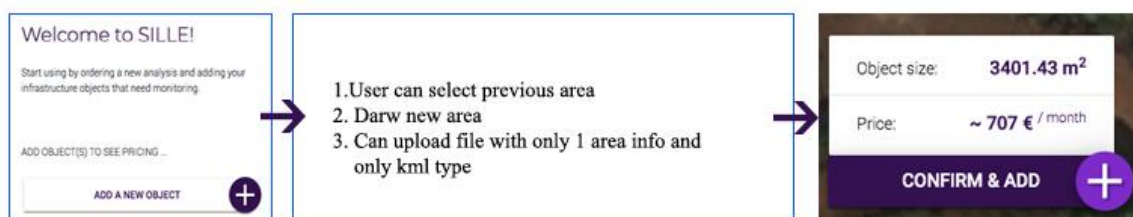


Figure 24. Workflow for ordering service through SILLE

As we can see from Figure 24, now a user is only getting few functionalities to confirm their area of interest.

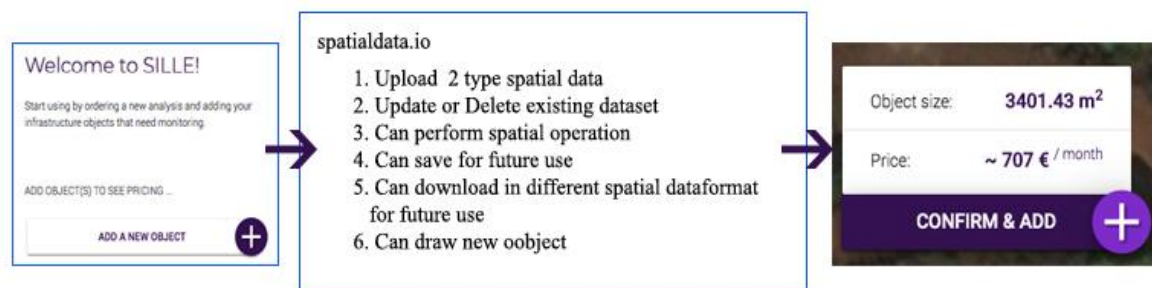


Figure 25. Workflow for ordering service through SILLE after adding spatialdata.io tool

If we allow a user to use this tool before confirming their order then the user will get more functionalities (Figure 25) which will help them use existing dataset rather than drawing hundreds of bridges, houses, lines, etc.

Additionally, we came up with a discussion that automatically merging the nearest location, adding more spatial data format for upload to customize and download, better workflow for easy use could be the works for further development of this tool.

8 Summary

The scope of the thesis was to develop a free and open source, user friendly web solution for non-GIS specialist end users so that they can customize spatial data graphically with the help of performing different spatial operations in the background.

The final solution enables users to import spatial data in geojson or kml spatial data format for customizing and it is also offering different geospatial operation to customize imported data for their desired AOI. Additionally, it allows users to check the visual representation of customized data and provides different data sharing options.

The implemented web solution was tested using different data sets from multiple sources. Few datasets were provided by AS Datel⁴³ and others are from worldwide web in geojson and kml format. Functionalities of this application is evaluated by acceptance and automated test. Additionally, web application performance is also tested by different free tools like google page speed, gt-matrix, etc.

Future work could be adding more data format support for customizing and downloading options. The implementation of more spatial functionalities in this solution will make this application more useful to a user.

⁴³ <https://www.datel.eu/en/>

9 References

- [1] "Geographic information system," [Online]. Available: https://en.wikipedia.org/wiki/Geographic_information_system. [Accessed 12 1 2019].
- [2] Hélder Lopes , Paula Remoaldo, Vitor Ribeiro, Sara Silva, "Geographic Information System - the past, present and future", Category: Geographic Information Systems , Book: Encyclopedia of Information Science and Technology, Fourth Edition.
- [3] Feng Wang , Yunfei Shi , Xuguang Qin, Huan Zhang, "*Spatial Data Sharing And Interoperability Based On Web Spatial Data Service And Gml*". ISPRS Workshop on Updating Geo-spatial Databases with Imagery & The 5th ISPRS Workshop on DMGIS
- [4] Butler, Howard and Daly, Martin and Doyle, Allan and Gillies, Sean and Schaub, Tim and Schmidt, Christopher , "GeoJSON ", [Online]. Available: <https://geojson.org/>[Accessed 12 08 2019].
- [5] "OpenLayer", [Online]. Available: <https://openlayers.org/>, [Accessed 12 08 2019].
- [6] "WEb Mercator projection", [Online]. Available:https://en.wikipedia.org/wiki/Web_Mercator_projection, [Accessed 12 08 2019].
- [7] Robert Laurini, Derek Thompson, "Fundamentals of Spatial Information Systems", 1992, [Online]. Available: <https://www.sciencedirect.com/topics/computer-science/spatial-operation>, [Accessed 12 08 2019].
- [8] "Buffering In Geographic Information System" by Chhabi Baral, [Online]. Available: [https://www.academia.edu/11832680/What do you mean by buffer in GIS](https://www.academia.edu/11832680/What_do_you_mean_by_buffer_in_GIS), [Accessed 12 08 2019].
- [9] Essentials of Geographic Information Systems v. 1.0 , Publisher: Saylor Academy , Year Published: 2012, [Online]. Available: https://saylordotorg.github.io/text_essentials-of-geographic-information-systems/index.html, [Accessed 12 08 2019].
- [10] Alberto Belussi, Barbara Catania, Eliseo Clementini, Elena Ferrari, "Spatial Data on the Web: Issues and Challenges", book: Spatial data on the web: Modeling and management (pp.1-12)
- [11] Representational State Transfer (REST), [Online]. Available: https://en.wikipedia.org/wiki/Representational_state_transfer, [Accessed 12 08 2019].
- [12] "Turf.js, Advanced geospatial analysis for browsers", [Online]. Available: <https://github.com/Turfjs/turf>, [Accessed 12 08 2019].
- [13] "Angular", [Online]. Available: <https://angular.io/>, [Accessed 12 08 2019].
- [14] Dunlu PENG, Lidong CAO, Wenjie XU, "Using JSON for Data Exchanging in Web Service Applications ", Journal of Computational Information Systems 7: 16 (2011) 5883-5890
- [15] "ArcGIS", [Online]. Available: <https://www.arcgis.com/index.html>, [Accessed 12 08 2019].
- [16] "QGIS:a free and open-source cross-platform desktop geographic information system", [Online]. Available: <https://qgis.org/en/site/>, [Accessed 12 08 2019].
- [17] "PostGIS", [Online]. Available: <https://postgis.net/>, [Accessed 12 08 2019].
- [18] "PostgreSQL: The World's Most Advanced Open Source Relational Database", [Online]. Available: <https://www.postgresql.org/>, [Accessed 12 08 2019].
- [19] "SpatiaLite", [Online]. Available:<https://www.gaia-gis.it/fossil/libspatialite/index>, [Accessed 12 08 2019].

- [20] “Leaflet”, [Online]. Available: <https://leafletjs.com/>, [Accessed 12 08 2019].
- [21] “Open Geospatial Consortium (OGC)”, [Online]. Available: <http://www.opengeospatial.org/>, [Accessed 12 08 2019].
- [22] A.M.Andrew, “*Another efficient algorithm for convex hulls in two dimensions*” , [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0020019079900723> , [Accessed 12 08 2019].
- [23] “Bounding Box”, [Online]. Available: https://wiki.openstreetmap.org/wiki/Bounding_Box, [Accessed 12 08 2019].
- [24] “Bootstrap: Front-end framework”, [Online]. Available: <https://getbootstrap.com/>, [Accessed 12 08 2019].
- [25] “Node.js”, [Online]. Available: <https://nodejs.org/>, [Accessed 12 08 2019].
- [26] “What is node js”, [Online]. Available: <https://www.quora.com/What-is-Node-js-1> , [Accessed 12 08 2019].
- [27] “HTTP request methods” , [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>, [Accessed 12 08 2019].
- [28] “Angular: Best Use Cases and Reasons To Opt For This Tool”, [Online]. Available: <https://yalantis.com/blog/when-to-use-angular/>, [Accessed 12 08 2019].
- [29] Nina Fat, Marijana Vujovic, Istvan Papp, Sebastian Novak , “Comparison of AngularJS framework testing tools”, 2016 Zooming Innovation in Consumer Electronics International Conference (ZINC), DOI: 10.1109/ZINC.2016.7513659
- [30] “Express.js: A Server-Side JavaScript Framework” , [Online]. Available: <https://www.upwork.com/hiring/development/express-js-a-server-side-javascript-framework/>, [Accessed 12 08 2019].

Appendix

I. List of abbreviations and terms

GIScience	Geographic Information Science
GIS	Geographic Information System
AOI	Area of Interest
KML	Keyhole Markup Language
JSON	JavaScript Object Notation
SHP	Shapefile shape format
API	Application Programming Interface
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
PC	Personal Computer
HTML	Hyper Text Markup Language
SDK	Software Development Kit
JS	JavaScript
lat	Latitude
long	Longitude
URL	Uniform Resource Locator

II. License

Non-exclusive licence to reproduce thesis and make thesis public

I, Partha Sarathi Kar,

(author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

SpatialData IO: A web-based spatial data customization tool for collection of area of interest information,

(title of thesis)

supervised by Juhan-Peep Ernits, Ph.D., Dietmar Pfahl, Ph.D and Andreas Kiik MSc.

(supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Partha Sarathi Kar

13/08/2019

