POLITECNICO MILANO 1863
SCUOLA DI INGEGNERIA CIVILE, AMBIENTALE E TERRITORIALE

Development of a voice virtual assistant for the geospatial data visualization application on the web

Tesi di Laurea Magistrale in GEOINFORMATICS ENGINEERING

Author: **Homeyra Mahmoudi**

Student ID: 10718622
Advisor: Professor Maria Antonia Brovelli
Co-advisors: Professor Silvana Camboim
Academic Year: 2022-2023

Abstract

Embedding voice assistant technology in web-based platforms for geospatial data visualization and processing is necessary due to the democratization of data and the need for more accessible and interactive tools. Voice assistants offer a natural and intuitive way to interact with data, especially for users who struggle with traditional methods. However, virtual assistants' effectiveness is still limited in certain domains, including geospatial data visualization and processing, which requires specialized technology to understand and process user commands' semantics and context. This study explores the potential of embedding a voice virtual assistant within a web application platform which called BStreams for geospatial data visualization and processing. Based on the searches that has been done, this criterion still lacks structured prototypes and efforts. The study addresses the challenges of integrating virtual assistants into the Geographic Information System (GIS) filed, including understanding semantic context, and the cognitive load of users. For initializing the process of creation of a structured prototype, a methodology has been presented for developing a discourse framework specialized for geospatial data. Since the final goal of this application is to enabling users to interact with geospatial data visualization with voice commands, a survey has been done. The survey accesses the most frequent terminologies that is been used by English Language, for interacting with geospatial data to achieve special tasks. The application was developed using open-source frameworks, including the Web Speech API, leaflet, Mapbox geocoding, and JavaScript. The study concludes that a voice virtual assistant can enhance human-computer interaction for geospatial data visualization and processing and recommends further research in this area.

Keywords: Voice user interface; Geographic Information System; human-computer interaction; multimodal interface; natural language; Web application

Abstract in lingua italiana

L'integrazione della tecnologia degli assistenti vocali nelle piattaforme web per la visualizzazione e l'elaborazione dei dati geospaziali va nella direzione di democratizzazione dei dati e della disponibilità di strumenti più accessibili e interattivi. Gli assistenti vocali offrono un modo naturale e intuitivo di interagire con i dati, soprattutto per gli utenti che hanno difficoltà con i metodi tradizionali. Tuttavia, l'efficacia degli assistenti virtuali è ancora limitata in alcuni settori, tra cui la visualizzazione e l'elaborazione dei dati geospaziali, che richiedono una tecnologia specializzata per comprendere ed elaborare la semantica e il contesto dei comandi dell'utente. Questo studio esplora il potenziale dell'incorporazione di un assistente virtuale vocale all'interno di una piattaforma applicativa web per la visualizzazione e l'elaborazione di dati geospaziali. In base alle ricerche effettuate, questo criterio manca ancora di prototipi e sforzi strutturati. Lo studio affronta le sfide dell'integrazione degli assistenti virtuali nei sistemi informativi geografici (GIS), tra cui la comprensione del contesto semantico e il carico cognitivo degli utenti. Per avviare il processo di creazione di un prototipo strutturato, è stata presentata una metodologia per lo sviluppo di un framework discorsivo specializzato per i dati geospaziali. Poiché l'obiettivo finale di questa applicazione è quello di consentire agli utenti di interagire con la visualizzazione di dati geospaziali con comandi vocali, un'indagine è stata condotta per capire questi aspetti. L'indagine ha permesso di individuare le terminologie più frequenti utilizzate in lingua inglese per interagire con i dati geospaziali e svolgere compiti particolari. L'applicazione è stata sviluppata utilizzando framework open-source, tra cui Web Speech API, leaflet, Mapbox geocoding e JavaScript. Lo studio conclude che un assistente virtuale vocale può migliorare l'interazione uomo-computer per la visualizzazione e l'elaborazione dei dati geospaziali e propone ulteriori ricerche in questo settore.

Parole chiave: Interfaccia utente vocale; sistema informativo geografico; interazione uomo-macchina; interfaccia multimodale; linguaggio naturale; applicazione web

Contents

Introduction

Voice user interfaces facilitate communication between humans and devices by leveraging speech recognition technology to interpret spoken commands and execute associated functions. However, most of these interfaces have not yet been fully integrated into Geographic Information Systems[13]. This research is dedicated to present a complete prototype for embedding the voice virtual assistants on a web application which enable users to visualize and process geospatial data. To achieve this goal some challenges have been recognized. These challenges include recognizing a variety of voice commands and languages, processing and displaying geospatial data in real-time, accommodating the cognitive faculties and restrictions of users, considering data democracy and domain knowledge, and leveraging artificial intelligence technology while being aware of ethical and social implications. The lack of existing applications in this field and the absence of specialized infrastructure for speech recognition in geospatial data visualization pose further challenges.
This study aims to explore the feasibility of voice commands in an open-source web application and bridge the gap between user behavior and the use of voice recognition technology in geospatial data visualization. The study seeks to understand the cognitive aspects of user interactions with such applications. In order to evaluates all the available aspects regarding this matter, the following goals for this thesis has been established:

1. To embed a voice assistant into an application specialized for geospatial data visualization and processing, it is necessary to develop a methodology for conceptualizing the discourse between the user and the system. *This study utilized a methodology called PlanGraph[55]*, which allows for the visualization of all tasks and actions in a visible approach for both the user and the system.

2. It is important to investigate how verbal commands can enhance the user experience. To achieve this, a survey was conducted to evaluate users' vocabulary and simulate interactions with maps. The survey included questions on user characteristics to identify potential patterns. The survey results led to improved map control design and enhanced user experience. Understanding language use and communication strategies can help establish more effective communication and better interactive system design in our application.

3. To analyze the data derived from the survey, a large corpus of voice commands was collected and trained using ChatGPT 4.0 [50], an advanced artificial intelligence language model. The survey results were compared to the ChatGPT output to examine the similarities and differences between the natural language of real humans and artificial intelligence. This comparison was conducted in the context of evaluating the usage of natural language for dealing with geospatial data visualization and processing.

4. The development of the application was the final step. The main cores of this application were designed using the most used programming languages and frameworks for geospatial data, *Web Speech API[45]*, and BStreams platform, which hosts the application in an online and web-based source. Challenges were encountered during the implementation of the application, as it was considered one of the pioneers in the industry of geospatial data applications.

1 | Voice recognition in online mapping

The embedded new technologies such as voice recognition and Natural Language Processing (NLP) in the tools that citizens are using every day, become a novel feature of the development process. Although implementation of this feature can raise the user experience and accessibility, the approaches for usage, design, and applications are altering for different fields. Cartography and Geoinformatics have become increasingly relevant in the emergence of new technologies such as map and geospatial data visualization. *The ability to store, manipulate, and analyze large amounts of data has changed the way we think about and use maps, and has led to the development of new cartographic techniques and tools that are revolutionizing the field of geography[50].* Advances in technology have enabled the development of new mapping and geospatial technologies. Online mapping applications considered as one of them, which commonly provides powerful ways to analyze and visualize geospatial data. *The availability of online mapping applications and open geospatial data has democratized the production and consumption of spatial information, enabling individuals and communities to participate in the creation of geospatial knowledge[28].*

Speech recognition technology involves converting human speech into a form that can be understood and processed by a computer. This is accomplished through a combination of digital signal processing, pattern recognition, and machine learning techniques[32]. In recent years, speech has increasingly become a more ubiquitous means of interacting with automated systems. This is due to the growing availability of speech recognition technology that enables users to issue voice commands to perform a variety of tasks, from controlling home automation devices to accessing digital assistants. *In addition to long established telephony based or interactive voice response (IVR) interfaces, voice enabled intelligent personal assistants (IPAs) like Amazon Alexa, Apple Siri, Google Assistant, and Microsoft Cortana are widely available on a number of devices [17].* Using speech recognition technology in an online mapping application, increase the efficiency of the application, user experience, accessibility, and remove the boundaries of required skills and domain knowledge for dealing with geospatial data. *The integration of speech recognition technology with map applications can enable users to interact with map data and access information in a more natural and intuitive way, without the need for complex user interfaces or manual input.[10]* The traditional bound of accessories such as mouse and keyboard for data visualization and analysis can be removed with embedding these types of technologies. Moreover, Various criteria such as Education, and Emergency system can be benefited too.

1.1. Current situation

Several research studies have explored the potential of using speech recognition and natural language processing (NLP) in geospatial applications. *Pei-Chun Lai and Auriol Degbelo presented a prototype for retrieving metadata using both text and speech inputs in a web map application. In their experiment, they found that typing and speech had similar input duration times but differed in user experience properties[35]. Meanwhile, Thomas Gilbert developed VocalGeo, a software that uses speech recognition and web mapping to provide real-time geospatial context on a map, which enables interactive teaching of geospatial information using accessible web technologies[26].*
Another study by Davide Calì , Antonio Condorelli's research was aiming to make comparative analysis between the traditional Geographic Information System (GIS) and new ones which new technologies such as NLP and speech recognition has been embedded on them. The novel technology that was under investigation called iTour, which is a project for intelligent multi-model mobility services[16]. The embedded speech recognition and NLP on the system proved, the reduction of user's knowledge requirement to operating the system, also facilitating the understanding of user's intentions based on human language.
Furthermore, *Hongmei Wang, Guoray Cai, and Alan M. MacEachren introduced a computational model, PlanGraph, to track the SharedPlan in human-computer communication and a prototype software agent, GeoDialogue, for natural interaction between users and conversational GIS. Limitations of GeoDialogue include difficulties in interpreting recognized input and unsuccessful attempts at focus actions. Future research aims to improve interaction and update knowledge dynamically during human-GIS communication[55].*
While the previous work was the first theoretical framework for embedding the voice technology for geospatial data, the project by Jacobson and Sam,K created a multimodal web-GIS application that combines voice and graphic interfaces[30]. It shows the potential of using voice commands to navigate and visualize spatial data. The application offers benefits for visually impaired users and those working in the field. The project demon-

strated the feasibility of using existing voice-enabling technologies to create sophisticated multimodal interfaces for GIS practitioners.
The status of the development of the speech recognition in applications that are dealing with geospatial data visualization, process, and analysis has dated to few decades ago. *Andrew Hunter and Dr. C. Vincent Tao did the study which investigated whether using a mobile mapping tool with speech recognition capabilities could be an effective method for identifying and locating issues with roads, sidewalks, and curbs[29]*

1.2. The main gaps

*As the Blanco, discussed the lack of the infrastructures and practical experiment towards implementing the voice speech recognition on the GIS interface, this study also emphasizes on this aspect to reveal the challenges and obstacles for a voice speech geospatial data visualization[13]*Moreover, it is noteworthy, that the studies focused mainly on the interactions which have navigation characteristic. Therefore, the necessity of evaluation of Natural language for dealing with geospatial data visualization and customization considered as one of the main pillars of this research.
In the rapidly evolving technological landscape, the development of intelligent products requires not only the incorporation of advanced technologies, services, and communication protocols but also the exploration of novel paradigms that describe the interactions and relationships between humans and technology. These paradigms should reflect a deeper understanding of the cognitive and behavioral aspects of human-computer interaction and should consider the implications of emerging technologies on user experience, user engagement, and user satisfaction. The current research in the field of voice-enabled geospatial data visualization applications is limited and outdated in terms of incorporating new technologies and application requirements. The present study aims to explore the feasibility of voice commands in an open-source web application specialized for data and geospatial data visualization, and for public use, while also examining the cognitive aspects of users' interactions with such applications. The study seeks to bridge the gap between the cognitive aspects of user behavior and the implementation of voice speech recognition technology in geospatial data visualization applications.

1.3. Problems and questions

Developing a voice speech recognition map application on the web is a complex process that requires addressing several technical and conceptual challenges. It involves creating a user-friendly interface that can recognize a wide range of voice commands, analyze, and display geospatial data in real-time, and ensure that user data is secure and private. In this context, some of the main problems and questions that developers may encounter include:

1. Variety of vocabularies for voice commands and languages: we can address the challenge of recognizing a wide range of words, phrases, and pronunciations from different accents, dialects, and languages. The language itself plays a crucial rule in comprehending the relationship between the geospatial data visualization and the user, since the deep nature of the language depends to the cultural, social, historical, and other variables. The questions that can address these issues are firstly "How can the application be trained to recognize the different accents, dialects and languages?", and second "How the most compatible and meaningful vocabularies and lexical choices regarded the geospatial data visualization can be collected?".

2. Digital earth: Developing a voice speech recognition map application on the web requires accessing and processing large amounts of geospatial data, including maps, satellite images, and other information. The necessity of processing and displaying geospatial data in the real-time and also removing the barriers of the domain knowledge for realizing the digital earth's goals need to be taken into consideration.

3. Cognitive aspects of human-computer interaction: To effectuate the development of a voice speech recognition map application on the web that is amenable to a heterogeneous user base, comprising individuals who may be unaccustomed to voice recognition technology, it is imperative to devise a user interface that is both intuitive and accommodates the cognitive faculties and restrictions of the intended user population. The impediments in this realm encompass the mitigation of errors and misconstructions, progression of the user's inclinations, and the assimilation of the application's design in alignment with the idiosyncratic requirements of the tasks at hand.

4. Data democracy and domain knowledge: The concept of data democracy and domain knowledge are accommodating different users with wide variety of the knowledge level regarding the geospatial data visualization in a plain perspective. It enables the users regardless of their technical expertise or familiarity with geospatial data to gain and deal extra knowledge. Embedding the Voice speech recognition requires to confront these matters: considering the heterogeneity of the users in terms of needs and preferences and concisely balancing the user's experience in these matters.

5. Artificial intelligence technology and lack of the application in this field: Developing

a voice speech recognition map application on the web requires leveraging artificial intelligence technology to enhance accuracy and reliability. However, the lack of existing applications in this field poses a challenge, and developers must consider ethical and social implications. Training and developing such applications require a basic knowledge regarding the preferred lexical choices and as a matter of a fact, a Tag cloud. There are no such a practices or basic infrastructures in this area which specialized for the usage of speech recognition in field of geospatial data visualization.

2 | State of the art

Technology have made people's life easier, and the progress is breaking through every field. Humanity has the challenge of automation, and in daily basis there are majority of tasks that has been eased by this aspect. Automation and artificial intelligence (AI) can be found as the basic infrastructures of tools or events that humans deal. The instinct nature of artificial intelligence (AI) is simulating the functionality of a human brain, and most of the studies and technologies in this industry are following the same path. Communication is the inseparable skill of the humanity which in late 1950 turned to a domain in the research area of artificial intelligence (AI). Natural language processing (NLP), introduced as the combination of the communication, computer science and AI. This science aims to facilitate the path of understanding and interpreting of natural language for verity of applications. In the past decade the Natural Language Processing has been embedded in our daily life: The automatic translation in the social media, text classification keeps our email inboxes from collapsing under a deluge of spam, the search engine which are not depended to string matching, and other numerous examples. *According to E.D. Liddy, NLP is a set of computationally based techniques that are theoretically driven and used to analyze and represent texts that occur naturally[38].* This is done at one or multiple levels of linguistic analysis, with the ultimate goal of achieving language processing capabilities that are similar to those of humans, for various tasks and applications.

Over the last seventy years the research and developments are classified into five main areas, which one of them is Speech or Voice recognition. Speech recognition is one of the subfields of Natural Language Processing (NLP). *Speech recognition has held a long-standing allure as an alluring and captivating means of providing input to computer systems[54].* It digitizes the received voice's signal with the help of the Hidden Markov Model, decode it to the possible phonemes, and finally with a special algorithm finds the most likely word. Advent of the technology, internet, and World Wide Web influenced the increase demands on software and applications that potentially can recognize and process all kind of text. Beyond of only recognition texts, usage of the voice channel turned to another path to pilot them. *Usage of the speech or voice recognition in the interaction between the users and the interfaces, will increase the efficiency, and mobility[33].*

The geospatial data visualization, interpretation and customization has been always a challenge, in many perspectives. Regardless of the user's knowledge, the interaction among the interface and the user are mainly some default and limited orders. Moreover, the interfaces that are open sources for everyone, might not take into consideration accessibility aspects for disabled people, or complexity of data in terms of multiple layers harder the process of customization and comprehension. Accessing to geospatial data, in real time and providing tool for processing it, in a multi device environment can be eased and improved by using multimodal interface.

In the field of the geospatial data which mainly deals with two known formats; raster and vector, the lack of the multimodal interfaces and Human-Computer Interactions (HCI) for visualization and processing, has been a challenge for experts and naïve users. The gap mainly was because of the sophistication of demonstration, level of the details, volume of the data, additional dimensions and the utilization of outdated visualization methods. However, efforts have been done in the field of designing a user-friendly interface to improve the navigation and interaction, but the geospatial data platforms available in mass-market are not fully multimodal interfaces. The design and usability of the interfaces which deals with geospatial data has traditionally been tailored with mouse and keyboard in an office environment. *Empirical research has evinced that the incorporation of multimodal interfaces can yield a substantial reduction in the complexity of interfaces utilized in the domain of Geographic Information Systems (GIS)[25], [19].* A voice input can be considered as one of the features that overcome the mentioned obstacles. Furthermore, it can allow users to control their interaction and selection, while they are processing the data. In order to achieve personal tasks more quickly and efficiently, it is possible to personalize spatial map content to tailor maps containing specific feature information with help of voice command. *There are still platforms that support the implementation of voice recognition, such as ESRI ArcQuickCapture, which utilizes the built-in voice assistant of iOS, Siri[30].*

Blanco, T and colleagues studied and presented the initial efforts to create specialized technology for voice user interfaces in Spanish language GIS. Three experiments were conducted to develop a minimum corpus of Spanish terms, validate the findings, and identify user feedback importance, user profiles, and use cases[13]. They present a fundamental deficit in the field of geospatial data, while their methodology also covered some critical points regarding the implementation of voice speech recognition technology in the web map application. *One crucial component of digital assistants is the database of words, vocabularies, phrases, and lexical choices, which is essential for their proper functioning. However, from a scientific standpoint, an important challenge for these assistants is to establish effective interaction with users, as highlighted by Loopventures' assessment in[53].*

This assessment underscores the gap between the success rate in understanding user input and executing appropriate responses, indicating the need to bridge the effectiveness of grammar and the establishment of prompts. Notably, the lack of a corpus of terms that would enable the usual functions of a GIS to be commanded has impeded even the basic level of success in understanding natural language input for GISs.

2.1. Literature

Numerus research and experiments for facilitating geospatial data platforms with voice command has been done in the past decades. The efforts mainly focused on embedding the voice command control on Geographic Information System platform (web and mobile), metadata and feature acquisition by voice command, and improvement of the geospatial query with NLP technologies. Presented here is a brief overview of three categories of related studies and initiatives.

2.1.1. Embedding voice control on GIS platforms

To provide users with access to complex geospatial databases, Lokuge developed a reactive interface display that integrates spoken queries, dialog management, and domain knowledge. As a result, Lokuge 's database retrieval system utilizes only speech as an interaction modality, thus remaining unimodal[39]. The results of research conducted by Oviatt et al indicate that speech recognition, when used in conjunction with dynamic maps, can be error-prone and inefficient[48].
An interface based on pen and speech was developed by Cohen et al, allowing dynamically generated maps to be requested[20]. Evaluation of the pen-based interaction reveals that the users could express the spatial relations more efficiently.
Dan Jacobson described the design and architecture of multimodal web-GIS application with SALT[30]. The application has predefined voice interactions that including the navigation, location query and general interactions.

2.1.2. Metadata generation and Feature retrieval by voice

An application for generation of the metadata for web map search has been designed by Pai-Chun Lei[34]. The prototype mainly is based on the creation of metadata, features for annotation, API for speech recognition, and conversion of the metadata. The speech recognition infrastructure that has been used, is Mozilla web API. The platform specialized for generation of metadata for map, but it also provides the possibility of the

traditional interactions for realizing the interactions.
Another study that done by Chuanrong Zhang represent a prototype for automatic search of the geospatial feature in the case of the emergency by using NLP[58]. Their team used the Web Feature Service (WFS) to improve the search aspects such as time, SPARQL to be able to query data, and a customized NLP adoptable to the architecture.

2.1.3. Improvement of the geospatial query with NLP technologies

I-Tour is a project form European Union which aims to provide a multi-model intelligent framework for the mobility service. *The study by Davide Cali[16]* describes the comparison between a traditional GIS and NLP interface for processing the geospatial data on I-Tour. However, their result reveals the fact that the usage of NLP for a multi-model interface is quite beneficial for intelligent agent GIS users, but also ease the interaction process for the naïve users.
Qwhery,[5] is a cloud that joins open data and location-based services, events, projects and insights that is ongoing in the city Burlington, Ontario-Canada to a smart voice assistant called Q11. The program is designed to help the startups that are leveraging Geographic Information System (GIS) provided by ESRI. The program is now available for Alexa (Amazon's cloud voice assistant), and it can provide information for the users regarding the calendar of the events, retrieve information about the local offices, and provide answers for city services.
The integration of Google Assistant with maps and navigation is currently the most advanced, enabling users to search for specific places nearby and manage navigation using voice commands. The system executes these functions through associated apps like Google Maps. However, the interaction with the Assistant still falls short of emulating a truly natural interaction, unlike traditional software interaction models, and is limited to a few specific functions (such as the quoted questions mentioned above). *In contrast, the ongoing integration efforts in vehicles for hands-free operation of mobile devices are commendable[52].*

2.2. Best practice and implementations

The current literature reports various scenarios where Spatial Data Infrastructure (SDI) is a valuable tool, with some of these scenarios being researched in the context of risk management processes. *The study indicated some existing tools for geospatial programming, which has been specialized for the virtual assistant and speech recognition. However,* *it discussed the approaches for design and implementation are wide and still there are debates and lacks for an optimal methodology[44].* In the following, a brief explanation of all the levels of the development that this study went through has been included.

2.2.1. Design

The hypothesis that was taken into consideration started from the stage to increase the accessibility and user's experience with the tools that deals with geospatial data visualization and process. One unresolved matter in the domain of GIS concerns the challenges of complying with accessibility regulations and standards, which have long been a source of difficulty in this area. Despite the fact that government services provided to the public must adhere to stringent accessibility guidelines, the same cannot be said for maps at present. *The accessibility of online maps and mapping services is a problematic issue in the context of the European Directive that regulates public sector websites and mobile applications[23].* The Directive includes an exemption for maps intended for navigational use but specifies that essential information must still be provided in an accessible digital format.

According to the Hype Cycle introduced by Gartner in 2018[49], the utilization of virtual assistants as a technology is anticipated to reach a state of stable productivity within the upcoming 2-5 years. Smart assistants have become ubiquitous in our daily lives, but they have limitations in managing certain domains. The development of geographic information systems (GISs) and web map services presents a challenging task due to the complexity of the human-computer interaction process. To address this challenge, research is needed to create a corpus that aids in developing GIS capabilities and to devise specifications for the human-computer interaction process that meet users' needs for their smart assistants.

For designing such an application, which must cover all the mentioned lacks, and evaluate the situation of voice speech recognition technology in geospatial data, the prototype of the application has been designed and implemented on an open-source platform. The platform which is called BStreams has been already facilitated with geospatial data visualization features. There are two possible formats of visualization for maps: Graduated color (Choropleth) and Marker maps. The application is available through web browser, therefore a *Web Speech API[45]* used for detection and transcription of words and voice commands. The SpeechRecognition interface enables users to recognize spoken language from an audio input using the device's default speech recognition service and respond accordingly. One of the key advantages of the SpeechRecognition interface is its accessibility. The technology is available through all the major web browsers, including Chrome, Firefox, Safari, and Edge. This means that users can easily access speech recognition capabilities on any device with an internet connection, without the need for specialized hardware or software. The methodology to categorize and studying the nature of the task's characteristics, and its bound with the natural language is PlanGraph. A PlanGraph is like a recipe graph (Rgraph) that was developed by Lochbaum in the 1990s, but it goes beyond the Rgraph by handling knowledge-preconditions in collaborative plans. *In simpler terms, the PlanGraph represents the discourse context of the conversation between the human and the computer, and it helps to ensure that the computer has the necessary knowledge to carry out the conversation effectively[15].* As the sake of increasing the usability of the application and removing the limitation of vocabularies and commands, a survey has been designed based on the all the commands that extracted from the methodology and available features on BStreams. Finally, the whole application, was developed into two phases: a demo of the voice map which mainly was for testing the applicability of the voice speech API with Geospatial data visualization, and the immigration and adaptation of the code into the BStreams platform for final deployment and publication.

2.2.2. Process

In the changing technological context, designing smart products involves understanding user needs and behaviors. This requires a critical approach to defining product features and humanizing the software. *As stated in reference[18]*, it is important to recognize the challenges that users encounter, consider them from their point of view, take a critical approach to defining product features, and imbue the software we create with a human touch.

Demo Application

Designing a demo for an application is necessary because it allows developers to showcase the functionality and features of the application in a controlled environment. Additionally, testing the application through a demo can reveal any usability issues or bugs before the application is released, which can help to improve the user experience and avoid negative reviews. *Demos serve as a valuable means for giving users a framework that helps them gain a deeper comprehension of the system they aim to create[22].* The demo of the voice map application, designed by using the same API, as mentioned before. Noteworthy, the architecture of the demo in terms of programming language and geospatial data provider has coherency with already existing architecture in the BStreams. Briefly, the application enables users to navigate to different spots, visualize two formats of maps (marker and graduated color), implement cartographical customization, and query the visualized geospatial data by voice commands. In this section the details and the used architecture discussed.

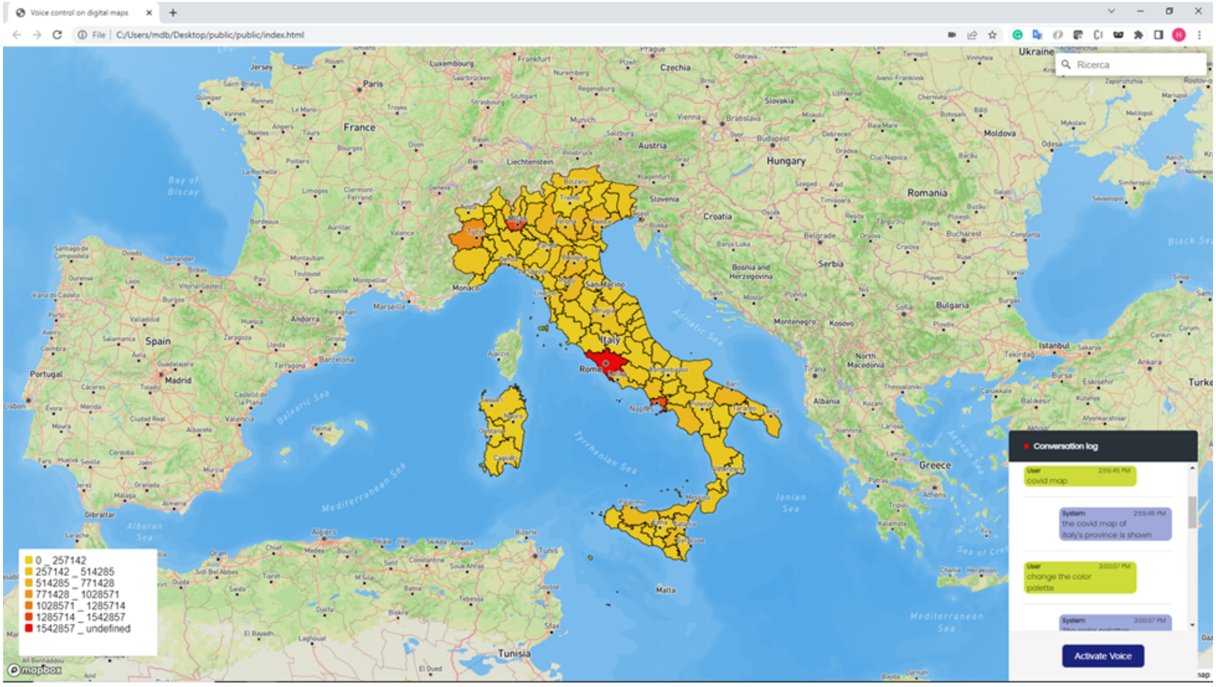

Figure 2.1: Screenshot of Demo of the voice speech map application

Lexical Choice

The traditional bound of using mouse and keyboard, while users are dealing with geospatial data visualization, and mainly with a map, considers as a familiarity concept for everyone. But if we were able to verbally command to a browser, to achieve normal tasks such as panning to left or zoom in, which words did we choose [13]? The necessity of having the correct, coherent, meaningful and practical terminologies, is undeniable. Therefore, for this reason, a survey has been designed and distributed among the general respondents, in order to evaluate and extract the lexical choice. *Lexical choice refers to the careful selection of words and vocabulary by a speaker or writer to convey a specific meaning or to evoke a particular response from the audience [43].* The survey aimed to evaluate the vocabulary of the people regarding to three main groups of tasks that developed and categorized based on the PlanGraph methodology. The tasks mainly were categorized into four groups: special domain actions, cartographical actions, spatial analysis actions, and spatial data retrieval actions. The questionnaire consisted of two main sections, and to convey the characteristic of the tasks which, the application must implement consequently after receiving the commands, visual contents in format of GIF, embedded into each question. Assessing the correlation between the user's profile characteristics and the lexical items employed, with the aim of discerning any potential patterns or associations that may shed light on the user's language use and communication strategies, has been taken into consideration in the survey with retrieving Age, Gender, Level of Education, Field of Education (not mandatory), Mother tongue, and Level of English Proficiency of the respondents. In the following section the details of the results and analysis has been discussed.

2.2.3. Materials and Technologies

The voice map developed with the latest technologies, and methodologies, to derive a deep knowledge of implementing the virtual assistant technology in the geospatial data visualization field. All the materials and technologies that has been embedded, has been originated from previous studies and research which already mentioned in the literature section.

PlanGraph

Discourse is a fundamental aspect of communication and plays a crucial role in shaping how we understand and interact with the world around us. Guoray Cai studied this matter in the field of geospatial data and GIS, and introduced the PlanGraph model for usage of the GeoDialogue. The PlanGraph concept shares similarities with Lochbaum's recipe graph (Rgraph) that was introduced in 1994 and 1998. However, PlanGraph expands upon Rgraph's approach by also taking into account the management of knowledge-preconditions in collaborative plans. The paper discussed the structure of PlanGraph by introducing three main concepts: Actions, Recipes, and Plans. Actions defined as the goal as well as the efforts that required to achieve it. Recipes is the knowledge and information of in terms of sublations, constraints, and parameters. The Plan described the schema corresponding to not only how to execute the Action, but also the mental attitudes towards it.

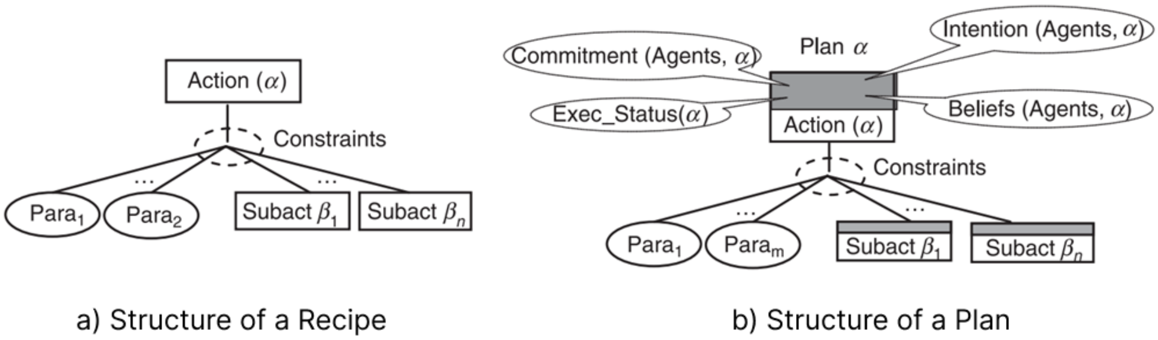

Figure 2.2: The concept of Action, Recipe, and Plan

Moreover, the characteristic of the Actions that done by the GeoDialogue are grouped into four main types:

- Type I Actions deals with spatial data retrieval.
- Type II Actions are about the analytical tasks.
- Type III Actions mainly categorized into cartographical and visualization tasks.
- Type IV Actions defined as the task that are domain specific tasks.

With respect to the defined methodology as above, the efforts done to conceptualize all the tasks which is executable by the voice map application. Initially the existing features in the BStreams' platform, derived and categorized into related format of visualization (Graduated color, Marker, Mutual for both) and the events that are involving the different parts of the architecture (Backend and Frontend) to accomplish the tasks. The nature of all the tasks has been examined and compared cautiously with all the available concepts that has been represented in the paper. With a deeper knowledge that gained through studying the backbone structure of the platforms, and also the priory knowledge, the commands have been conceptualized into the PlanGraph methodology. Considering the inputs and their formats (integer, string, and etc), the structure of applying customization and functions that are involving to accomplish a task (Backend involvement, Frontend involvement, or both), *boundaries for executing tasks such as limitation of color available color palettes or format of the marker, and industry standards for designing the user interface[51]*, that are coherent with the discourse of the domain, all the tasks restructured into PlanGraph model. The figure 2.3. Illustrated one of the examples of the tasks that has been redesigned based in the model. In the methodology section more details have been discussed.

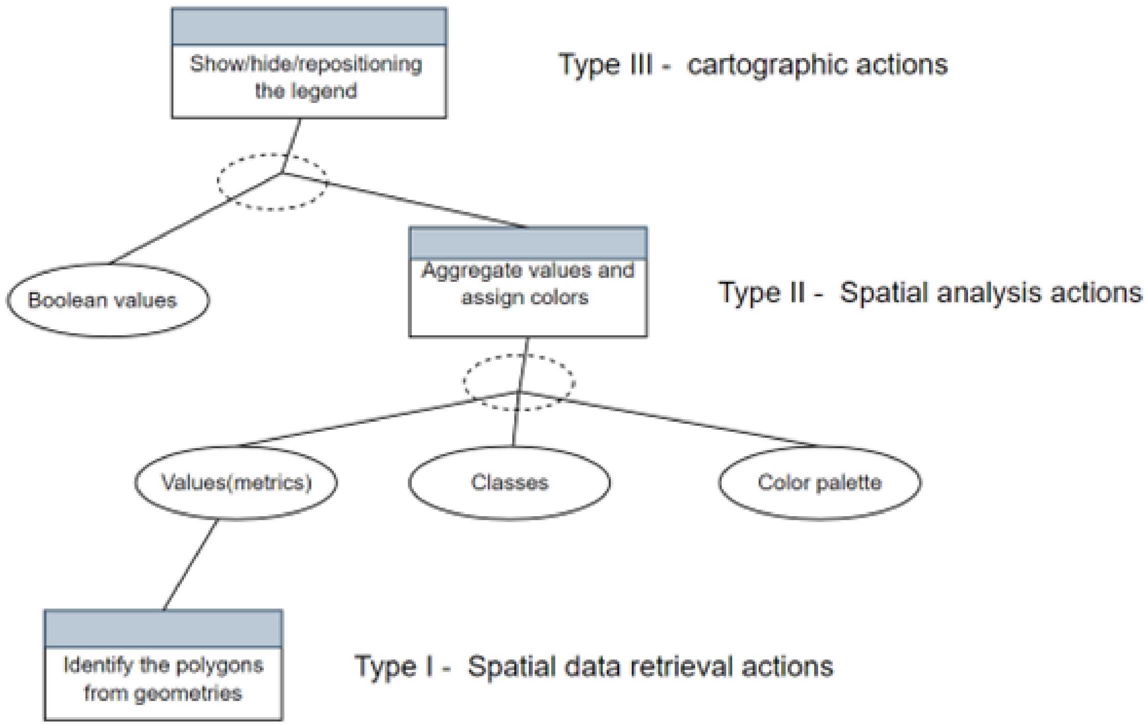

Figure 2.3: PlanGraph for customization of the legend task

Web Speech API

The Web Speech API[45] provides a way for developers to incorporate cutting-edge speech recognition and synthesis technology into their web applications. This API is based on machine learning models that have been trained on large datasets of speech data, which enables the browser to transcribe spoken words into text and read out loud the text provided by the developer with a high degree of accuracy. It uses the browser's built-in SpeechRecognition and SpeechSynthesis interfaces to process speech data, which makes it easy for developers to integrate this functionality into their web applications without requiring any additional software or hardware. Moreover, this API is cross-platform, meaning it can work on multiple devices and operating systems, which makes it a flexible solution for developers. The API is based on the same programming language of the BStreams, which is JavaScript and also is Document Object Model (DOM).

BStreams

The demand for data visualization which is combined with the Business Intelligence. The interface is designed to represent an online, web based and open-source platform for users, to create reports which is richen with interactive visualizations. The main goal of

Rank	Business insider	IEEE Survey of members (2018)	StackOverflow Developer survey (2019)	GeoGeomatics: Top languages in the GIS world	Full stack Academy
1	Jaca-Script	Python	Java-Script	Python	Java-Script
2	Java	C++	HTML/CSS	Java-Script	Swift
3	Python	Java	SQL	R	Java
4	PHP	C#	Python	SQL	C/C++
5	C++	R	Java	Java	Python
6	C#	PHP	Bash/Shell	C/C++	PHP
7	Type-script	Java-Script	C#	C#	Ruby

Table 2.1: Common programming languages in the geospatial industry

the BStreams is to ease the data visualization and helps to everyone to be able to analyze and demonstrate data. The platform focuses to convert the datasets (mostly in format of excel sheet, CSV and Google sheets), to graphs, charts and maps. The necessity of the customization has been pointed out in easy steps for the users, and the platform provided facilities for collaboration, export and share.

Frameworks

There is no universally accepted, and standard definition for architecture, but the definition that provided by *Shaw and Garlen[40]* can describes in a plain language the definition of the software architecture: "Software architecture [is a level of design that] involves the description of elements from which systems are built, interactions among those elements, patterns that guide their composition, and constraints on these patterns." One of the critical elements is to taking decision about the language framework of the application. Nenad Medvidovic and Richard *N. Taylor [42]* defined the language framework as a collection of libraries and tools that enable developers to build applications. There is verity of the factors that need to be considered before choosing an application, such as scalability, application's requirement, coherency, skills and adoptability of the data that will be processed and used by the application. *In terms of language frameworks for geospatial data, Penn State collage of Earth and Mineral science, studied and classified the most used languages[47].* The results shown in Table 2.1 imply that the architecture of the BStreams platform is coherent and adaptable with JavaScript. The main framework of this application is programmed in JavaScript.

Geocoding and Geospatial Dara provider

Granell C. and others, presented a study in 2020 which, deeply investigate the scope of virtual assistant on geographical data[27]. Implying to their research, there were eleven categories for all the applications that has been facilitated with virtual assistants in terms of Geospatial usage. From majority of the categories, the domain Positioning, where mainly used in Tourism and Recommendation application domain, caught the attention of our research. *Recommendation for user positioning studied[46]*, which demonstrate usage of virtual assistants for finding the nearby restaurants, and *another study[9]* discussed the variable of being close(geographically) for finding resources a virtual library. Consequently, the importance of positioning by virtual assistant, assumes as a growing and demandable feature.
The Voice map has been facilitated with the concept of geocoding and positioning, to enable the users navigating through desired locations and places. The *MapBox Geocoding API [41]*is a service which is provided by the Mapbox that allows users to retrieve the coordinates of the location by reverse engineering system. The retrieved coordinates can be displayed on a map since it is based on the WGS 84 coordinate reference system (CRS). This API takes in a query string that includes a location or a partial location and returns a JSON object that includes information about the matching locations. The response may include the geographic coordinates of the location, the type of location, and other relevant details such as its name, address, or administrative region.
The basic providers of geospatial data such basemaps and other features has been sources by *OpenStreetMap [8]*and *Leaflet[37]*. These two libraries are the most used technological, efficient, and reliable infrastructure which have stability in terms of implementing, usability in terms of being open-source and free, and adoptability in sense of being updated and synchronous with chosen frameworks. In the Figure 2.4 a schema of the architecture of the application shown.

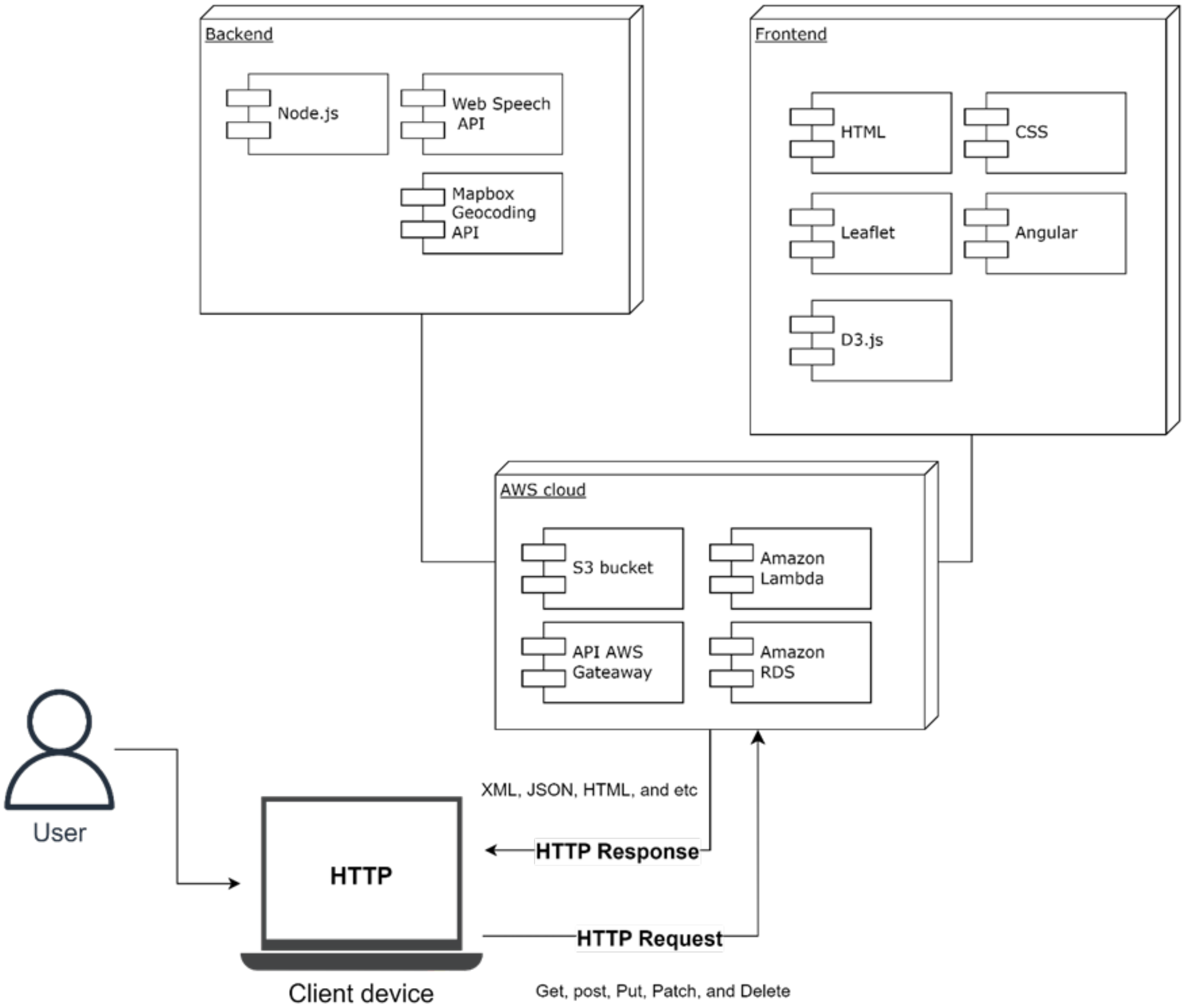

Figure 2.4: Schema of the architecture of the voice map application

3 | Methodology

In order to proceed and design of the voice map application, all the aspects from the beginning of the case study, the procedures which has to follow, the theoretical and practical frameworks, till the logic and analysis under the development and implementation process, required to be assessed and scripted. The industry for virtual assistants and voice recognition technology is thriving, and this project aimed to present a structured prototype from hypothesis till implementation of a voice map on web. The traditional paradigm of application development starts from the case study, which examine how and why this research selected. Then the process will follow by introducing procedures that considered for designing of the application. The next step is to examine and select the theoretical and practical frameworks to implement the design. Finally, the process followed by the implementation and the logic behind of the features of the voice map application. In this chapter, the details of all the approaches that mentioned before is presented.

3.1. Case study

An intelligent assistant is equipped with a range of technological tools that allow us to give voice commands that are then translated into specific tasks. This process involves using advanced voice recognition capabilities and artificial intelligence systems to interpret requests and convert them into device commands. *Despite their ability to cover many scenarios and a large amount of content, these assistants lack advanced services related to geographic orientation[13].* Geographical applications tend to be more complex, which has resulted in them lagging behind general tool adoption. The industry of virtual assistants started to thriving into the industry of smart home from 2014 when Google and Amazon introduced the first version of the virtual assistants tools. But the history of the virtual assistant originated in the Apple Siri tool which was available on the iPhones 4s. The advent of the technology in this field and more important the pace of it, is growing at an unprecedented rate. And more important, there is the lack of the studies and research for implementing voice speech recognition technology on GIS system, on advanced, current, and modern infrastructure. Platforms and spatial data providers which are dealing also with geospatial data visualization and analysis, needs to improve their user's experience in general standards and particularly for impair clients. The industry has been specialized mainly for professional people who has the domain knowledge of working in practical manner with geospatial data and beyond that the platforms which are designed to present geospatial data visualization to public user such as *Data wrapper[2]*, and *flourish[3]* still lacks the generalization of cartographic and analytical tasks. The implementation of voice speech recognition technology on web browsers to support geospatial data visualization applications is a practice that still requires initial development.

In consequence of the absence of the first steps, the other industries such as Machine Learning and Artificial Intelligence, did not have any great progress nor approaches for representing applications with using of the virtual assistants specialized for geographical data. *Qwhery[5]* might be one of the numerous and official example of it, but the problem of the scalability of the geospatial data for using in dimension of a country or even worldwide temporarily decelerate the progress.

The lack of sufficient studies and research on appropriate verbal choices for geospatial data visualization and cartographical tasks is a critical need that deserves attention and action. In addition to facilitating effective communication and precise data interpretation, a standardized terminology can serve as a basic infrastructure for natural language processing (NLP) applications for geospatial data. NLP applications require a deep understanding of the specialized terminology used in geospatial data visualization and cartography to accurately analyze and interpret data. Therefore, the establishment of a standard terminology is not only important for communication and reducing errors, but also for providing a foundation for NLP applications. Collaborative efforts among researchers, practitioners, and developers in this field are needed to conduct more studies on this topic and establish a common language and terminology for geospatial data visualization and cartographical tasks. With a standardized terminology in place, professionals in this field can not only improve communication and reduce errors, but also pave the way for more advanced and effective NLP applications for geospatial data.

One area of education that can benefit from the use of technology is geography, which traditionally relies on static maps and textbooks. By using virtual assistants and voice technology to show maps and geographical data, educators can provide a more interactive and engaging learning experience for students. Virtual assistants and voice technology can be used to show maps and geographical data in real-time, allowing students to explore and interact with the data in a more dynamic and immersive way. This can help to increase students' understanding of geographical concepts and improve their spatial reasoning skills. In addition, virtual assistants can provide students with personalized assistance and feedback, allowing them to learn at their own pace and level. The use of virtual assistants and voice technology can also help to increase the accessibility of education, particularly for students with disabilities or who may struggle with traditional learning methods. By providing a more inclusive and engaging learning experience, educators can help to close the education gap and ensure that all students have access to quality education. However, *Thomas Gilbert on his studies, proved the usage of voice speech technology on classroom, can be beneficial in different aspects, but still this filed requires cautious planning and suitable applications, which provide all the necessary materials for enhancing the environment of classrooms for teaching[26].*

With the advancement of technology, geospatial data visualization has become more sophisticated and complex, making it challenging for users to interact with and understand the data. However, by embedding interfaces and platforms that deal with geospatial data visualization with voice technology, users can experience a more intuitive and interactive interface, leading to an improved user experience and increased accessibility. Voice technology allows users to interact with the data using natural language, making it easier for them to explore and understand the data without the need for complex navigation or technical skills. This can be especially beneficial for users who have visual impairments or other disabilities, as it provides an alternative mode of interaction with the data. In addition, voice technology can help to increase the speed and efficiency of data exploration, as users can quickly navigate through the data using voice commands. Furthermore, embedding voice technology in geospatial data visualization platforms can lead to an improvement in user interface design standards. By using natural language to interact with the data, designers can focus on creating more streamlined and user-friendly interfaces that reduce the cognitive load on users. This can lead to a more pleasant and enjoyable user experience, which in turn can lead to increased user engagement and adoption. However, it is important to note that the implementation of voice technology in geospatial data visualization platforms requires careful consideration and planning. Designers must ensure that the technology is integrated in a way that complements the existing interface design and functionality, rather than disrupting or replacing it. They must also consider the potential security and privacy implications of using voice technology with sensitive data.

3.2. Application domain and phenomena description

To describe the requirement, schema, and usage of the voice map application, there is the need to define and describe the application domain and phenomena description. This step considered as a phase in software engineering, which aims to recognize the requirement analysis for development and design of an application. Initially there is the necessity to describe clearly all the elements and the component that involved into the application environment, to gain a clear understanding from the relationship and functionality of the components together and identify the functional and nonfunctional requirement in the next step.

3.2.1. System

The system is a web application, composed of three main pillar portions: Frontend, Backend, and the Database. Each of these components interact directly and indirectly together, to achieve some predefined goals. The system is a clients-server architecture composed by:

1. The client is a web browser which displays the output interface of our system, with which the user interacting. This client can visualize static data composed of description elements of the project and of the components that build-up the application.
2. The web browser interacts with dynamic elements. These elements will make the requests to the web browser, providing the content and in response, a part of web browser will display this information. The user can interact with the information in real time, in order to visualize it in different ways.
3. One of the most important elements in the system is the microphone, which the web browser asks the permission from the user, to activate it through web browser. This crucial element will transfer the requests of the users via transcription of the verbal commands and sent it back to the server. In plain language there is a package of rules and protocols that already has been designed and embedded on the web browsers, which allows the systems to interact particularly with the users through voice and microphone.
4. Since this application is a part of the development and new feature for an already existing platform which is called BStreams, the system has an independent URL [49?] which is accessible to the public with registration. The domain .io is a country top-level domain name. It is popular as a generic domain in tech world since IO or I/O means Input/Output in computer science.
5. The system hosts by AWS (Amazon Web Service) cloud. All the components of the systems such as the codes for the frontend and backend, the repository for static data, and other necessary elements are hosts by S3 amazon bucket service, Lambda function gateway.

6. The DBMS is supported by the PostgreSQL, which interact with the infrastructure provided by the already existing architecture of the BStreams. The platforms dedicated tables for organizing the interactions and sourcing the elements of the system. The DBMS considered the security of the users, and the passwords, will be secured by hash techniques.

3.2.2. Users

As it discussed in the case study section, the goal of this application to present a prototype and execute it on a platform, to enable the users, to talk and interact with geospatial data visualization through verbal communication and commands. The users have the possibility to interact with the application through two possible way:

1. Already built visualization, which enables the users to navigate and interact with the map and visualized geospatial data with some limited commands (View mode).
2. The users can build her/himself the visualization and can interacts with the application in all the embedded voice commands (Edit mode).

The information displayed could be different in aspects of consideration also the level of the domain knowledge of the users. The application provided two formats of the visualization:

1. Marker map
2. Graduated color maps

Based on the cognitive load of the users and meaning of the tasks in terms of the logic and concept of doing customization with voice command and its feasibility for the users, commands has been designed and categorized also based on the theoretical model that this study selected. However, the domain knowledge of the users can be an important variable in comprehension of the results, and also the commands itself, this study dedicated specific efforts to remove the boundaries for this matter in two significant matters: first the commands have aliases, to provide the freedom for users to use their own vocabulary and words of chose. The aliases derived from a survey that has been disturbed among general respondents with variety of domain knowledge, Education, Age, Gender, Level of English proficiency, and mother tongue. Second the concept of geospatial data visualization and cartographic tasks can be eased and normalized by familiar phenomena such as "COVID-19 map". Therefore, the practice of dealing with geospatial data with verbal commands can started with this demonstration of this familiar phenomena.
Moreover, the case study has already examined the possible users and use cases. *Sussane*

Bødker in her article discussed about the use of scenarios in usability testing and proposes the use of multiple scenarios with different purposes and emphasis depending on the stage of the evaluation process [14]. She introduced three category of scenarios and represented them with example and justification. The first approach that she discussed was "Usability testing of prototypes". The scenarios are designed to motivate the user to carry out a certain activity, and the same scenarios are used throughout the test cycle. The article also proposes two supplementary strategies for making better use of scenarios in design, one for early testing of overall design ideas and the other for more focused testing later based on what the design needs to deal with at that particular point in the process. Influencing by this methodology, the user and use cases can be categorized into main domains and related activities that it is possible to be done with the application. The related domain of the application can be:

- Linguistic
- Education
- Field work
- Disaster
- Education and Entertainment

Related activities also can be grouped in:

- Mapping
- Reasoning on spatial relationship
- Analysis and decision making
- Place recognition
- Cartographic representation analysis

Moreover, the users can be categorized based on:

- Normal user
- Impair user
- Domain knowledge
- Lack of domain knowledge

Depending on the categories and the users profiles the following can be the possible use scenarios of use cases:

1. One use case is for users to explore different places on the map. The application provides an interactive map on the browser that responds to voice commands. For example, the user can say "Find Mantova" to obtain information about the location of the city and other nearby cities. The user can also use voice commands such as "Zoom in", "Zoom out", "Pan left", and "Pan south" to explore the area. This use case is relevant to all users and falls under the domain of linguistic place recognition.

2. Another use case is for users to visualize spatial relationships on the map. For example, the user can visualize a punctual marker map of noise pollution from Pune-India and query the markers by saying "Show me the markers with maximum noise pollution". The application will query the pins and visualize the places with maximum noise pollution. The user can then navigate through the area and understand that the areas with high noise pollution are mostly streets with commercial land use. This use case is relevant to users with domain knowledge and falls under the domain of linguistic reasoning on spatial relationships.

3. A third use case is for teachers to use the application in education settings. For example, a teacher can use the application to introduce the topic of geographical location or topography by visualizing a map and exploring different spots using voice commands. The teacher can navigate to different cities and change the basemap to topography, all through voice commands. This use case is relevant to both normal and domain knowledge users and falls under the domain of education mapping.

4. A fourth use case is for users to modify the color palette of a graduated color map. For example, a teacher has a graduated color map of the "Mean temperature of USA from 1900-2020" but finds that the color of the map is not suitable for representing the temperature. The teacher can modify the color palette by saying "change the color palette" until they find a suitable color palette for presenting the phenomena. This use case is relevant to all users and falls under the domain of education cartographic representation analysis.

5. A fifth use case is for users to query and filter data on the map. For example, in the event of a natural catastrophe, a user can access the application and visualize a graduated color map of the population in the affected area. The user can then query and filter the data with a command such as "show me the commune with a population of more than 100,000". This use case is relevant to users with domain knowledge in disaster management and falls under the domains of mapping and analysis and decision making.

6. A sixth use case is for users to highlight points of interest on the map. For exam-

ple, a geologist or environmental researcher can use the application to demonstrate gathered data from the field and monitor a phenomenon. The user can filter the data using voice commands and highlight the points of interest. This use case is relevant to users with domain knowledge in field work and falls under the domains of mapping and analysis and decision making.

7. Finally, a seventh use case is for users to simply explore the map. For example, a user who is interested in geospatial data or hobbyists can access the application, locate themselves, and then explore the map and the area around them. This use case is relevant to all users and falls under the domains of education and entertainment, as well as place recognition and positioning.

3.2.3. Product functions

The voice map application provides the following functions:

1. Navigation with voice: the embedded Voice Speech recognition API and Geocoding feature, enable the users to navigate through different locations. The application designed to have different level of the zooming with respect the provided tag of the entity that will recognize by the Geocoding library. The entities can vary from the highest zoom which is an address or postcode to the lowest zoom which is continent or a country.

2. Cartographic and visualization tasks: The voice map application can execute some tasks regarding the cartographic and visualization tasks such as:

   - Changing the basemap.
   - Altering the color palette.
   - Customization of the color and format of the markers.
   - Increase/decrease the opacity of the map.
   - Convert the graduated color map to marker map.
   - Hide/show the legend.
   - customization of the classification for graduated color map.
   - Hide/show the layers of data.
   - Increase/decrease the radius of the cluster markers.

3. Data analysis and filtering: The application allows the users to query and filter data with respect to the phenomena and the level of subdivision of the map. With respect to the type of the visualization that users choose, the qualitative and quantitative format of the data can query in different way. But the application should provide the scalability of receiving and interpreting the different methods for filtering the data.

3.3. Theoretical framework

Conceptualizing based on a theoretical model can be beneficial for research in several ways. Firstly, theoretical models provide a framework for research by outlining key concepts, variables, and relationships between them. This structured approach can help to guide research design and data collection. Secondly, theoretical models can generate testable hypotheses about the relationships between variables. These hypotheses can then be tested through empirical research, contributing to the advancement of knowledge in a particular field. Thirdly, theoretical models can facilitate communication and collaboration between researchers working in the same field by providing a common language and conceptual framework. Lastly, theoretical models can enhance our understanding of complex phenomena by providing a way to organize and make sense of data. This can lead to new insights and discoveries, ultimately contributing to the development of theories and models in a particular field of study.

When requesting geographical information in natural language, users may use vague or ambiguous terminology, which can cause confusion for the system. For instance, the user may use imprecise spatial references like "over there" or "nearby" instead of specific location names or coordinates. Similarly, non-spatial concepts like "close proximity" or "large area" can also be unclear without additional context. These ambiguities can make it challenging for the system to accurately interpret the user's request and generate an appropriate response. Therefore, natural language processing systems used for geographical information must be capable of handling such imprecisions by utilizing techniques such as context analysis, machine learning algorithms, or by prompting the user for additional information. By addressing these challenges, the system can provide more effective and efficient responses to the user's requests.

3.3.1. PlanGraph methodology

Using natural language, gestures, and other human modalities can be a potential alternative to interact with computers. However, despite its promising advantages, this method

has not yet been explored for interactions with geographical information systems.
To expand further, natural language is the way humans communicate with each other in everyday life. This includes the words we say, how we say them, and the context in which they are used. Geographical information systems are computer systems that capture, store, manipulate, analyze, and present spatial or geographic data. They are widely used in fields such as urban planning, environmental management, and transportation. However, interacting with these systems can often be challenging and require specialized knowledge.
By incorporating natural language and gestures into the interaction with geographical information systems, it could potentially make the system more accessible to a wider range of users. This could lead to new insights and discoveries in various fields, as well as improve decision-making processes. *G Cai, H Wang, A M MacEachren and S Fuhrmann studied a new prototype and a concept to ease the path for facilitating the GIS applications, with conversational features[15]. Their introduced concept later has been uses and and implements on a sample-human GIS on 2008 by Hongmei Wang and colleagues[55].* The use of the term "conversation" as a metaphor for human-GIS interaction is particularly advantageous when dealing with multimodal interfaces. This is because conversations are multi-turn and require contextual understanding, which can be achieved by incorporating various modes of communication such as natural language and gestures. The process of constructing requests in a conversational interface is incremental and interactive, meaning that the user can provide input in smaller segments, and the system can clarify any uncertainties or mistakes in real-time. This is different from the traditional query-based systems, which strictly follow a three-phase process, including collecting user input, processing the query, and generating a response. In these systems, each phase must be fully completed and successful before proceeding to the next phase. In contrast, conversational interfaces enable a more flexible and natural interaction, where the user and the system engage in a back-and-forth dialogue to arrive at the desired result. This iterative approach allows for a more efficient and accurate exchange of information.
The obstacles and issues for embedding the natural language processing on a GIS interface requires a schema or a model that eases the process of retrieval of information from database and reduce the cognitive load from the user's perspectives. The PlanGraph methodology which was introduced by G Cai, H Wang, A M MacEachren and S Fuhrmann, indicates this challenge and considers as a solution, which already implemented and testes by the Hongmei Wang and colleagues on their GeoDialogue.
The discourse context of a human-computer conversation is represented as PlanGraph or Plan Graph. This concept is similar to the notation of the recipe graph (Rgraph) which has been developed by Lochbaum (1194, 1998), except the aspect that PlanGraph

Actions	Recipes	Plans
An action refers to a specific goal as well as the efforts needed to achieve it. An action can be basic or complex.	A recipe describes the components of an action in terms of parameters, sub-actions, and constraints	A plan corresponds to a schema describing not only how to perform an action, but more important the mental attitude towards it such as beliefs, commitments, Execution status

Table 3.1: Three pillar concepts of the PlanGraph

increases the Rgraph theory on handling the knowledge-preconditions in tasks that are collaborative. The Plan graph describes its structure by representing three main concepts, which has been explained on Table 3.1 With all their parameter.

3.3.2. Action

As the definition of the action, explained on the Table 3.1 The action can be categorized into two main groups, which bellow with examples has been clarified:

- Basic: Based on the definition of the PlanGraph, a basic action refers to the types which can be executed just only with one user and directly within the interface. It points out the simplicity of the task, and without accessing to other infrastructure on the platform you can achieve the goal. Example of the basic actions can be “changing the basemap” or “change the color palette”.
- Complex: On contrary with basic actions, complex action is nor executable directly withing the interface because certain knowledge and pre-conditions or details about the performing of the action is needed. Therefore, in case of having a complex action, the concept of the recipe is needed.

3.3.3. Recipe

A recipe describes the action’s components, in different groups:

- Parameters: Which refers to the knowledge, and pre-condition for executing sub-actions of that recipes. All the possible values i.e., input values from users, thresholds, and their possible formats need to be collected.
- Sub-action: it defined as the basic action which needed to complete the complex actions. In a plain language a sub-action considered as the branch of the whole action, which need to be achieved to accomplish the main goal on the top of the

nodes.

- Constraints: it specifies any pre- or post- condition and partial orders of the sub-actions.

3.3.4. Plan

A plan describes the flow of the performance of the action(s), and beside of it, the mental statement regarding the action(s) is also taken into consideration. The mental statement that involved into executing of an action is categorized into three main groups:

- Beliefs: It refers to all the mental attitudes that a user has towards the performance of an action. A user that participates in execution of a plan in an action, has to identify the recipes (parameters, sub-actions, and constraints), and has the capability of doing an action.
- Commitments: It defined as the level of collaboration of the user to accomplish an action. In many cases the commitment of a user to an action described as the sources such as time that s/he must spent to execute the action.
- Execution-status: A plan can have status that declares the level of the execution of it. It can be "executable", "not executable", "executed with success", or "executed with failure".

The Figure 3.1 and Figure 3.2 Are describing the schema of a recipe and a plan:

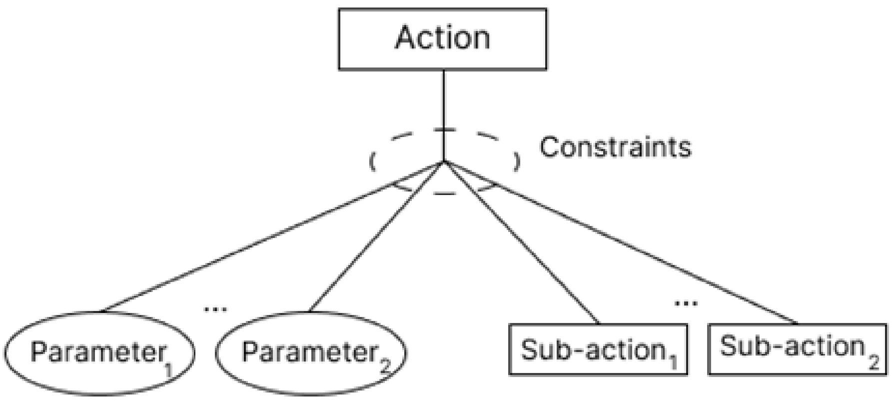

Figure 3.1: Structure of a Recipe

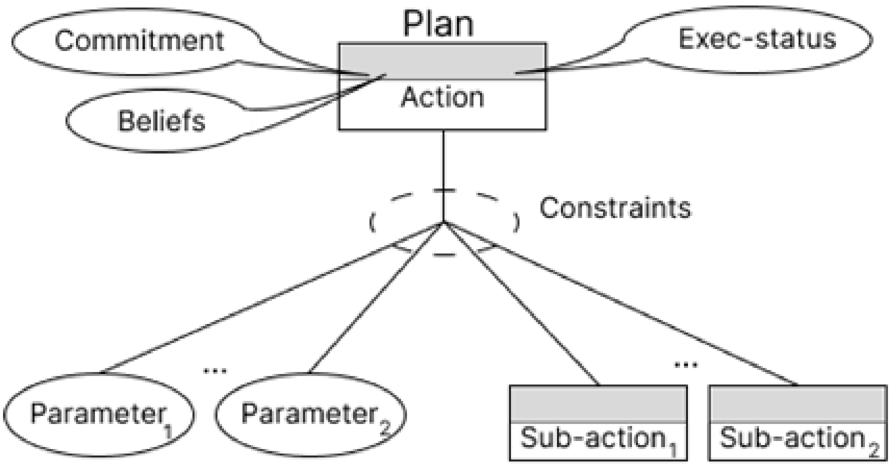

Figure 3.2: Structure of a Plan

Adopting the concept of the PlanGraph with all the mentioned structures, the final definition of it is: PlanGraph is a diagram that shows all the plans and subplans involved in achieving a big, complex goal. Usually, the PlanGraph starts with the main goal and then breaks it down into smaller subgoals or actions using recipes. Because of this, the PlanGraph is usually organized hierarchically with different levels of plans.
Figure7. Explains the general structure of PlanGraph. Nodes with oval shapes pointing out the parameters, and nodes with rectangle shape represent subplans. A plan as a child node of the root, considered as the plan for identifying the parameters. For example, the, Plan χ_1 is for identifying the parameter1 of Plan α.

Type of actions in geographical discourse

Since the PlanGraph designed for Human-GIS computer, which was able to handle a conversation, the model should adopt its schematic with the formats of the actions that is adoptable, visible, related, and intuitive with the geospatial data processing and visualization. Therefore, all the actions in context of a Human-GIS computer are categorized into main four categories:

1. Type I: Actions that deal with the acceleration of spatial data include retrieving a map layer, for example, "retrieving a map layer".
2. Type II: actions that are dealing with analytical tasks (spatial and/or statistical) such as "finding spatial clusters" and "areal aggregation".
3. Type III: actions that are cartographic and visualization tasks, for example "zoom

in/zoom out", "pan", and "highlighting".

4. Type IV: actions that are domain specific tasks such as "Planning for evacuation" in a hurricane response domain.

Embracing the PlanGraph for voice map application

Utilizing the PlanGraph model into voice map application, leaded to structuring all the steps that its necessary for executing all the considered task for the application. For this purpose, the first step was to list all the features that is available already on the platform, for two formats of visualizations. After derivation of all the features, the effort spent to find the notation of the tasks. Some tasks have some characteristics in mutule, while the others designed to accomplish a particular task. Since the whole concept of the two formats of geospatial data visualizations (Marker, and Graduated color map) is to demonstrate geospatial data, the two main notations for all the tasks have been considered:

1. Cartographic tasks: This group of tasks, aims to enable the user to accomplish proper visual modification for the two maps. Since BStreams is a web application, and the architecture of the application is client-server, the tools and features provided should not be heavy in terms of the process and loading for the users. These important reasons, concluded to simple but usable cartographic tools. The Figure 3.4. And 3.5. Visualize some of the cartographic tasks with their notation on the BStreams platform.

2. Analytical tasks: The aim of each geospatial data visualization available on BStreams' platform is to demonstrate data, but in depth the embedded tools also let the users do analytical tasks. Although there are some limitations in terms of embedding a web GIS and using the analytical tools that already exists in professional software and platforms, there are some features which simplified based on usability and visibility for the public users who are not familiar with the domain knowledge. These analytical tasks mainly concluding to filtering geospatial data. The feature "show filter panel", enable users manually to filter data with already predefined methods. For Graduated color (Choropleth) map, which is a thematic map for a selected quantitative phenomenon, the feature can query the phenomena with input values by the user. On the other hand, the Marker map which is punctual map aiming to visualize the position and distribution of a dataset, the filter feature can query the visualized pins based on both qualitative and quantitative phenomena. The Figure 3.6. And 3.7. demonstrate the usage of the filter feature on both graduated and

marker maps.

Figure 3.3: Layout tools which aim to do cartographic modification such as changing the format of the marker

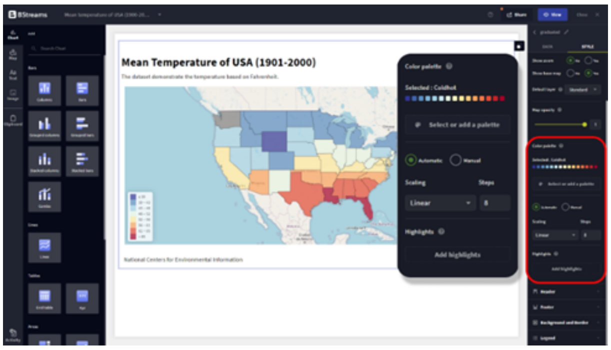

Figure 3.4: Color palette selection for the graduated color map, enables users to control the color pattern of the map.

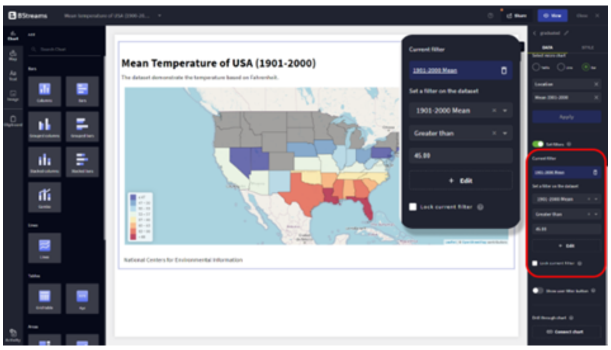

Figure 3.5: Filtering the Mean temperature with the values greater than 45F

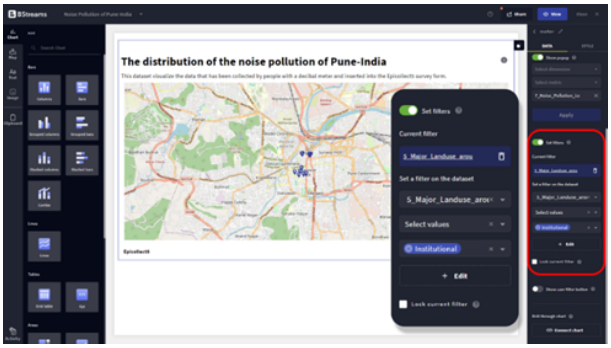

Figure 3.6: Filtering the markers map of noise pollution with qualitative filed of only Institutional landuse

Classifying the tasks and their features into two main categories, eased the proceeding to conceptualize the tasks into the PlanGraph methodology. Since the main type of the task has been distinguished, for fulfilling the remaining component of the PlanGraph structure, which are parameters, constrains, and sub-actions, the process of inquiry through the User Interface continued. Moreover, the main functions of the two visualizations in the backend, and also the default configuration file of them on the database which reveals information regarding the parameters and the initial setting of the two maps, helped to acquire a better understanding of the necessary parameters and conditions for each derived task. The Listing 3.1. Visualize a section of the configuration file of the Marker map in format of .JSON which supported the process of parameter and constraint derivation.

```
{
    "layout": {
        "marker": {
            "shape": "pin-basic",
            "size": 5,
            "color": "rgba(48,63,159,100)"
        },
        "mapOptions": {
            "showZoom": 0,
            "showLayers": 0,
            "defaultLayer": "OsmStandard"
        }
    }
}
```

Listing 3.1: Configuration file of the Marker map

The process of creating a PlanGraph model went smoothly after exploring through the configuration code of each map and retrieving the hints from the User Interface (UI) itself. The UI of the BStreams depends on interaction between the user and the platform itself to accomplish the tasks and modification, however a normal process of examination of each tool and feature was necessary to gain the accurate format, threshold, and structure of the inputs that will receive from the users through the UI. Following examples are the usage of the UI features for derivation of constraints, and parameters:

1. The first task involves changing the steps of graduated color, and requires inputs such as methods, number of steps, color, thresholds, and label for manual steps. The inputs for this task are primarily strings and integers, and there are no specific conditions to consider. The figure 3.7 visualize the detection of the parameters from the user interfece.

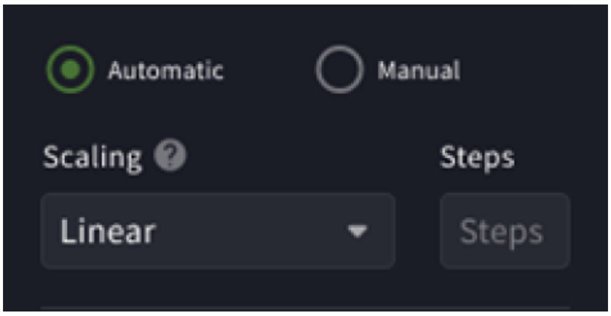

Figure 3.7: Classification method for graduated color map on UI

2. The second task pertains to changing the opacity of the map, which can be achieved by providing a number as an input or using a slider. The inputs for this task are integers, and the opacity threshold is between 1 and 10. The figure 3.8 shows the sliders and formats of the inputs that retrieved from the user interface.

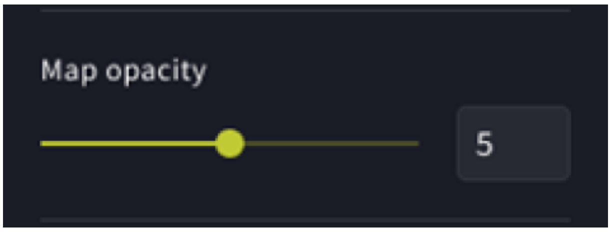

Figure 3.8: Map opacity slider on UI

3. The third task involves changing the basemap, and users can select from the available basemaps by providing a string input. The realted feature from the user interface shown on figure 3.9 There are no specific conditions for this task.

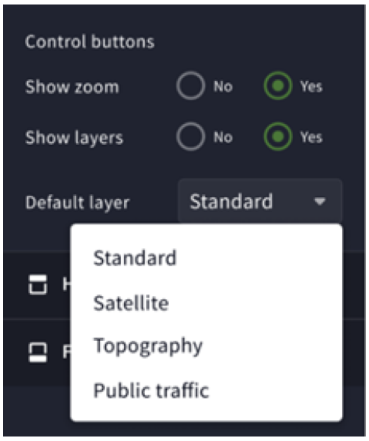

Figure 3.9: Control layer feature on UI

4. Finally, the fourth task focuses on changing the color of the marker. Users can select the color using a slider or by inserting the HEX/RGB color code. The inputs for this task are integers, there are no conditions to consider and the features related to this task visualized on figure 3.10.

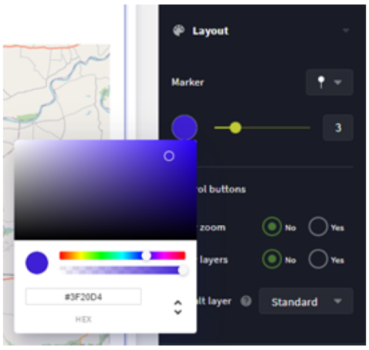

Figure 3.10: Color customization feature on UI

By retrieving all the parameters, constraints, and all the necessary component for a PlanGraph models, the process of conceptualizing the actions finished. As the Figure 3.4. Reveals the schema of the model, we continued with the same procedure as mentioned on the methodology of the PlanGraph. Starting from the root and declaring the main action/plan the parameters and constraints for each task proceeded. After completion of the model for each task, the next step was to identify the type of action. As the PlanGraph types was discussed in a few sections before, and the hints that provided on the description of the PlanGraph, each model can be categorized into variety of types depending to the point of being a complex or basic action. The models can be basic if the user does not need to derive any pre-knowledge and all of the provided visual information on the interface is sufficient enough to comprehend the task. On the other hand, a task considered as a complex in context of our application, if the user needs to have domain knowledge to proceed with task. For instance, "Zoom in" or "Pan to left" considered as a basic task, since these are the very initial steps that a user who can interact with computers and maps can do it without any difficulties or pre-knowledge. In contrast the task "Changing the steps for graduated color" considered as a complex task and it requires sub-actions and plans for accomplishing the task. This is because of the necessity of having domain knowledge regarding the processing and working with a thematic map, which not a user might be acknowledge about it. With understanding the type of the actions in terms of being complex or basic, the process of identifying the type of each action progressed. The Figure 3.11 demonstrate the model based on the PlanGraph methodology for the basic task "Zoom in/out". The root plan is to accomplish the task either zoom in or zoom out, and the parameters are the window size of the map, the predefined scale for zooming in or zooming out, and the sub-action of reloading again the basemap with all the pre-loaded data on the map. Based on the description of the type of the action, the root plan considered as a type III action, since its only containing the modification of the visualization, and the sub-action considered as the type I.

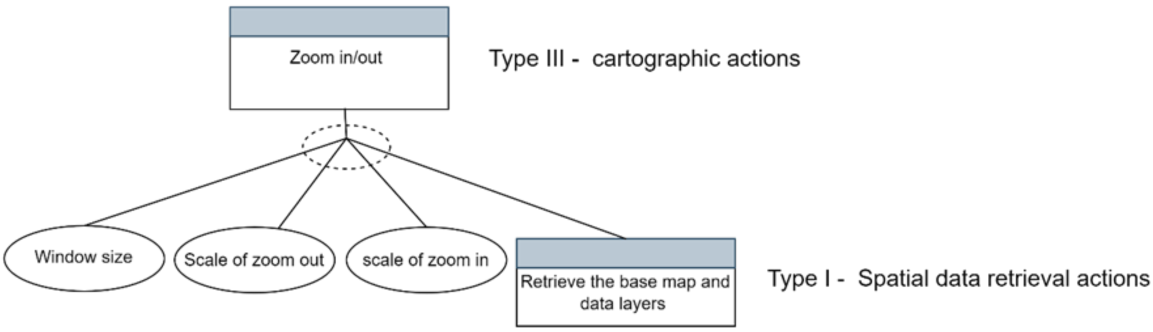

Figure 3.11: The PlanGraph Model of the task "Zoom in/out"

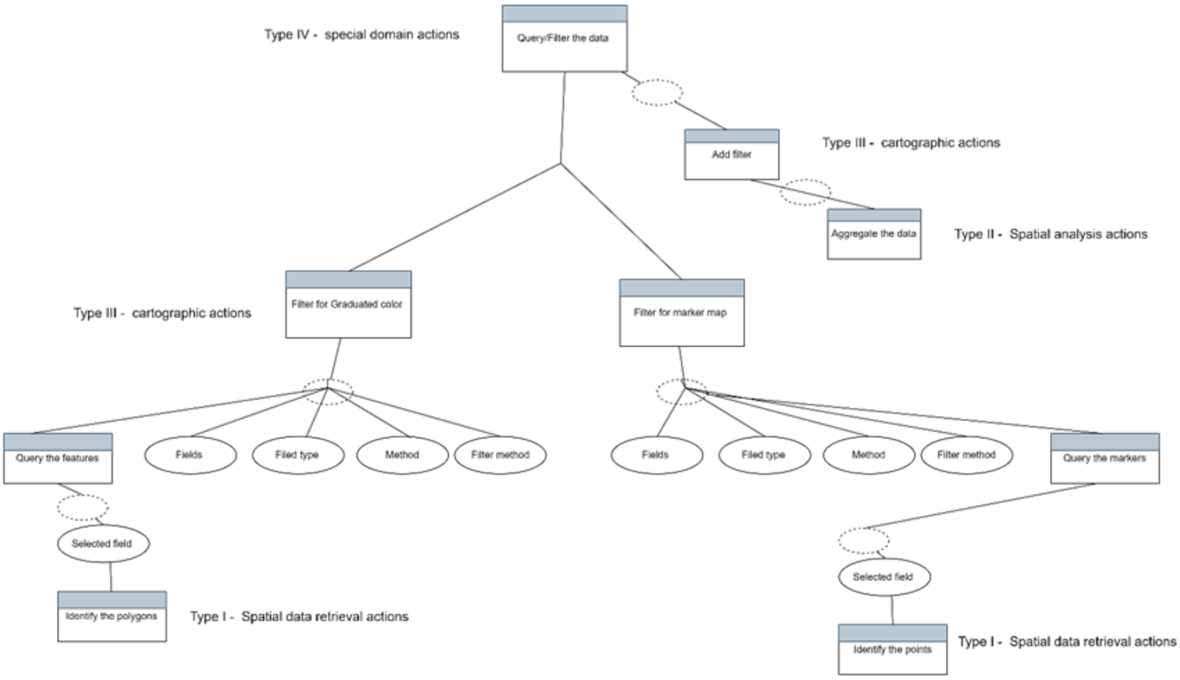

Figure 3.12: The PlanGraph Model of the task "filter"

The Figure 3.12 On the other hand, visualizes the task "filter" which can be available for both map, and since the domain of the action requires pre-knowledge about the data and also methods for querying the data, considered as a complex action. For accomplishing the task there are sub-actions and plans to retrieve the necessary parameters. As the Figure 3.12 Indicating the plans should be identifying for which type of the maps, the filters is going to be apply. Depending on the path and selected format of the map, the parameters only alter in the format of the spatial data retrieval. If the graduated color map, selected for the query, the spatial data retrieval in the very basic level is geometries of the polygons. On the other hand, if the selected map is the marker map, the only data that needs to be acquired is the latitude and longitude of the marker. The rest of the parameters is mutual since the filter feature has the same functionality for both maps. Depending on the selected field, which can be qualitative or quantitative, the type of the field will be identified. And by selection of the filter method and input values by the user for breaking the data, the process of the filtering will proceed. Consider that this whole process is a manual process which done by the user through the UI interaction and using the mouse and keyboard. The final goal of the conceptualizing of the task into the PlanGraph model is to create an infrastructure to automize the available and visible tasks with voice command.

The conceptualization of a task into the PlanGraph model involves breaking down the task into smaller sub-tasks and representing the dependencies and relationships between them.

This process provides a comprehensive understanding of the task and enables the creation of a detailed plan of action. The ultimate goal of this process is to create an infrastructure that can automate the execution of tasks using voice commands. By converting the task into a PlanGraph model, the dependencies and relationships between each sub-task can be visualized, and potential bottlenecks or inefficiencies can be identified. This provides a clear understanding of the various steps involved in the task and the order in which they need to be completed. This understanding can then be used to develop our application that can automatically execute the sub-tasks based on voice commands.
Automating tasks through voice commands can have significant benefits, particularly in situations where hands-free operation is essential or where multiple tasks need to be executed simultaneously. By integrating voice command automation into the PlanGraph model, it is possible to create an infrastructure that can execute complex tasks with minimal human intervention, thereby increasing efficiency and reducing the potential for errors.
On the link you can access to a complete schema of all the models of the tasks that has been identified and conceptualized with PlanGraph attached.

3.4. Survey

Over time, the speedy progress of Natural Language Processing (NLP) technology has presented significant possibilities for investigating a natural-language-based approach to interacting with data visualization [11],[57]. however, the efforts for easing this technology for the geospatial data is not considerable. The utilization of natural language text is a vital source of geographical information, including spatial locations, spatial relations, and non-spatial attributes. *Extracting geographical information from natural language text can not only enhance the information repositories of Geographic Information Systems (GISs), but it can also enhance the representation and comprehensibility of GISs[6].* From a scientific perspective, the conventional approach of using a mouse and keyboard to interact with geospatial data visualization, particularly maps, is considered familiar to most users. However, there is a need to investigate how verbal commands could potentially enhance the user experience in performing tasks such as panning or zooming. To achieve this, it is essential to have appropriate and practical terminologies that are coherent and meaningful to users.
To address this issue, a survey was designed and distributed among the general population to evaluate and extract the lexical choices that users make when interacting with maps on a web browser. The survey aimed to assess users' vocabulary in three main groups of tasks based on the PlanGraph methodology: special domain actions, cartographical

actions, spatial analysis actions, and spatial data retrieval actions.

The questionnaire consisted of two main sections, with each question accompanied by visual content in the form of GIFs to convey the characteristics of the tasks that the application should implement after receiving the commands. To further investigate the language use and communication strategies of users, the survey included questions on the respondents' age, gender, level of education, field of education, mother tongue, and level of English proficiency. The purpose of this was to identify potential patterns or associations between user characteristics and the lexical items employed.

By analyzing the survey results, we can gain valuable insights into users' preferred vocabulary for interacting with maps on a web browser. This information can be used to improve the design of map controls and enhance the user experience by incorporating commonly used verbal commands into the interface. Additionally, understanding the language use and communication strategies of users can lead to more effective communication and better design of interactive systems in the future.

3.4.1. Visual contents

The inclusion of visual content in the form of GIFs is an innovative approach to improve the comprehension of the users about the characteristics of the tasks that the application should implement after receiving the commands. By embedding visual content, the questionnaire provides a more comprehensive and engaging way for users to understand how the application should respond to their commands. It enhances the communication process between the user and the application, reducing the likelihood of errors and improving the overall user experience.

The use of GIFs is particularly effective in helping users to understand complex spatial data visualization tasks. Visual content can quickly communicate the expected outcome, making it easier for users to understand how to interact with the application. The visual content also provides users with a more intuitive way to learn how to use the application, reducing the time and effort needed to become proficient in using the tool.

Moreover, the inclusion of GIFs in the questionnaire can be beneficial in various other ways. It can help to make the survey more engaging and interactive, which is likely to increase response rates. Additionally, embedding visual content can make the survey more accessible to users who may have difficulty understanding textual descriptions of the tasks.

3.4.2. Open questions

Given that the primary objective of the survey was to assess users' proficiency in using vocabulary related to geospatial data, open-ended questions were deemed the most appropriate type of question. This was because it provided the respondents with the freedom to express their own thoughts and ideas without any hint or suggestion provided by the survey. Open-ended questions allowed the participants to give their own perspective on the tasks and commands they used to interact with geospatial data. It also gave them the opportunity to use their own words and phrases to describe their commands, making the results more authentic and reliable.

The survey was divided into two distinct groups of questions: navigational tasks and geospatial data visualization and analysis tasks. The objective of the first group was to comprehend the terminology used for geocoding and to identify the possible verbs and collocations that a user could use to command the application to find a specific location. The questions in this group also investigated the commonly used cartographic commands such as zoom in, zoom out, and panning in different directions such as pan to the left or pan to the right.

On the other hand, the second group, which was titled "talk with geospatial data," aimed to understand how users could perform common tasks that typically require the use of a mouse and keyboard with voice commands. For instance, one of the tasks investigated in this group was how a user could filter geospatial data using voice commands. However, since the context of querying requires prior knowledge about the data, this task was also dependent on other variables. Therefore, an example of a thematic map of COVID-19 in Italy was used to simplify and clarify the task. The other tasks included in this group were the customization of the colors, size, and other features of the map.

To prevent respondents from becoming bored or frustrated with the open-ended survey questions, the survey incorporated a storytelling theme for both categories. This approach allowed respondents to follow the flow of the tasks and better understand how they would be implemented in the application. By providing context and a narrative structure, the respondents were able to engage more deeply with the questions and provide more thoughtful and accurate responses. This helped to ensure that the survey yielded useful and actionable insights into users' vocabulary and language use when working with geospatial data.

3.4.3. Parameters, respondents, and distribution

The application is designed in the English language, therefore parameters such as level of the English language proficiency and mother tongue, was taken into consideration. The other parameters such as Age, Gender, Level of the Education, and the field of study (which was not mandatory) to respond, asked from the users. This type of question, which are not sensitive information, benefits the survey to determine the relationship between the used word and the characteristic of the respondents. *Brian N. Larson, on his studies presents a guideline for using of the gender in surveys and research topics related to the NLP[36]* . On his studies initially he introduced three approaches for understanding the bias between the natural language and the gender. The first approach is that some words have more usage among some specific gender, for example the word "folk". And this is only because of the concept behind the word "folk", which is refers as a "heteronormative gender binary". Therefore, by including this type of question, the derivation of the biases and relationship between vocabularies, sentences, and their format became easier.
The respondents were not limited to a specific group of people. The survey distributed through social media channels to contain public. As it mentioned before, one of the main approaches of this study is to remove the boundaries for requiring the domain knowledge for dealing with geospatial data. Therefore, the efforts to collect the answers from not only academical group of respondents, but any group of people who can communicate with English language (but in any level) were considered.
The survey was designed through Google form, because it is one of the most popular forms of questionnaire, and it allows all the people to cooperate and provide responses without any difficulties. The survey considered to not collect the email of the respondents, because the email considered as a sensitive information, and there was no need to collect for this survey this type of a sensitive information. A document regarding the "Terms of privacy" were designed by the legal studio department of the Environmental, and Civil Engineering, which also approved by the legal office of the Politecnico di Milano. The survey was distributed on a 20-day period and after that the survey did not accept any responses.

3.4.4. Data cleaning and procedures

As the survey is already designed to evaluate the most used terminology for dealing with geospatial data visualization and analysis, the answers are containing the strings, characters, and symbols. The results were collected from the Google Form in format of "CSV". The data on each row were separated into words by usage of the Text to Columns function of the Microsoft Excel. The selected and suitable approach was delimited the words with “space”, “semicolon”, “comma”, and “tab”. For some columns with manual investigation if some specific symbols were repeated, we used them for splitting the sentences to words. The process continued for all the 12 questions available in the two main sections. After derivation of the all the separated words, some of the rows in particular questions, were discarded as null, since we asked the users that if they do not know the answers simply write “I do not know”. Moreover, there were some outliers for some questions, and people lost their interest to the survey at the final questions, so with manual reading and investigating through the data, the process of cleaning was finished.

One of the main important findings through the process of cleaning of the data was the point that respondents mainly used the type of a “command” sentences during the survey. The survey was aimed to simulate a scenario of a story, that user can go through and interact with the map. However, there were some specific cases, that the respondents, used “Question” format of the sentence for communicating with the application. During the analysis of the derived results, we took into consideration this important finding, and classify the data on in each question in the used format of the respondents into four categories:

- Command
- Question
- Statement
- Exclamation

For each question, the most used verbs have been estimated and the collocation of the verbs and words has taken into consideration. Every question has its special task and characteristic, and for implementing the verbal commands on the application, we need to find the aliases of each word, verb, and in general the most used terminology for that specific task. That is the reason, the main approach for analyzing the derived results is the frequency of the vocabularies.

Moreover, by estimation of the frequency of the most used vocabularies, we can predict what is the most used verbs and words for the cloud word, and in general in human-computer interaction case. From the mentality of the human, which all the time seeing the machine and application as a tool for service, defiantly the most used verbs and words, have the commanding and ordering behavior. This application is a user-centered design application, and aimed to recognize, prioritizes the needs, preferences, and experiences of users.

3.4.5. Results

In total 66 responses were collected, which was near to the estimation of the people. Figure 3.13 Demonstrate the level of the participation in terms of gender. More than half of the respondents were Men. From the sampling bias perspective and the topic bias approach, probably the topic would more raise attention in the male gender rather than the female gender. Moreover, it is noteworthy that a poster was designed with a Qr code to ease the accessibility and caught the attention.

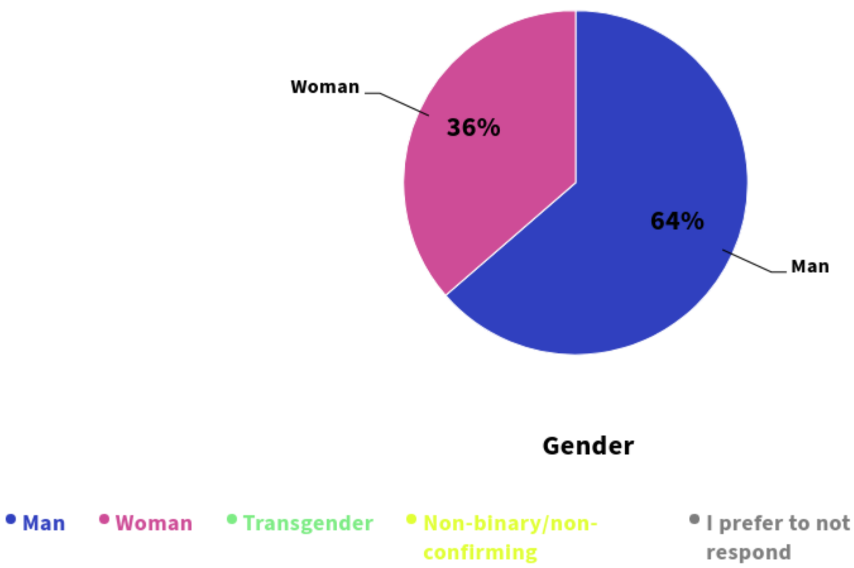

Figure 3.13: Gender distribution among the respondents

The next parameter as taken into consideration in the survey was age. In the figure 3.14 The ages have been classified into different age ranges. More than 65% of the respondents, were between age ranges of 18-30 and these results is due to the topic of this survey. Since implementing voice technology on a map seems strange and not useful for older generation, we can justify the lack of the contribution of older generation in this survey.

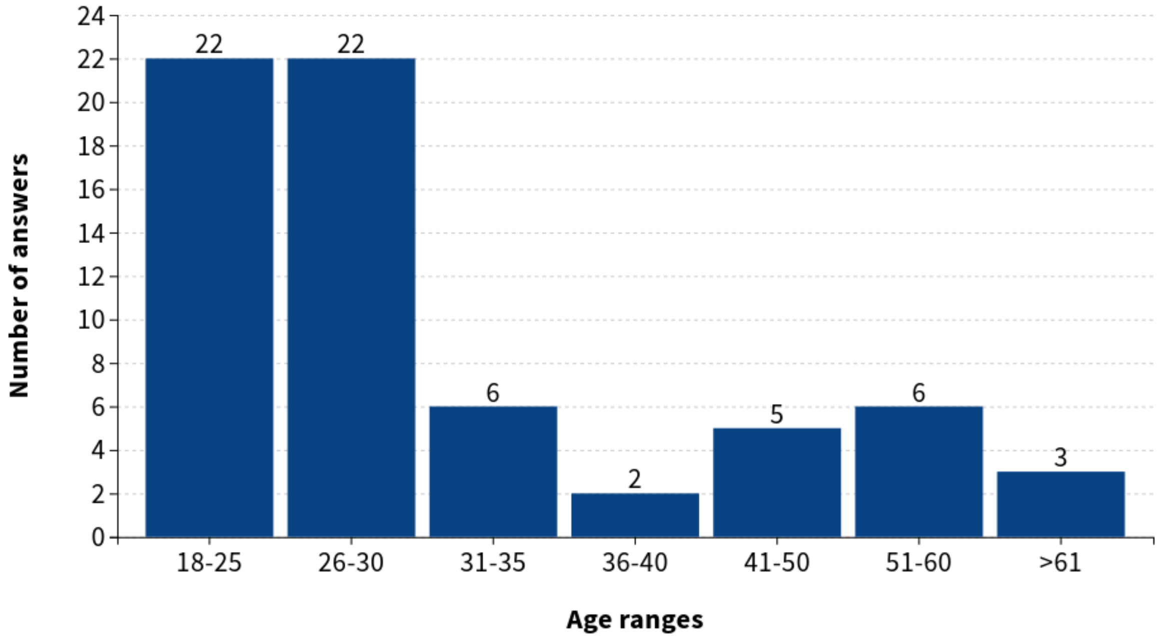

Figure 3.14: Age ranges among the respondents

The next two parameters were the level of the education and the field of their study if its applicable. As expected more than 58% of the respondents has higher level of the education in degree of Master and the rest of the participants have lower or higher academical degrees. The pie chart on figure 3.15 Demonstrate the level of the education among the respondents. Since the filed of the study was necessary to evaluate the domain knowledge of the respondents, we asked from the respondents to mention their field of study. Although, this was not a mandatory field, we received 50 responses from this question. The Graph 3.16 Visualize the field of the study of the respondents. Since the answers were from different fields and majors, the best approach to derive the relationship between the domain knowledge of the respondents and the topic was to categorize the field into their related general department or school. The groups are involving into: Civil, Environmental, and land management Engineering, Industrial and information technology, Information technology and computer science, and Not Engineering.

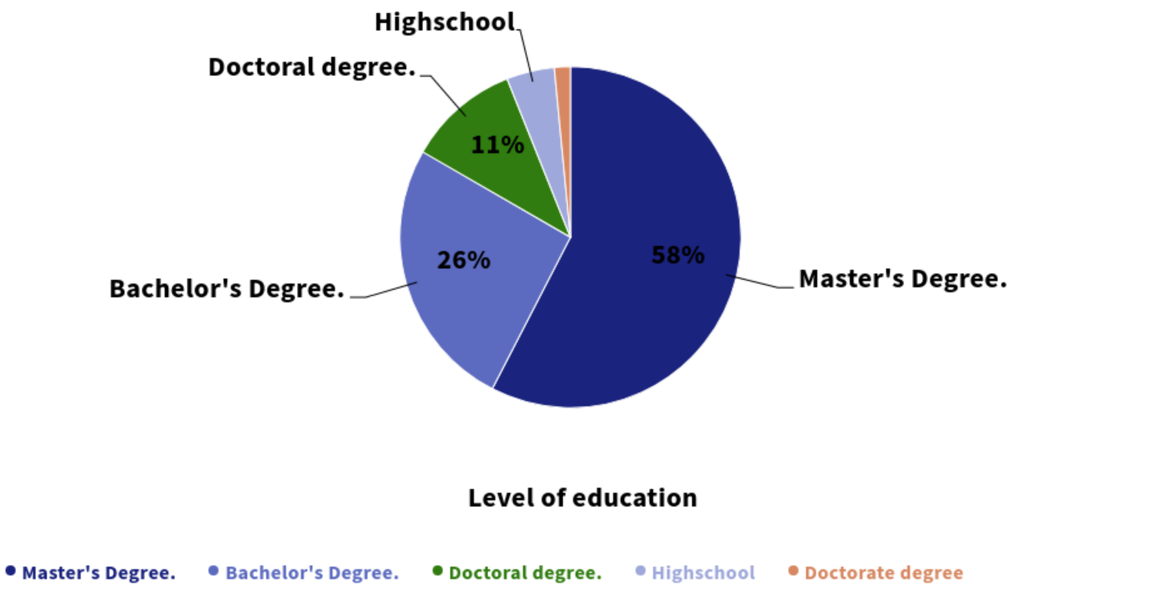

Figure 3.15: The level of the education among the respondents

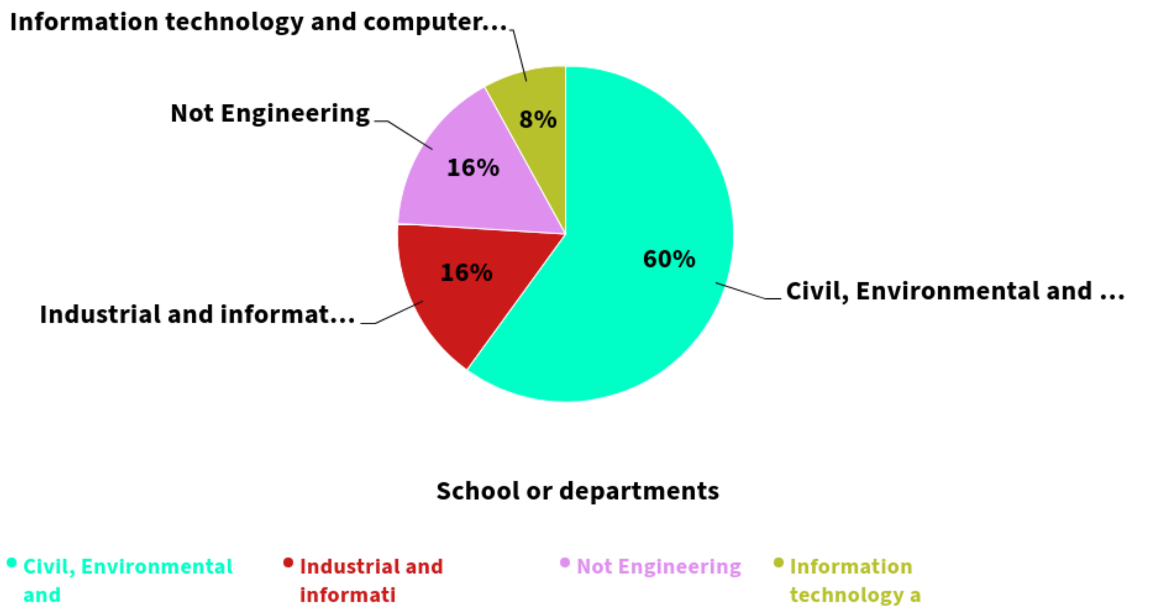

Figure 3.16: The school or department related to field of the study among the respondents

The last two important variables were the mother tongue and the level of the language proficiency. Since the survey distributed through channels which most of the users were Italian, so based on the prediction the biggest group of the respondents were Italian speakers. On the other hand, because of the ethnicity of the researcher who related to

Persian language, as the expectation the second largest of group of the speakers were Persian speakers. The figure 3.16 Visualized the distribution of other mother tongues among the respondents.
The level of the English proficiency was a very important factor, since as much as your knowledge in English vocabulary would be higher, as expectation the domain knowledge of the words and terminology that you must use would be much larger. Based on prediction, most of the respondents have more than advanced level of the proficiency, and no one had basic knowledge in contrary. The figure 3.17 Demonstrate the number of the respondents per level of the English proficiency.

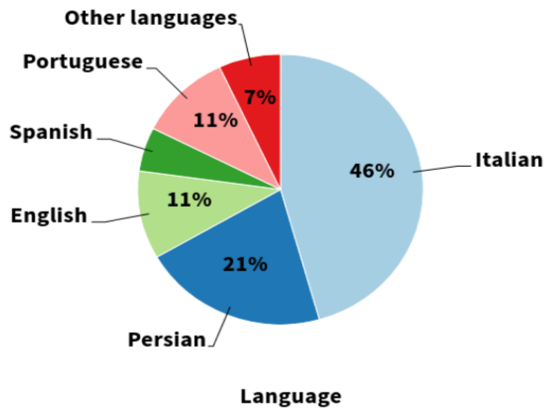

Figure 3.17: The mother tongue language among the respondents

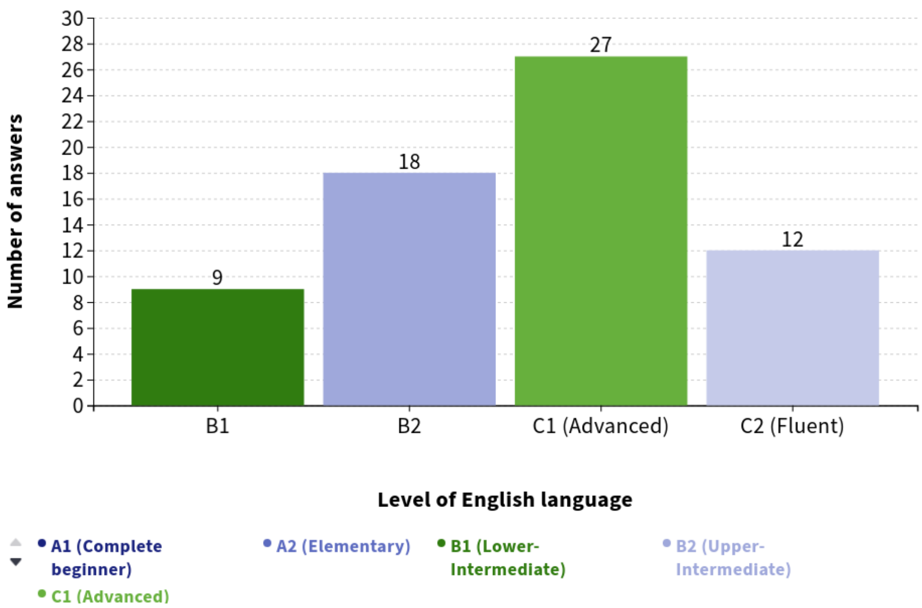

Figure 3.18: The level of the English language proficiency among the respondents

Type of sentence	Count
Statement	0
Command	52
Question	3
Exclamations	0
unacceptable =>Null	11
Total	66

Table 3.2: Type of sentences in Q1

Most used verbs	Count	Verb	collocations	Count
Show	28	Show	me	21
Find	7	Show	name of the location	3
Search	5	Show	The + name of the location	1
Go	7	Find	The + name of the location	3
Zoom to	6	Find	name of the location	4
		Go	to	7
		Zoom	to	4
		Zoom	in	2
			Egypt	4
			in Egypt	3
			of Egypt	1

Table 3.3: The most frequent verbs and words for Q1

In the following a brief statistic of the results of all the questions, that were on the survey has been discussed. Since the survey was categorized into two main section and each section has in order 5, and 7 questions, the following description is dedicated to unlocking the potential and aim of the questions and analyzing the results regarding the expectation.

Question 1: Find the Pyramid of Giza

This question was designed to understand what type of terminologies people used to navigate through different location on a map. The question was facilitated with a GIF which, was conveying the characteristic of the task to the users. The expectation was that the most frequent verb would be “find” and “search”, but the statistics shows that the most frequent verb for this task was “show”. Probably the reason behind of this popularity is that the content and also the topic of the task is related to visual contents and involvement of the users with the application, rely on the fact that after receiving a command, the application should react in a visual way to the users. The Table 3.2. And 3.3 Demonstrating the results of analysis from this question.

Question 2: Geographical direction vs Normal direction

This question aimed to analyses two objectives:

1. The terminology for doing the task panning on the map. There are synonyms for panning on the map, and similar vocabularies to convey the characteristic of the task “panning”. Therefore, we asked users with the help of the visual content embedded on the question to say their own commands to pan on the map.
2. The second goal was to understand in the context of the geospatial data visualization and talking with a map, users will rather use the geographical directions such as: North, South, West, etc., or the normal directions such as: Left, Right, Down, and Up.

As the expectation the respondents most, used verbs was “Go”, and surprisingly again the verb “Show”. The respondents tend to use the normal direction more than geographical direction, since the frequency of the word “left” was around 63 times. In contrast the word “west” just found 3 times in among all the answers. The reason probably is that from the normal direction are more understandable, practical, and visible in terms of using a Human-Computer Interaction. Another possibility is that in the question, the word “left” mentioned, in a way that asked from the user how they will approach and change the view to the left side. Therefore, the respondents used the exact same word which already provided on the question. Table 3.4 and 3.5 Present more information about the frequent words and verbs for this question. The response from this question has the highest type of the “Question” format among all the other questions, which mainly can be justified with the characteristic of the task.

We can also analyze if there is a relationship between the words used and the field of study of the participants. From the responses given, we can see that participants from different fields of study used similar commands. However, there were some variations in the specific words used by participants from different fields. For example, participants from Geomatics and Geographer and Systems Analyst used the word "move," while participants from Arts Studies/ English Lit used the word "pan."

From the responses given, we can see that participants who spoke different languages used similar commands. However, there were some variations in the specific words used by participants who spoke different languages. For example, participants who spoke Portuguese used the word "direct," while participants who spoke Italian used the word "explore."

Type of sentence	Count
Statement	0
Command	57
Question	6
Exclamations	0
unacceptable =>Null	3
Total	66

Table 3.4: Type of sentences in Q2

Most used verbs	Count	Verb	collocations	Count
Go	22	Go	to	14
Move	10	Go	left	7
Show	12	Show	me the left	8
Pan	4	Move	to	5
Zoom to	6	Move	left	5
left	5	west		3
		left		63

Table 3.5: The most frequent verbs and words for Q2

Question 3: Zoom in

This question was designed to derive the most used terminology for the task “Zoom in”. There were some hypotheses, that people can proportionally understand and declare the scale that they wish to zoom in or zoom out. And in sense of user experience and user need recognition, this question and next question were aimed to evaluate this hypothesis. The question asked from the user to change the view with a command to have the most possible zoom on a parking lot. However, majority of the respondents thought that finding the location and zooming on it is priority, not a zooming on an enormous scale. For this reason, the number of unaccepted answers for this case reached to 10. However, the verb “Zoom” and its collocation such as “zoom in” or “zoom to” were frequent among all the other words. The Table 3.6 And 3.7 Reveling information and data regarding this question.

In terms of age and gender, there doesn’t seem to be a significant difference in the use of these words. However, there are some variations in the responses based on the field of study. For example, respondents with a background in Geomatics, Geography, and Cartography used specific terms related to maps such as "drag," "center," and "tile." Respondents with a background in language-related fields used more phrases such as "oriental face" and "mão amolego," which may be related to their knowledge of foreign languages.

Type of sentence	Count
Statement	1
Command	51
Question	3
Exclamations	0
unacceptable =>Null	10
Total	65

Table 3.6: Type of sentences in Q3

Most used verbs	Count	Verb	collocations	Count
Zoom	28	Zoom	in	15
Show	10	Zoom	to	8
		Show	me	9
			the + location	34

Table 3.7: The most frequent verbs and words for Q3

Question 4: Zoom out

As mentioned on the previous description for the question 4, the aim of this question is to understand if the users, can proportionally mention an estimation during the zoom in or zoom out process. The question asked from the users, what will command if they wish to see more area around? With this purpose and the provide GIF, most of the users understood the tasks and number of unaccepted responses for this case was just 4. The noticeable point about the responses for this question is that, out of the 57 correct responses, just 5 of the respondents mention a proportion in different ways such as: percentage, level, km around the area, and fixed zoom scales. The rest of the respondents use the word "around" to set proportionally their desired view. Table 3.8 And 3.9 Visualizing information about the most frequent words and verbs in the responses of the question 4.

Type of sentence	Count
Statement	0
Command	57
Question	5
Exclamations	0
unacceptable =>Null	4
Total	66

Table 3.8: Type of sentences in Q4

Most used verbs	Count	Verb	collocations	Count
Zoom	45	Zoom	out	44
Show	7			

Table 3.9: The most frequent verbs and words for Q4

Question 5: Basemap or map

The term “Basemap” in geospatial science and cartography refers to a the base layer of a map, which can be in different format such as: Topography, Satellite imagery, or traffic. This expression is familiar with the people who are dealing with the geospatial data visualization and cartography. But for the users who do not have the professional or basic domain knowledge with the mentioned field, is not a common term. The aim of this question to examine people with providing the visual contents and explanation that showing the basic definition of a basemap as a basic layer, how they will refer this phenomenon in a natural language. The question asked from the respondents: “what you will say in order to see the traffic map of Cairo in day or night, or in satellite imagery?”. Expectedly, most of the respondents used the term “map” instead of the “basemap”. The ratio between the frequency of the map vs basemap is 30 to 8. Another discovery is that, since we could predict that people does not have any knowledge about the name of each basemap, we provided their names in the question. More than 68% of the respondents used the exact same name on the question. So, we can predict that to reduce the domain knowledge dependability, we must use a command to generalize the task for everyone. The Table 3.10 And 3.11 Present information for all the frequent words and verbs in the answers to question 5.

Women tend to use more words in their voice commands than men and are more likely to use longer phrases such as "switch to the night traffic map of Cairo" or "change the map base of the view of the Nile River near the pyramid of Giza". Men tend to use shorter phrases such as "show me traffic" or "show satellite".

The choice of words used in the voice commands seems to be influenced by the respondents' mother tongue to some extent. For example, those whose mother tongue is Portuguese are more likely to use the phrase "change basemap to night" or "show Traffic now", while those whose mother tongue is Persian tend to use phrases such as "switch to night mode" or "show me traffic at night/day".

Question 6: Color palette

This question assumed as one of the easiest questions, but the explanation provided might was challenging and distractive for the respondents and the result unexpectedly was not

Type of sentence	Count
Statement	0
Command	60
Question	4
Exclamations	0
unacceptable =>Null	2
Total	66

Table 3.10: Type of sentences in Q5

Most used verbs	Count	Verb/word	collocations	Count
Show	34	Show	me	22
Change	9	show	the	3
Switch	12	show	name of the map	8
		change	the	3
		change	to	4
		switch	to	8
		switch	on	1
		basemap		8
		map		30
		traffic	night	42
		traffic	day	41
		satellite		33
		background		1

Table 3.11: The most frequent verbs and words for Q5

Type of sentence	Count
Statement	0
Command	29
Question	2
Exclamations	0
unacceptable =>Null	35
Total	66

Table 3.12: Type of sentences in Q6

Most used verbs	Count	Verb/word	collocations	Count
Change	18	color ramp		4
show	5	color palette		4
colorize	2	color scale		4

Table 3.13: The most frequent verbs and words for Q6

good enough. The aim of the question was to evaluate what the most used verbs for is applying modification on the color palette of a thematic map. The verbs and the word color palette can be said in different synonyms. The explanation of the question described about the topic of the map, which was a COVID-19 total positive cases from year 2020 to 2021 in all Italy's provinces and asked how the users will command to the map to apply the modification visualized on the gif. 35 respondents did not understand the characteristic of the tasks and responded about the creation of the thematic map with the classes that already exists on the visual content (GIF), it means unfortunately more than 50% of the answers were not reliable and useful. The verb "change" was quite frequent among the responses, and significantly among the 31 correct answers there were a huge variety in the synonym words for the color palette. Color ramp, color palette, and color scale was frequent equally 4 times. Table 3.12 And 3.13 Provide some insights about the statistics of the frequent word and collocation in the question 6.
Respondents with a background in languages seem to use more specific and varied vocabulary when giving voice commands, such as "chloroplectic", "color palette", "color style", "color scale", and "visual change". This is particularly evident in respondents who spoke Portuguese, Persian, and Marathi.

Question 7: Query and filter quantitative data

The aim of this question was to understand the vocabularies for filtering the geospatial data. Is the methods that people filter is the same method as exists already on the BStreams platforms or not? For sure, the concept of filtering data is predominant to

Type of sentence	Count
Statement	1
Command	55
Question	3
Exclamations	0
unacceptable =>Null	8
Total	66

Table 3.14: Type of sentences in Q7

Most used verbs	Count	Verb/word	collocations	Count
show	28	show	me	20
Filter	7	with	more than	41
Highlight	7	over		4
		above		4
		provinces		41
		regions		3
		cities		2

Table 3.15: The most frequent verbs and words for Q7

the pre-knowledge regarding the data and the goal of filtering. Therefore, we tried to conceptualize this matter with a very well-known phenomenon (COVID-19) in such a way that respondents see the thematic of the total positive cases in the previous question and s/he going to asked to filter province which have total cases more than 1 million. With the provided hints on the questions, respondents use the same structure for filtering. "The provinces with more than 1 million cases" was frequent in 41 total counts. Another issue here is about the number, since Voice speech API transcript the number in a random way to integers, and integers + strings. Table 3.14 And 3.15 Demonstrate information regarding the frequent words and verbs in the answers from the question 7.

Question 8: Label the features

The goal of this question to determine the verbs for adding label and flags to features. Probably because of the survey fatigue, 45% of them did not paid attention to the contexts of the provided GIF on the survey and they just get inspired and copy from the question. But the derived verbs for those who understand the question is useful and acceptable. The respondent expect that the application keeps in mind their previous conversation, and in consequence do the actions regarding the history! Among all the responses, 32 of them were unacceptable and considered as null answers. However, the rest of the respondents provided good variety of synonyms verbs and vocabularies. The Table 3.16 And 3.17 Are

Type of sentence	Count
Statement	0
Command	32
Question	2
Exclamations	0
unacceptable =>Null	32
Total	66

Table 3.16: Type of sentences in Q8

Most used verbs	Count	Verb/word	collocations	Count
Highlight	8	Add	label	5
Indicate	7	Put	label	1
add	8	the		12
show	4			
Tag	1			
flag	3			

Table 3.17: The most frequent verbs and words for Q8

providing information regarding the statistics of the frequent words and verbs.

Question 9: legend customization

The cartographic customization is one of the main pillars of the application. The legend for a map describes the hidden narratives on a visualization and gives more information for analysis tasks. The aim of this task was to understand the terminology for applying the customization on the legend's position and also how many of the respondents will declare the position of the legend in order to reduce the complexity and reptation of the task. The most frequent verb was "move" which was repeated in the answers 20 times, while other respondents have the tendency to just mention 6 times the feature "legend" and declares its position afterwards. The Table 3.18 And 3.19 Describes the most used frequent verbs and words in the answers of the question 9 in statistical matters.
There does appear to be some correlation between field of study and the specific words used to give voice commands related to changing the position of the legend. For example, respondents with backgrounds in Geography/Geomatics/Cartography were more likely to use words like "change," "move," and "adjust," while respondents with backgrounds in Engineering were more likely to use words like "put" and "show."

Type of sentence	Count
Statement	0
Command	56
Question	2
Exclamations	0
unacceptable =>Null	8
Total	66

Table 3.18: Type of sentences in Q9

Most used verbs	Count	Verb/word	collocations	Count
Move	20	top	right	11
Show	6	top	left	12
Change	5	bottom	left	5
		bottom	right	5
		corner		16
		upper		6
		lower		1
		legend		23
		the	legend	33

Table 3.19: The most frequent verbs and words for Q9

Question 10: Markers color modification

As mentioned before there are two forms of geospatial data visualization available on BStreams:

1. Marker map
2. Graduated color (thematic map)

The last 4 question mainly was related to tasks specialized for the Graduated color map, but the last three questions on the survey was specifically designed to evaluate the terminology for the features and customization of the marker map. The terminology for the word “marker” is quite vast, and in more general purposes the customization for markers requires to be evaluated in terms of most used vocabularies. The verb “change” was the most frequent verb among the 52 correct responses, and majority of the responses was containing the word “marker” or “markers”. The evaluation focused on whether the respondents preferred to use familiar colors to control the application or if they were generally uncooperative. On the discoveries we found that 21 respondents mention the name of the color and in consequence the collocation “to” before the name of the color. The Table 3.20 And 3.21 Describes the analysis regarding the most frequent words and verbs on the answers to question 10.

Type of sentence	Count
Statement	0
Command	51
Question	1
Exclamations	0
unacceptable =>Null	14
Total	66

Table 3.20: Type of sentences in Q10

Most used verbs	Count	Verb/word	collocations	Count
Change	32	Markers		22
show	6	pin		5
		of		15
		to		20

Table 3.21: The most frequent verbs and words for Q10

In terms of the relationship between these voice commands and the participants' age, gender, field of study, level of English proficiency, and mother tongue, some patterns can be observed. For instance, younger participants tended to use more direct and simple commands such as "make markers yellow/red/blue" or "switch on markers, toggle color," while older participants used more complex language, such as "select all that match" or "indicate based on numbers."

Additionally, some differences can be observed between men and women, with women tending to use more descriptive language, such as "customize the marker color scheme for the population markers on the map," and men using more direct language, such as "markers red" or "use mark color <color>."

Regarding the participants' fields of study, those who studied Geomatics, Geography, and Geo Informatics tended to use more technical language, such as "marker color to red" or "change markers color according to your pollution (or other indicator)." Participants who studied Foreign Languages and Communication tended to use more descriptive language, such as "change the color of the pins to red" or "color the marker in red/blue/green etc."

Regarding the participants' level of English proficiency, those who had a lower level of proficiency tended to use simpler language, such as "change colors of the pins until I say 'Stop' to choose the preferred color," while those with a higher level of proficiency used more complex language, such as "indicate based on numbers."

Finally, regarding the participants' mother tongue, some differences can be observed.

Type of sentence	Count
Statement	0
Command	58
Question	1
Exclamations	0
unacceptable =>Null	7
Total	66

Table 3.22: Type of sentences in Q11

Participants who spoke Italian tended to use more direct language, such as "markers red" or "change the color of the markers." Participants who spoke Portuguese tended to use more descriptive language, such as "change marker color to red" or "change marker color to red/blue/green, etc." Participants who spoke Persian tended to use more technical language, such as "change markers to red/blue" or "customize the marker color scheme for the population markers on the map."

Question 11: Filter qualitative data for marker map

As the filter feature exists also for the graduated color map, it exits for the marker map. The filter for marker map, can be both cases qualitative and quantitative, however this requires the pre-knowledge about the dataset, and knowing all the fields from the dataset. But we could ease this scenario, as we did for the graduated color map with predefining some information about the topic of the marker map. For example, the marker map that we used for this section was a map visualizing the population of the whole provinces in Italy. For concept of the filtering, we proposed the question in a way that they have to only visualize some specific provinces (Liguria, Lombardia, Lazio) on the map. As an unexpected tradition verb "show" was frequent, 22 times among the respondents. "Show only Liguria,Lombardia, and Lazio" were repeated and seen a lot in the responses. Respondents in 10 cases uses the reverse engineering, in a way that they asked to remove/delete/cancel all the pins/markers except Liguria, Lazio, and Lombardia. Table 3.22 And 3.23 Visualize the most frequent words and verbs among the answers from the question 11.

Question 12: Size and format of the marker

The aim of this question to derive and evaluate the terminology that used specifically for the size and format of the marker. Size has a proportional concept, and for describing the size, and applying modification there are variety of possibilities in terms of synonyms.

Most used verbs	Count	Verb/word	collocations	Count
Show	22	Show	me	11
Keep	6	Show	only	11
Remove	7	Keep	only	4
Filter	5	marker		10
		pin		19
		Liguria		57
		Lombardia		57
		Lazio		57

Table 3.23: The most frequent verbs and words for Q11

Type of sentence	Count
Statement	0
Command	64
Question	2
Exclamations	0
unacceptable =>Null	2
Total	66

Table 3.24: Type of sentences in Q12

Therefore, this question was designed to asks the respondents how they can command to the application to apply the modification that they are seeing in the GIF. The respondents show the tendency to the verb "change", and as the consequence the verb "make" was less frequent. since this question was combination of two tasks and it was the last question, the respondents were not very keen to respond completely. The 23% of them did not respond to the question regarding the size customization. Table 3.24 And 3.25 Provides details regarding the frequent words and verbs in the answers to question 12.
Respondents with different levels of education used similar vocabulary and phrasing. How-

Most used verbs	Count	Verb/word	collocations	Count
change	43	change	the	15
Increase	7	markers		55
decrease	4	symbol		10
make	12	pin		7
		format		3
		type		5
		shape		6
		size		22
		flag		61

Table 3.25: The most frequent verbs and words for Q12

ever, those with degrees related to geography, cartography, or engineering used more technical language related to symbols and markers.
Respondents with different mother tongues used similar vocabulary and phrasing. However, those whose mother tongue was not English tended to use simpler language, and their commands were more straightforward.
Respondents with different English proficiency levels used similar vocabulary and phrasing. However, those with a lower level of English proficiency tended to use simpler language, and their commands were more straightforward.

3.4.6. Comparison of the results with ChatGPT

To gain a broad knowledge from the results that achieved form the survey, and access the correctness of the commands, the survey was simulated through *ChatGPT 4.0[1]*. ChatGPT is an advanced AI language model that trained by *OpenAI[4]*. The goal of this AI is to assist and communicate with the users through Natural Language. This feature of the ChatGPT makes it as a best source for a metric in our analysis.
The survey designed with the usage of the visual contents, in order to ease the understanding and comprehension for the respondents. Since the only available input on the ChatGPT is text format, so the questions on the survey were simulated with the texts. To keep the coherency with the predefined metrics such as age, gender, etc. the users defined as general users with the different domain language and with different level of the English language proficiency.
To analyze data, a large corps of voice commands collected based on the questions that were available in the questionnaire. As it mentioned before the users assumed as “general user” with different background knowledge. Each question transferred and trained based on a textual format and asked from the chatGPT. The results showed that the most common verbs used in the voice commands for map interaction were “show”, “change”, “locate”,” zoom”, ”find”, “move” and “filter”. These verbs were used with varying probabilities depending on the type of the interaction being performed. We also found that the probability of using certain verbs varied depending on the user’s background. The ChatGPT considered that based on the domain knowledge of the users the probability of usage for each command is different. For example, users with environmental engineering background, were more likely to uses the verb “locate”, while those with a background in computer science were more likely to use the verb “search”.
In addition to the most used verbs, we also analyzed the most common words used in the voice commands related to map interaction. These included words such as “north”, “south”, “left”, “right”, “marker”, “pin”, and “color palette”. The probabilities of using these

words varied depending on the type of the command given and also user's background. In the following a summery of all the answers in three main categories with their probabilities that driven by the ChatGPT has been included.

Movement and Navigation

The primary goal of the first five questions was to examine the most commonly used words and phrases that people employ when interacting with a voice-controlled map. These inquiries were designed to explore how users navigate through the map, including how they search for points of interest, move in different directions, adjust the zoom level, and modify the underlying map. The aim was to gain a better understanding of the commands and phrases that people use most frequently when interacting with this type of technology. All of the question has been converted into a plain format of text. An important aspect with ChatGPT is that it uses the words itself and paraphrase them to generate a response. To reduce giving hints to the AI to not use the exact same words and verbs the questions were simulated the situation of the tasks with a very plain grammar and words. The effort was to not use the exact same verbs that found already on the result of the survey, therefore keeping the balance with comprehension of the task with text and not using the already discovered terminologies has taken into consideration.
For the first five question, the ChatGPT simply correctly understood the questions, and provided similar answers to the results of the survey. The first question that asked after conveying the whole scenario and the type of the users was "Imagine users have a map and wish to command to map with voice to visualize the pyramid of Giza, tell me the possible commands with the probability of the answers in percentage". The answers and their percentage of their probability was very similar to all the answers that we collected from the respondents. The commands that it provided for this question was:

- "Show me where the pyramid of Giza is ": 30%
- "Find the pyramid of Giza": 20%
- "Zoom in on the pyramid of Giza": 15%
- "Locate the pyramid of Giza on the map": 10%
- "Center the map on the pyramid of Giza": 10%
- "Display the pyramid of Giza's location": 5%
- "Navigate to the pyramid of Giza": 5%
- "Pinpoint the pyramid of Giza on the map": 5%

As the expectation from the results that we collected from the responses for this question, the command that contains verb "show" has the more probability of usage and frequency. As Table 3.3 Reveals the frequency of the all the verbs, words, and their collocations, from the respondents there is a high similarity with all the commands that retrieved from the ChatGPT. Both datasets are written in a formal and informative style with clear sentence structures, and use technical vocabulary related to their respective topics, such as "supervised learning," "voice commands," "map," "probability," and "pyramid of Giza." However, the collected answer from the respondents uses more complex sentence structures, such as conditional and comparative clauses, while the answers from chatGPT uses simpler sentences with imperatives. Both datasets use verbs related to the topics, such as "retrieve," "find," "show," "zoom in," "locate," "center," "display," "navigate," and "pinpoint." Another important matter is that both datasets use the present tense for describing the topic.

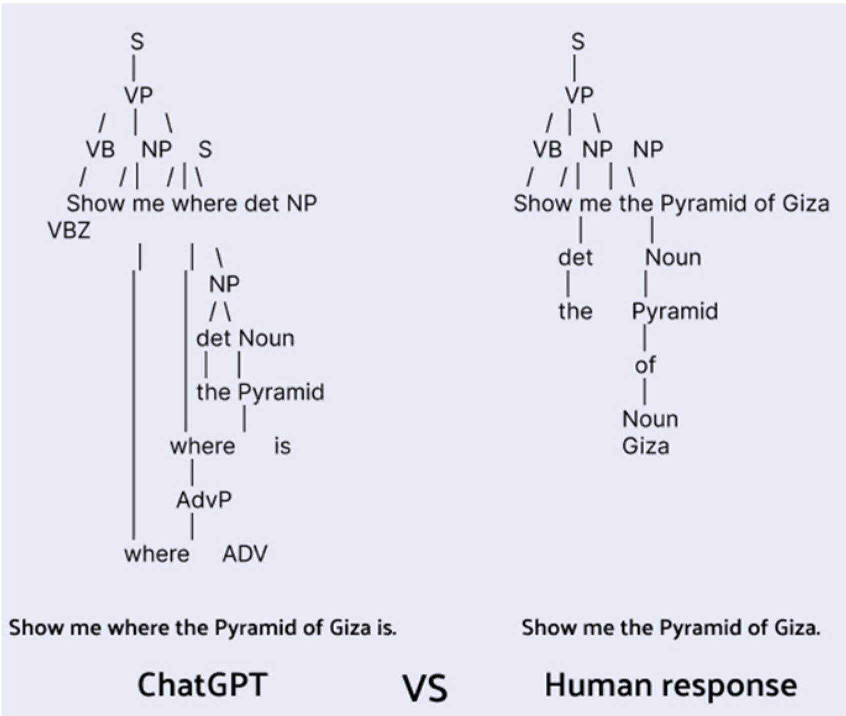

Figure 3.19: Comparison of the most used sentences from ChatGPT and Human Response in answers from Navigating question

The Figure 3.19 shows the parse tree of the sentence structure from two sentences that had a high probability of frequency on both datasets. In Human response parse tree, S represents the sentence, VP represents the verb phrase, VB represents the verb, NP represents the noun phrase, and det represents the determiner. In ChatGPT parse tree, S represents the sentence, VP represents the verb phrase, VB represents the verb, NP represents the noun phrase, det represents the determiner, VBZ represents the linking verb, and AdvP represents the adverb phrase. These parse trees show how the different components of the sentence are structured and how they relate to each other in terms of

their syntax.
In the first sentence "Show me the Pyramid of Giza", the VP consists of only one verb "Show", whereas in the second sentence "Show me where the Pyramid of Giza is", the VP consists of two verbs "Show" and "is", with a subordinate clause "where the Pyramid of Giza is" acting as the complement of the verb "Show". This difference in the complexity of the VP reflects the fact that the second sentence conveys more information since it generates by an AI. While the sentence “Show me the Pyramid of Giza” is generated by Human and shows that people use less complex and shorter sentences to communicate with a voice chatbot.
Second question that is being asked from the AI was “Imagine there is a map in front of you, that you can only interact with the map through voice commands. You wish to enlarge the area and see a location with more details. what are the possible commands you will say and how much is the probability (in percentage) of their usage with a normal user above 18 and have different domain knowledge?”. The provides answers was:

- "Zoom in" - This is a common command used in mapping applications to enlarge the area of the map. Probability: 30-40%
- "Enlarge the map" - A user might use this command if they are not familiar with the "zoom in" command or if they want to be more specific about what they want to do. Probability: 5-10%
- "Get a closer view of [location name]" - This command specifies the location the user wants to see in more detail. Probability: 10-20%
- "Increase the map scale" - This command may be used by users with more technical or analytical backgrounds who are used to working with maps and may prefer more precise language. Probability: 5-10%
- "Zoom into [location name]" - Like the first command, but more specific about the location the user wants to see in more detail. Probability: 10-20%

The Table 3.7 Is showing the results that we derived from the respondents. With a quick comparison with the answers that are collected from the ChatGPT, we can understand both datasets involve commands or instructions given to a machine or system, so they share a similar vocabulary. The verb "zoom" is common to both datasets, and it is used to indicate a closer look at something. The verb "enlarge" in the second command is similar in meaning to "zoom in". The verb "increase" in the fourth command is also similar in meaning to "enlarge" or "zoom in". Overall, both datasets share a similar vocabulary and purpose, but the second dataset includes more specific and complex language, which may make it more suitable for users with more technical or analytical backgrounds.
The third question has the opposite characteristic of the pervious question: “imagine you wish to see the pyramid of Giza with a smaller scale in order to see where it is located

for example with respect to the nearby cities, what are your commands and what are the probabilities of their usage in percentage?" and the provided answers from the ChatGPT were:

- "Zoom out" - This command can be used to decrease the scale of the map to show a larger area, allowing the user to see nearby cities as well as the location of the pyramid of Giza. Probability: 30-40%
- "Reduce the map scale" - Similar to "zoom out," this command may be preferred by users with more technical or analytical backgrounds. Probability: 5-10%
- "Show me the pyramid of Giza and nearby cities" - This command explicitly states the user's intent to see both the pyramid of Giza and nearby cities on the map. Probability: 10-20%
- "Display the pyramid of Giza in context" - This command may be used by users who are looking for a more high-level view of the area, with the pyramid of Giza as a focal point. Probability: 5-10%
- "Zoom out and show me the surrounding cities" - This command combines the "zoom out" command with the user's intent to see nearby cities. Probability: 10-20%

Investigating through the answers that provided by the respondents and shown on the Table 3.9 there is a common use of the verb "zoom" and its variations ("zoom in", "zoom into"), which is similar to the responses from survey where the verb "scroll" was used. Additionally, the verb "enlarge" is used in one command and it is similar in meaning to "zoom in". Both datasets use specific verbs related to the task at hand, but the previous dataset also included a wider variety of actions beyond just zooming or scrolling. Additionally, the previous dataset had more complex sentence structures and included more contextual information beyond just the command itself.

The next question aimed to understand what are the answers that ChatGPT will provide for proceeding in different directions. The question that asked was: "The voice map allows you to proceed in different directions. What will you say for instance if you want to explore the left area of the pyramids?". The answers that collected from AI were quite similar to our collected answers, however these are the commands:

- "Move left" or "Pan left": These commands are relatively straightforward and are likely to be used by many users. I would estimate the probability of these commands being used to be around 30-40%
- "Shift to the left": This command may be slightly less common than the previous two, but still relatively straightforward and understandable. I would estimate the probability of this command being used to be around 10-20%
- "Go towards the left" or "Explore the left side of the pyramids": These commands

use more natural language and may be preferred by some users but may also be less precise or efficient than the previous commands. I would estimate the probability of these commands being used to be around 20-30%

Analysis between the datasets from the responses that is shown on Table 3.5 and ChatGPT shows that there is one key difference is the level of specificity in the language used. "Move left" and "Pan left" are both very specific and direct, providing clear and concise instructions for the user to follow. "Shift to the left" is slightly less specific, but still relatively straightforward and easy to understand. In contrast, "Go towards the left" and "Explore the left side of the pyramids" are more general and open-ended, which may make them less effective in certain contexts. Another difference is the degree to which natural language is used. While all of the commands are relatively simple and straightforward, the latter two ("Go towards the left" and "Explore the left side of the pyramids") use more natural language and may be preferred by some users. However, as noted above, this may come at the expense of precision and efficiency. In terms of the structure and type of sentences used, all the commands are relatively short and simple, with clear subject-verb-object structures. There are no complex clauses or conditional statements, which may make them easier for users to understand and follow.

The last question for this section was related to the usage of different maps and how the AI can provide different voice command regarding the task and the percentage of the probability of the commands. The question was "You can see the traffic in Cairo during the night or the day on this map. You can also see Cairo's satellite imagery on the map. What would you say in order to see these? give me the probability of the answers with percentage?". These were the answers:

- "Show me the traffic in Cairo during the day/night": This command is relatively straightforward and specific. I would estimate the probability of this command being used to be around 40-50%.
- "Display Cairo's satellite imagery": This command is also relatively straightforward and specific. I would estimate the probability of this command being used to be around 30-40%.
- "Switch to daytime/nighttime view in Cairo": This command uses more natural language and may be preferred by some users but may also be less precise or efficient than the previous commands. I would estimate the probability of this command being used to be around 20-30%.
- "View traffic in Cairo" or "Show me the satellite imagery of Cairo": These commands are similar to the previous commands but use slightly different phrasing. I would estimate the probability of these commands being used to be around 10-20%.

Table 3.11 showing the frequency of the words and verbs that collected from the re-

Analysis	Human responses	ChatGPT responses
Average length of sentences	7 words	8 words
Use of natural language	Minimal	Moderate
Specificity of commands	Very specific	Mostly specific
Probability of command usage	Varies, highest 60%	Varies, highest 50%
Use of verbs	Limited	More frequent
Use of nouns	More frequent	Less frequent
Structure of sentences	Simple, mostly subject-verb-object	Simple to complex, more varied structure
Similarity between commands	Low to moderate	Moderate to high

Table 3.26: Summary of the result for the question regarding changing the basemap

spondents for the same question that asked from the ChatGPT. The Table 3.26 Gives a summary of the analysis that has been done between two datasets.

Choropleth map and customization

The next four questions on the questionnaire were designed to evaluate the understanding of the respondents from customization of the thematic maps. The results were shown that still the customization is a task that mainly related to the domain knowledge and the visual perspectives of the map and geospatial data.
The question that asked from the ChatGPT was mainly aimed to access the distribution of the voice commands regarding the changing color palette of a thematic map. The question was “imagine you have a thematic map and you can visualize it with different colors, that already implemented on the technology. what commands will you say to do so and what are the probability of the usage of the commands in percentage?”. The responses from the ChatGPT were:

- "Change the color of the map to [color name]": This command is straightforward and specific and allows the user to choose a color they prefer. I would estimate the probability of this command being used to be around 30-40%.
- "Show me the map with a different color scheme": This command uses more natural language and may be preferred by some users, but may also be less precise or efficient than the previous command. I would estimate the probability of this command being used to be around 20-30%.

- "Switch the color of the map to [color scheme name]": This command is similar to the first command but specifies a pre-defined color scheme rather than a single color. I would estimate the probability of this command being used to be around 20-30%.
- "Change the color of the [thematic feature] to [color]": This command allows the user to change the color of a specific feature on the map, rather than the entire map itself. I would estimate the probability of this command being used to be around 10-20%.
- "Customize the color of the map by [attribute]": This command allows the user to create a custom color scheme based on a specific attribute of the thematic map. I would estimate the probability of this command being used to be around 10-20%.

The answers that were collected from the respondents regarding this question were shown in Table 3.13 We can see that both datasets involve natural language commands related to a specific domain, which is map customization. However, the commands in the ChatGPT's answers are more specific and focused on changing the colors of the map, whereas the respondents' answers involved more general commands related to interacting with the map. In terms of the structure and type of sentences used in ChatGPT dataset, we can see that all the commands are imperative sentences, which is common in giving commands. Additionally, each command is composed of a verb phrase that specifies the action to be performed, followed by a prepositional phrase that specifies the target of the action (i.e., what to change and to what).

Another question regarding the faltering and querying the data asked from the ChatGPT. The question simulates the whole situation and asks with a case scenario the similar question that asked from the respondents on the questionnaire. The question was "Imagine with one command you can filter the map and show specific geometries, for example imagine you have a thematic map of covid in Italy on different provinces, and you wish to see the provinces with greater than 1 million cases. how you will command to the map to do so and what are the probability of the usage of the commands in percentage?"

These were the answers that are given by the ChatGPT:

- "Show only provinces with more than 1 million cases": 40%.
- "Filter map for provinces with over 1 million cases": 30%.
- "Display provinces with more than 1 million cases": 30%.

The respondents' thoughts and commands related to this question has been shown on Table 3.15 Both categories of the answers are related to querying data and filtering for provinces with more than 1 million cases. Therefore, it is not surprising that there are some similarities in terms of the words and verbs used in both datasets. For example, both datasets use the word "provinces" to refer to the regions of Italy being analyzed.

Both datasets also use the verb "filter" or variations of it, such as "highlight" or "display", to describe the action being taken to show only the provinces with more than one million cases. Additionally, both datasets use the phrase "more than 1 million" or "1 million total cases" to specify the threshold for the number of cases being analyzed.
The next question that asked from the ChatGPT aimed to understand how much of the voice commands that it can predict are similar to the answers that we collected for the legend's position. The question that asked from ChatGPT was “Imagine you have also a legend and you can customize the placement of the legend in different corners of the map. How will you command to the map to do so and what are the probability of the usage of the commands in percentage? you think people will also say the position or they just command without declaring the position?” and these were the answers:

- "Move the legend to the top left corner": 50%
- "Place the legend on the bottom right": 30%
- "Change the position of the legend to the top right": 20%

Table 3.19 shows all the frequent terminologies that has been collected from the responses, with the similar question. Obviously, respondent's dataset had more complex sentences with technical terms, whereas the second dataset contains simpler and more common words. The ChatGPT dataset has more commonly used verbs like "move," "place," and "change," which are easy to understand and interpret. So, in terms of similarity of words, the ChatGPT dataset is more similar to everyday language. The sentences in the second dataset are shorter and more straightforward compared to the first dataset. The second dataset has a clear subject-verb-object structure, which makes it easier to understand and process. The sentences in the first dataset had a more complex structure, with longer sentences and more technical jargon. So, in terms of sentence structure, the second dataset is simpler and easier to follow.
The sentences in the ChatGPT dataset are shorter and more straightforward compared to the respondent's dataset. The ChatGPT dataset has a clear subject-verb-object structure, which makes it easier to understand and process. The sentences in the respondent's dataset had a more complex structure, with longer sentences and more technical jargon. So, in terms of sentence structure, the ChatGPT dataset is simpler and easier to follow.
The Figure 3.20 describes the comparison between the probability of the similar words in both datasets related to three questions for Thematic map. For making the comparison between two dataset the answers from the three questions collected and then the percentage of the probability of frequency for both datasets calculated.

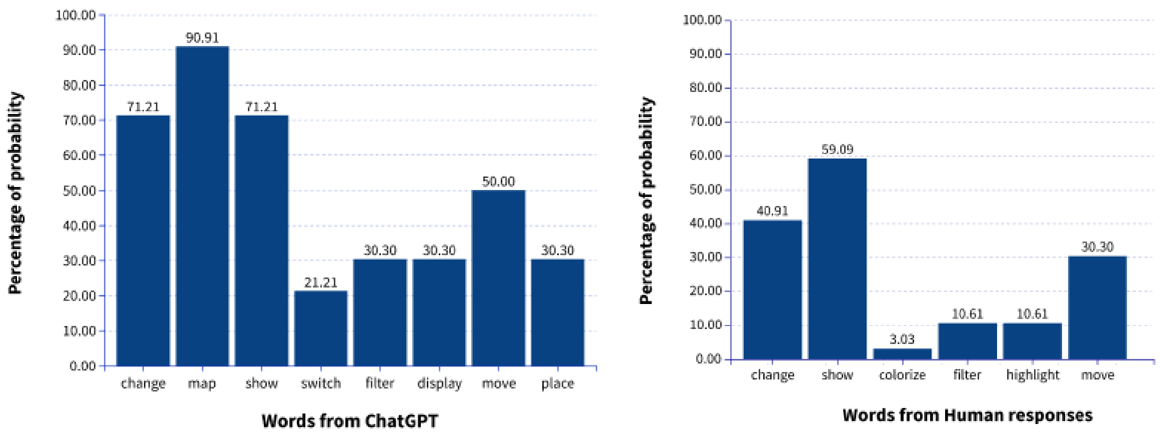

Figure 3.20: shows the percentage of the probability of the appearance for the frequent words that derived from both datasets.

Customization of the markers

the last three questions on the survey were specifically designed to evaluate the terminology for the features and customization of the marker map. We asked two main questions from the ChatGPT to examine what are the terminologies that it can predict for customization of the markers.

The first question that asked from the ChatGPT was related to color customization. The question was: “Imagine you have a marker map, and you can customize the markers into a different color. How will you command to the map to do so and what is the probability of the usage of the commands in percentage?” and these were the responses:

- "Change marker color to red" - 30%
- "Make markers blue" - 20%
- "Set marker color to green" - 15%
- "Switch marker color to yellow" - 10%
- "Color markers purple" - 8%
- "Change all markers to black" - 7%
- "Make markers orange" - 5%
- "Set marker color to white" - 3%
- "Switch markers to pink" - 2%

Table 3.21 shows a summary of all the frequent terminologies that has been collected from the responses, with the similar question. In terms of natural language analysis, the respondent’s dataset was more challenging due to its complex language structures and sentence variations, while the new dataset is simpler and more straightforward. Both datasets require the model to understand the meaning of specific words in the context of the sentence, such as "change," "color," and "marker," but the new dataset is more

focused on the similarity between colors and the action to be performed on markers. The ChatGPT dataset has a higher frequency of certain commands, such as changing marker color to red and making markers blue, indicating that these commands are more common or more important in the context of the application.
And the last question that asked from the ChatGPT was related to changing the shape of the makers. The question was: “Imagine you have a marker map, and you can customize the markers into a different shape. How will you command to the map to do so and what are the probability of the usage of the commands in percentage?” and these were the answers:

- "Change marker shape"
- "Switch marker style"
- "Choose different marker shape"
- "Set marker icon"
- The probabilities of these commands would depend on the specific user and their familiarity with map customization. However, based on general usage patterns, "Change marker shape" and "Choose different marker shape" are likely to be the most commonly used commands, with probabilities around 40-50%. "Switch marker style" and "Set marker icon" may have slightly lower probabilities around 20-30%.

The answers were collected from real respondents and the result shows on the Table 3.25. The comparison among the two collected answers of dataset shows that:

1. Both datasets contain similar phrases such as "set", "choose", "change", and "select".
2. The first dataset includes the phrase "add a marker", which is similar in meaning to "place a marker" in the second dataset.
3. In both datasets, there are references to "marker shape" and "marker icon".
4. The second dataset includes the phrase "switch marker style", which is similar in meaning to "change marker style" in the first dataset.
5. Both datasets use language that implies customization and personalization, such as "customize map markers" and "choose a style that suits you".
6. Both datasets mention the concept of familiarity with map customization, implying that the user’s experience level may impact their likelihood of using certain commands.

It’s clear that both are focused on map customization and provide users with options to change marker style, shape, and icon. The primary differences are in the specific phrases used to describe these options, and the probabilities assigned to each command. Figure 3.21 Visualizes the word cloud from the most frequent with high probability of the words that has been generated by the ChatGPT.

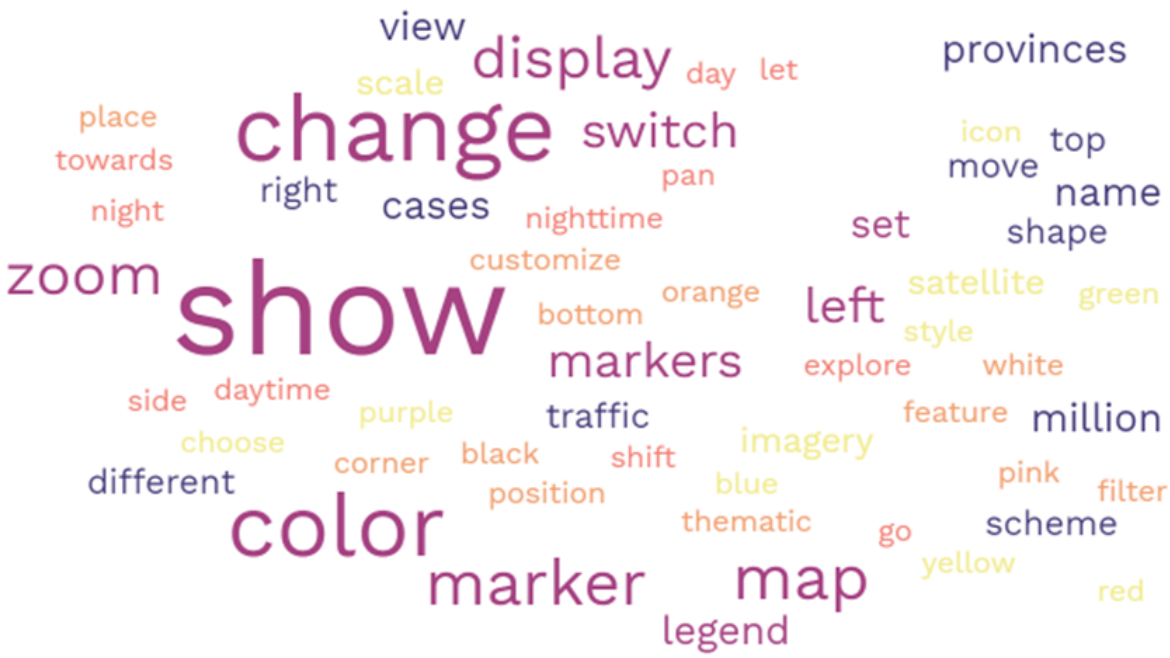

Figure 3.21: The word cloud from the frequent words that used by the generated commands from ChatGPT

3.4.7. Insights from analysis

The objective of collecting all the terminology regarded the geospatial data visualization and analysis was:

- Determine a collection of the words and vocabularies which are specialized for geospatial data visualization and similar tasks. Few studies have investigated this issue, and the results are inconclusive. Therefore, the need for having at least a structured database of the terminologies that particularly using for this geospatial data visualization and analysis, leads us to create the survey, collect the data, and produce the result as a readable and accessible format for everyone.
- The gathered verbs, words, structures, and vocabularies is used as a source in the application during the process of programming. These datasets of the terminology embedded into the code of the application to expand the accessibility of the users. Since the main pillar of the application is Web Speech API, and the main functionality of this library is to transcript the words and basically any signals that received by the microphone, so the application would be limited to only some specific commands that we already considered. And the application will react only if the user would say the exact same commands. By using the results from this survey, we expand the choice of the words and commands that implemented on the code of the

application and in this way, we can benefit users in different way such as:

- Improving the user experience: Synonyms and aliases can help users to interact with the application in a more natural way. The users can interact and talk with the application without the need of reading the instruction. This matter also helps to increase the flexibility of the application in such way that the user can do one task with multiple ways of commands. Moreover, it helps to reduce the risk of frustration and confusion.
- Variety of inputs: by tracking the user's inputs, there is the possibility to improve the performance of the application. Although the results from the survey can help to expand and predict the possible commands, but with tracking the unpredicted commands and implementing them on the code we can provide the basic infrastructure for an Artificial Intelligence application that is going to be trained and improved its skills by interacting with users.
- Accuracy of the words: with implementing the aliases and synonyms, the application's scalability can increase, in terms of understanding the user's intent and deliver more accurate response.

- The correlation between the metrics that we considered for analyzing the data, for all the cases were not very homogeneous. In some specific questions there were some obvious relationships between the frequent words and the metrics such as field of study, age, and mother tongue. However, for other metrics such as gender, level of English proficiency and level of education there were not intuitive relationships between the used frequent terminologies.
- One of the expectation results from this survey was to create a word cloud. The word cloud also known as the tag cloud, is a graphical representation of the data in format of text that are highlighted the most frequency words that used in a dataset. As much as the word would be frequent the bigger the word it will appears on the cloud. This type of data visualization, often used to present a quick summery of the most important or frequently used terminologies in a dataset. In data analysis, the word cloud is useful for determining quickly the most frequent words and understanding the trend or tendency of the users. For our scenario case, the word cloud could help us to quickly understand in context of the possible tasks that we defined for our application and put under verbal examination on the survey, which words are most used and most frequent in a natural language of our respondents. The Figure 3.12. Is word cloud that specifically visualize the frequency of the terms that we derived for the survey.

Figure 3.22: The word cloud derived from the results of the survey

3.5. Architecture

This section represents the details regarding the architecture of the application. The structure that is introduced bellow is generally used for mainly data visualization purpose by BStreams infrastructure. Since the motivation of this project as it mentioned on the state of the art chapter dedicated to evaluation of the open source speech text APIs for geospatial data, firstly a brief recap of the architecture of the interface itself presented, and the rest of this chapter explains the technical efforts for facilitating the BStreams application for the realizing the purpose of the project.
Gaining knowledge and presenting the backbone of the architecture, was necessary for implementation of the project. The chosen procedure for development should has the most coherency and compatibility with the interface, moreover the existing architecture of the BStreams piloted the process of the development and decision making.

3.5.1. BStreams

The BStreams is the product of the NextInt Company which is a consultant in the field of data science. They designed it based on their experience of their customers, and all the challenges that they had for the data visualization. The analysis and interpretation of complex graphs and charts was not an easy task for people. Regardless of the knowledge of the users who has to analyze the result, the user who has to create a report with different format of visualizations has to be an expert to understand which format of graphs, charts or maps is better to demonstrate the data in the most proper way. Therefore, NextInt founded BStreams in 2020, to facilitate the path of data visualization. Their motto is to ease the data visualization for users regardless of their knowledge. The platform has been designed based on data science field and it respects the roles of data visualization. The first version of application, launched at 2021. The last past year, BStreams has been facilitated with the Geospatial data visualization and other novel features, which increases the usability of the platform for its users. The effort was done by the cooperation of the development team and the author; therefore, the infrastructure is flexible for new development. The process which done by the author was as a form of internship. Since the platform recently has updated with geospatial data visualization, the company's team are always eager for further progress and development on their application and moreover, the author participated mainly for development of maps visualization on the interface, BStreams has been chosen as the platform for implementation of the project.

BStreams' architecture

The BStreams is a web application and the skeleton of it mainly is based on the *REST API[24]* architecture. A REST API is an application programming interface (API) which uses a Representational State Transfer (REST) architectural style uses *HTTP request[7]* to access and use data. Like any common web application, the architecture can be divided into main two side: Client side and Server side. The user agent defined as the client. It can be realized in common format, in any devices such as Personal computers, Laptops, Mobiles, Tablets and any other devices that has access to Internet. Basically, client side refers to a system which includes a browser that an agent has the possibility to interact and see the text, images, and the rest of the user interface within an application. On contrary the server side defines as complex structure which realizes any requests that done by the client side and provide the responses with help of the operations and programs that accumulated for purposes of the applications. In software engineering common terms is used for describing the mentioned architectures. The client side can be expressed also as a frontend, which means any component that manipulated by the user. The server side or the backend consists of the codes that agent has not access to it.

The skeleton of the architecture influences the connection between the mentioned side. And the used technologies, middleware, and framework make effect the network. As Figure 3.13. is describing the diagram of the architecture, each component defined in necessary detail that relates to the implementation of the project in bellow:

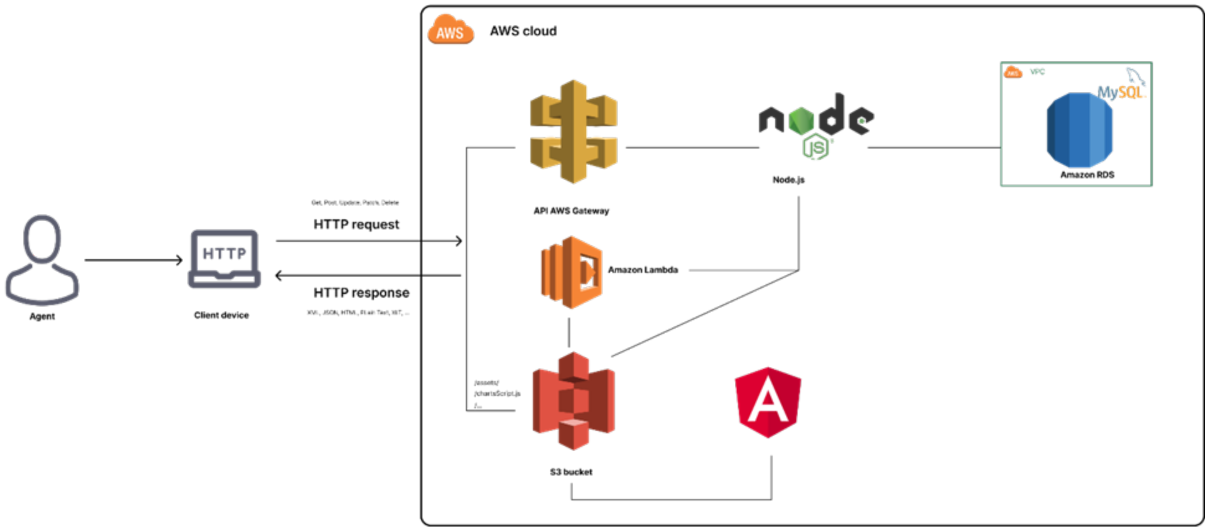

Figure 3.23: Architecture diagram of BStreams

- Node.js: is an open source and JavaScript runtime environment which runs on a JavaScript Engine and execute JavaScript code outside of a web browser. It is a scalable environment that can run on different platforms such as Windows, Linux, Unix, Mac OS X, etc. Node.js can generate dynamic page content, create, open, read, write, delete, and close files on the server. The compatibility of the Node.js with the implemented libraries, middleware and the frontend made it to a potential choice for programming the backend.
- Express.js: Since the architecture designed based on the REST API and Node.js, the most adopted backend web application framework is Express.js. Essentially, it introduced as the server framework for the Node.js. It eases the development of applications for web and mobile devices and in terms of placement, considered as a layer which built on top of the Node.js to enhance the management of the servers and routes.
- AWS service provider: It defined as a bridge between the different technologies, tools and databases so that integrate them seamlessly into a single system. It provides functionality to connect application brilliantly and adequately, till developers can fasten the process. In the architecture of the BStreams the middleware considered as the AWS service provider, which hosts the database, the service storage and Lambda function on the cloud. The benefits of using cloud computing are an undeniable fact in software engineering. BStreams using the middleware to access the cloud resources without being affected by the complexity of managing the infrastructures.
- PostgreSQL database: Is an open-source relational database management system

(RDBMS). A relational database manages data in one or more tables, where data may be related to each other. These relations help structure the data. SQL is a programming language used to create, modify, and extract data from relational databases. Since the middleware is based on AWS cloud services, BStreams' database is hosted by Amazon RDS. All information is stored in different tables that have relations with each other based on different fields. The tables are designed based on the main functionality of the application. Data from the interface is stored simultaneously in tables such as accounts, apps, charts, dataHubs, etc. Additionally, the application structure is based on loading default styles and customization for each chart, graph, or map from the database.

- S3 Bucket: Amazon S3 bucket storage service, is a Simple Storage Service (S3) which can store objects. The service provided by amazon, and it is adequate with other infrastructures that used on the architecture. It can store any type of object, that grants use like storage for application development, backups, data archives, data lakes for analytics, and hybrid cloud storage. In the case of the BStreams, S3 hosts the scripts, assets, shapefiles, fonts, icons, images, datasets, and etc.
- Lambda function: It is a feature from the AWS service for developers, which scale running of the scripts and functions only when it is needed. It can automatically leverage the number of requests that received by the application in an efficient way. BStreams uses the lambda function for asynchronous programming. Since its an ideal compute service for the data visualization and dealing with enormous size of the data, BStreams using it to improve the efficiency and productivity of the application for managing the load of the requests.
- Angular: Angular is a component-based framework for building scalable web applications, that developed by the Google. Angular facilitate a collection of well-developed libraries that covers a variety of features. Furthermore, it is a JavaScript framework built using Typescript, and uses for frontend developing. There are various versions of Angular, but the version which implemented on the BStreams is Angular 13. The selection of the Angular for frontend development framework was being lightweight for web applications, providing excellent materials and design libraries, efficient problem-solving patterns, and many other benefits that it has.
- D3.js: Grants the developers to bind the arbitrary data to a Document Object Model (DOM), and then apply data-driven transformation to the document. It emphasizes on web standards, which provide the full capabilities of HTML, CSS and SVG to demonstrate pretentious visualizations out of plain data. The instinct of the available charts and graphs on the BStreams has an origin from the D3.js library.

- Leaflet.js: As it mentioned BStreams has been recently added the geospatial (Map) visualizations to its interface. The Leaflet library is a lightweight JavaScript based, specialized for map and geospatial data features. It is open-source, adoptable with the used technologies and mobile devices, extensible with plugins, simple but high-performance library which majority of developers and application that handles with geospatial data visualization recommended it.

Visualization's process

In this section the efforts for creating data visualization within the application explained in a plain and brief language. The procedure is noteworthy since the design of the project mainly effected by the algorithm that implemented on the application. For the purpose of this project and consistency of the subject the scenario that explained here is dedicated to the geospatial data visualization.

Dataset retrieval

The process of the data visualization starts with data itself. The BStreams allows users to upload their own datasets in format of: xlsx, csv and links from Google sheets. Moreover, the application benefits the users by a public datahub, which updated daily. When it comes to geospatial data, the most main element of geospatial data which gives meaning to it, in form of BStreams application is, coordinates. The visualization of geospatial data is possible with datasets that contain latitude and longitude. The dataset can be richened by latitude and longitude if it includes any type of geocoding fields. Geocoding defined as the reverse engineering of retrieving coordinates from text-based description of a location, address, and sightseeing. In the process of developing geospatial visualization, this issue taken into consideration, and based on the efficiency and coherency of the whole architecture the Nominatim API chosen for the process of geocoding the datasets. Depends on the geocoding field that the user chose for retrieving the coordinates, also the number of the rows on the dataset the accuracy and the result can vary. The user will be notified by the notification center with the message about the finishing the geocoding process and if there would be null values for some rows, the interface provides a percentage of a retrieved coordinate which are not null, for the dataset.

Map creation

The BStreams enables the geospatial visualization in two possible formats of maps: Marker map and graduated color (Choropleth) map. Each of the mentioned formats

has their own specific features. BStreams is a platform which has its own theme, User Interface (UI) design, and functionalities for already developed visualizations formats. During the design process of maps visualization for the BStreams, the efforts done by the author to respect the consistency and coherency of the platform and consider the features which are not complex in a matter of interpretation and not in contrary with the existing template for the rest of the charts and graphs.
The process of creating a map, regardless of the format, starts from dragging the desired icons from the charts panel. The next step is to choose a dataset that the user wishes to visualize it in form of the map. The next step is customized based on the format of the map. In the BStreams architecture each visualization requires some specific fields from dataset to be created by the platform. These specific fields called Requirements, which defined on the entity (filed) of the charts on the database. The requirements of any charts are the necessary information that to be selected by the user on interface till the platform creates the visualization. Since there are two map formats available on the interface, the requirements filed of each of them mutually depends on the coordinate's fields on the dataset, and specifically for the graduated color map, a metric, and a vector layer of the polygons in format of maps defined as requirements. Technically each time that the user creates a visualization, the client side sends a request to the server side. From the beginning of the process that users select the format of the visualization, the response from the backend will send to the client side and provide the necessary information such as requirement fields and other attributes that specialized for each chart for the client side. Each specific chart has its own configuration and requirements which is designed on the database. Each specific chart has its own configuration and requirements which is designed on the database. Hence every time a user selects a chart, the request from the server side will provide a response from database with a JSON file that contains the necessary information for further processing and specialized customizations. Meanwhile user chooses a dataset and by selecting a dataset the necessary information which contains also the geofield entity will be rendered and cached from the server side on the client side. The cached dataset that contains the headers, id of the dataset, type of each column, filter cache, geofield, and other information. The subset is a virtual version of a dataset, and all the processing, filters and customization done on the subset not on the original dataset that has been uploaded by the users.

After selecting the dataset, the UI based on the response from the backend is changed to the setting specialized for that specific chart, since still the user is on the process of the "Edit" of a chart. The user selects the necessary field which are the requirements, and after that the Apply button will be enabled. The selected fields from the frontend will be collected and send to the backend, and the response will be a chart with all the provided fields. The chart will be stored on the charts table on the database, and configuration file will be the source of any customization that it will happen in the future. Any changes immediately reflect on the database and record all the information. The requirement field for the marker map are visualized in the Listing 3.2:

```
[
    {
        "coordinates": {
            "eq": 2
        },
        "popup": {
            "gte": 0,
            "lte": 3
        }
    }
]
```

Listing 3.2: Requirement field of the marker map

The requirement field for the graduated color map presented in Listing 3.3:

```
[
    {
        "coordinates": {
            "eq": 2
        },
        "metrics": {
            "eq": 1
        },
        "popup": {
            "gte": 0,
            "lte": 4
        }
    }
]
```

Listing 3.3: Requirement field of the graduated color map

The configuration file consists of property that has been defined based on the usability and instinct of each chart. In our scenario there are some properties that are mutual between both format of maps, such as: zoom control button, base map layers, filters, connect charts, header, and footers. But other properties are specialized for each format of the map. In a plain language the configuration file, explain the initial property and features for the visualization. With this engineering the load of the data for the client side reduced significantly, since most of the entities that are defined in the configuration file are customization regarding the visual perspective of the maps, such as: layout for markers, layout for the polygons of the graduated colors, customization of the header, etc. Therefore, by changing the layout feature and customizing the maps in terms of colors, size, and these types of features, the application receives the input from the users and just change the specific variable in the defined configuration file in the client side. The Listing 3.4 represent the configuration file in format of JSON for the graduated color map.

```
{
    "datavalues": {
        "popup": {
            "color": "rgba(0,0,0,100)",
            "fontSize": "M",
            "align": "center"
        },
        "cluster": {
            "show": 0,
            "size": 5
        }
    },
    "layout": {
        "marker": {
            "shape": "pin-basic",
            "size": 5,
            "color": "rgba(48,63,159,100)"
        },
        "mapOptions": {
            "showZoom": 0,
            "showLayers": 0,
            "defaultLayer": "OsmStandard"
        }
    },
```

```
    "header": {
        "show": 0,
        "align": "start",
        "titletext": "",
        "titlesize": "L",
        "titlecolor": "rgba(0,0,0,100)",
        "titleweight": "bold",
        "subtext": "",
        "subsize": "L",
        "subcolor": "rgba(0,0,0,100)",
        "subweight": "normal"
    },
    "help": {
        "show": 0,
        "text": ""
    },
    "footer": {
        "show": 0,
        "source": "",
        "url": "",
        "color": "rgba(0,0,0,100)",
        "size": "M",
        "weight": "normal"
    }
}
```

Listing 3.4: Requirement field of the marker map

3.5.2. Script

The engineering behind all the visualization in BStreams is based on creating the main script for each graph or charts as an extended class based on the main parent class which already exists. In JavaScript, the class extends keyword is used to create a subclass that inherits properties and methods from a parent class. This is part of object-oriented programming (OOP) and is called inheritance. The subclass is often called a child class, and the parent class is also known as the base class or superclass. When a class extends another class, the child class inherits all the properties and methods of the parent class.

The child class can also define its own properties and methods, and override methods from the parent class.
With this method with producing the new scripts for the new charts, graphs, and maps, they will inherit all the basic and predefined functions and features which defined for the parent class. The parent class in the architecture called, "skeleton.js". This script contains all the features and functions and their constraints, that enables the user to create a visualization.
After creation of the skeleton other class already extended as the second basic parent level for all the visualization. The logic behind the data visualization is to classify the visualization into structures which building the visualization and considered as the main components. For example, the axes can be shared in variety of charts, therefore, one specific class would be assigned to the axes and in this way the process of creation the charts and graphs would be much more comfortable and easier to follow and track. The main important class for the map creation, is the class chart or "chart.js". This class defining all of the functions and features that is available for any charts and graphs and containing the module of them from the D3.js library. Also, this script is responsible for retrieving data and distributing it to other classes in terms of defined inputs that receives from the user or from the frontend. Therefore, this class is an essential for the process of creation the maps.
The next two important scripts which are the main scripts of the map's creation are called:

- MarkermapChart.js
- grdauatedmapChart.js

the main functions from these two scripts have been briefly explained since there is the need to describe these components of the visualization in order to ease the path of the implementation and comprehension of the voice map application.
For the marker map which is being sourced from the script MarkermapChart.js, after retrieving data in terms of latitude and longitude from the dataset, and truncating them to the fifth decimal, the process of visualization and features enriching starts by "BASEMAP layers". All the available basemap layers is being sourced from the defined global variable which is containing all the basemaps with their predefined properties. The global variable called "defineMapLayers" as it shown on the Figure 3.14. Defined the properties and their sources.

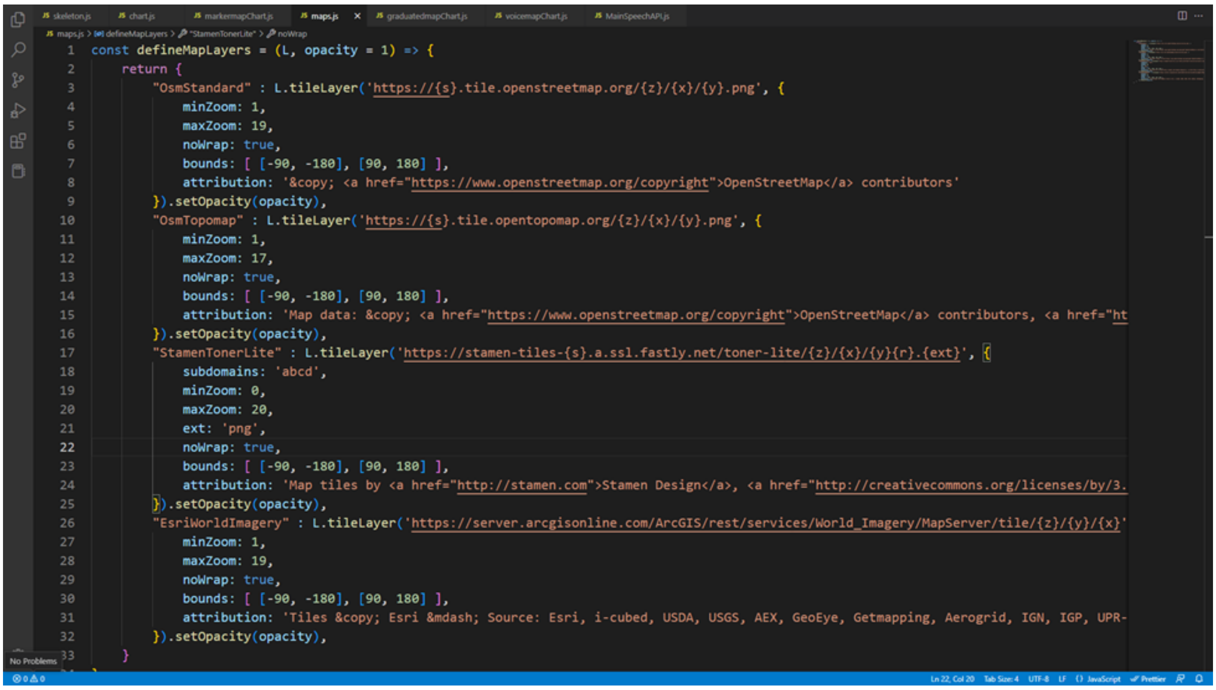

Figure 3.24: the defined base map layers

On the script of the marker map, the predefined layers has been called to provide the accessibility to the predefined basemap layers. The global variable Layers, receives the properties for each basemap layer and initialized it with most readable name that is available already on the UI. Every time a user creates the marker map, the script prepared all the available basemap layers and give a default value for the base map to the map. The Algorithm 5. Describes the functionality of the Layers on the script "markermapChart.js". One of the other important features is about the functions and variables regarding the marker's customization. All the markers on the visualization are in format of .SVG so it means they are editable in terms of size and color much easier. The makers are customizable in terms of format, color, and size. Initially all the possible formats of the markers in format of .SVG pre-loaded, on the script. To use this SVG code snippet in JavaScript, we create a new DOM element using document.createElement() and set its innerHTML property to the SVG code snippet. using single quotes (') to enclose the string and backslashes ("̈) to escape single quotes within the string. This is because the SVG code snippet itself uses double quotes (") to enclose its attribute values. After introducing the variable of the markers, the final steps for distributing the markers between data and giving default marker's size, format, and color continued. At the beginning of the scrip the selected data will be retrieved, and a subset of the data will be loaded on the client side. For this case, the only data in the subset would be latitude and longitude since they are the requirement for creation a marker map. The distribution of the coordinates through

markers, is conditioned to the point that, if the "cluster" property has been switched to "1", (generally is a setting that available after creation of the map) then with the help of the function cluster markers from the Leaflet changes the markers format to circles with predefined colors, opacity, and radius. If the cluster marker property "show" was "0" as it defined on the Algorithm 4. Then the script used the default marker "pin-basic" as the default marker with the provided size, and color.
Noteworthy that the cluster function is a predefined feature from leaflet which aggregated all the available markers on a map with a customizable radius. Every time the user changes the zoom scale on a map the cluster markers will change their distribution. The Figure 3.25 Is a schema of the cluster function and its related feature on the UI.

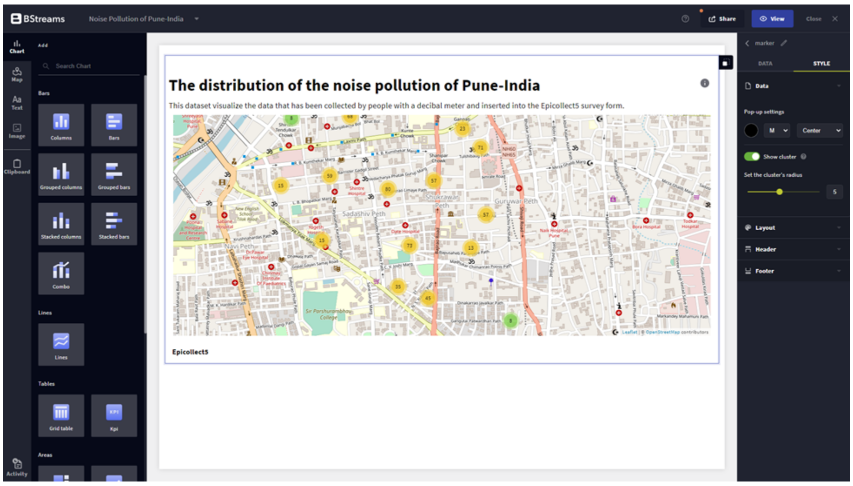

Figure 3.25: The cluster map of the noise pollution in Pune-India and the available feature on the BStreams

For graduatedcolormapChart.js which is the main script for the graduated color map, as it mentioned before regarding the requirement of this chart, "metric" is an essential, since it's a quantitative value or field of data that colorize the polygons and geometries, and in the end creates the final map. In the manual instruction for creation of a graduated color map, the users select the dataset, and choose the map that s/he wish to visualize. The map that user selecting here considered as a layer of the preloaded and light weight of .geo.json files. The moment that te user choose the desired map, the HTTP request will be sent as form of Get, and the application provided in form of HTTP response the desired map on the client side. Then the user must choose the corresponding fields for the latitude and longitude. And finally, s/he has to select a quantitative field as a metric. In order to reduce again the amount of the loaded data on the client side, the script already has been facilitated with a function from leaflet which convert the .geo.json to more light weight format of .topo.json. After retrieving the lightweight layer of the map, from the script,

the next step is to aggregate and calculate the metric to their corresponding geometries. The features and geometries already exists from the .topo.json file. The function checks if it contains any data points from the subset, if it does, it calculates the aggregate value of those data points and creates a centroid for that geometry. It also creates a subset of the data points, if specified. The region's name and ID are also stored. The bounds of the map are then calculated based on the geographic bounds of each region. Finally, the bounds are used to fit the map to the regions and markers, and the processed data is returned. The geometries which do not receive any data points, will be colored as black. The modification also will happen in case of filtering the data.
After retrieving the aggregated geometries and examining the boundaries and data points, the next step is to generate the color palette for associated dataset and geometries. The main function generatePalette receives 3 variables as inputs and return the colors variable. As the Algorithm 6. Showing the function, The function generatePalette() takes in a palette array, steps number, and a boolean shuffleColors as parameters. It returns an array of colors based on the palette and steps parameters. The function calculates the number of colors in the palette array and divides it by steps to determine the increment value (incr). If shuffleColors is true, it initializes the first color in the colors array as the color at the midpoint of incr. It then loops through the remaining colors in palette and adds them to the colors array at increments of incr until the colors array contains steps colors. If shuffleColors is false, it simply slices the first steps colors from the palette array and returns them in a new colors array. Finally, the function returns the colors array.The Listing 3.5 describes the functionality of the Layers on the script "markermapChart.js".

,

```
const Layers = defineMapLayers(L)
let map = L.map(`CD${this.container}`, {
    zoomControl: this.layout.mapOptions?.showZoom ?? 1,
})
map.addLayer(Layers[this.layout.mapOptions?.defaultLayer ?? "OsmStandard"])
if (this.layout.mapOptions?.showLayers ?? 1) {
    const baseMaps = {
        "Standard": Layers["OsmStandard"],
        "Topography": Layers["OsmTopomap"],
        "Toner Lite": Layers["StamenTonerLite"],
        "Satellite": Layers["EsriWorldImagery"],
    }
    L.control.layers(baseMaps).addTo(map)
}
```

Listing 3.5: Map Layers Code

The createSegments function generates a color palette and creates color segments based on the given method and data. It first calls the generatePalette function to create a color palette with a specified number of steps and whether to shuffle the colors. Then, depending on the chosen method, it either creates color segments based on quantiles or logarithmic distribution of the data. Finally, it creates segments by dividing the range of data into equal-sized intervals based on the chosen method and assigns a color from the palette to each interval. The resulting segments are returned as an array.

After producing the color palette and colorizing each geometry, the script checks the properties on the configuration file for the graduated color, for entity “legenda”. By default, this property is defined as truth property. It means that every time the user creates the map, the legend has to be appeared and added to the map layer. The provided code on the script, creates a legend on a Leaflet map, if the “legenda.show” property is truthy. It creates an object with a position determined by the “position” property or “bottomleft” as a default. The legend contains a color-coded scale based on segments (presumably generated elsewhere in the code) with numeric labels for each segment. It uses d3.formatPrefix to format the numbers in a readable way and adds appropriate labels based on the label property of each segment. The legend is then added to the map using the “addTo” method.

The point about the last two methods, is that they are synchronously connected. It means that if every time the user changed the method for creation of the segments, or the users changed the number of the steps, the legend automatically changed its value from the defined segments and colors from the parent function. Moreover, as it mentioned the position of the legend is already predefined into the upper and lower corner of the map. So, the user has four options to modify the position of the map. Other modification such as basemap layers, is as the same of the marker map.

Voice application scripts

Here in details the functions and codes that are involving and created the main pillars of the application, has been discussed. As it discussed in the section before, the whole architecture of the platform, is based on scripts that containing the necessary functions and APIs on the backend node server of the application. The important matter about implementing and embedding the application with voice assistant is to consider the coherency and already existing structure of the whole application and trying to write the code of the voice map application in a way to not raising conflict and overlap with other modules. As it mentioned on the state of the art, at the beginning of starting the whole process, the application was developing on a demo, for purpose of understanding and testing the compatibility, and adoptability of the voice speech API with a simulated interface which were like the BStreams. There were two main scripts for the demo application the main.js and the index.html. Basically, the structure of the demo is simple in form of software engineering since its was just as a matter of testing the compatibility of the application. The script Index.html contains all the HTML elements that needed to create the very simple frontend of the demo application. On the other hand, the main.js is containing all the functions in terms of the backend, controlling, and defining the actions. The elements and functions that has been implemented on the BStreams platform as the final architecture of the voice map application is very similar as the demo application. In the following, all the necessary components in terms of features and functions that are the main pillars of the voice map application is discussed.

MainSpeechAPI.js This script as other scripts available in the application programmed in the JavaScript language. It uses the Web speech API to recognize speech and perform certain actions based on the speech input. It defines variables for SpeachRecognition, SpeachGrammerList, and SpeechRecognitionEvent, and initializes other variables and sets default values, which are critical for the connection between the frontend and backend.
Basically, this script is the component of the voice application on the backend which by using the Web Speech API and it feature "SpeachRecognition" enables the whole structure to listen, transcript, and recognize the commands. This script is the bridge to connect the defined element on the frontend to the backend and control their functionalities. The following features are all of the available functions and important variables on:

- SpeechRecognition: This property enables the application to recognize speech inputs from the users. With speech recognition, users can interact with the web applications by speaking commands instead of typing and using the same traditional paradigm of the computers such as mouse and keyboards. Basically, this property is defining the initialization of the connection to the Web Speech API. For this purpose, the variables at the beginning of the code defined as it shown on the Figure 3.16. This way of initialization helps to the code to check that if the Web Speech API is available on the browser that users is interacting with the application or not.

```
var SpeechRecognition = SpeechRecognition || webkitSpeechRecognition
var SpeechGrammarList = SpeechGrammarList || window.webkitSpeechGrammarList
var SpeechRecognitionEvent = SpeechRecognitionEvent || webkitSpeechRecognitionEvent
```

Figure 3.26: Initialization of the Web speech API

- recognition events (onstart): Basically, the recognition is the property of the API that transcript the received signal and compares the transcript values by the library of the selected language. It compares the values and gives back the ones with accuracy of more than 99.99%. It follows by some events that are necessary to be mentioned. “onstart” event fires when the speech recognition has begun to listen for audio input. For our application this event only fires when users click on the button “start”. Moreover, as the sake of the other defined functionalities, this event set the Boolean variable of the “recognizing” to true. And also it sets the variable “resultindex” to 0, which will be used to keep the track of the current results of transcription.

- recognition events (onresult): When the recognition process detects a speech input, it fires the onresult event and passes an event object to the event handler function. The event object contains an array of SpeechRecognitionResult objects, each of which contains a sequence of SpeechRecognitionAlternative objects representing possible transcriptions of the spoken words. The code extracts the last result from the array using the resultIndex variable and calls the interpret() function, passing the transcribed speech text as a parameter.
- recognition events (onerror): this event triggered when an error occurs during speech recognition. The events in our case just logging the error message to the console, making it easier to debug and diagnose any issues.
- recognition events (onend): this event is triggered only when the speech recognirion has stopped listening, either due to the user stopping the speech input or due to occurrence of an error. In the code, it sets the recognizing variable to false, removes the "recognizing" class from the HTML document’s body element, changes the text of the startButtonElement back to "Activate Voice", and logs the event object to the console.
- recognition events (onspeechend): The recognition.onspeechend event is triggered when the speech recognition service has detected the end of the user’s speech. In the code, it stops the recognition service using the recognition.stop() method. This is useful to prevent the service from continuing to transcribe after the user has finished speaking and can help conserve resources on the client side. The Figure 3.27 Is showing the part of the code related to all the events that mentioned above.

```
//* RECOGNITION EVENTS
recognition.onstart = function(event) {
    recognizing = true
    resultIndex = 0
    document.body.classList.add("recognizing"); //adding the html class
    startButtonElement.innerHTML = "Listening"; //change the text of the button
    console.log('onstart ...')
}
recognition.onresult = event => {
    console.log('onresult ...', event)
    const lastEvent = event.results[resultIndex++]
    interpret(lastEvent[0]?.transcript.trim())
}
recognition.onerror = function(event) {
    console.log('onerror ...', event.error)
}
recognition.onend = function(event) {
    recognizing = false
    document.body.classList.remove("recognizing");
    startButtonElement.innerHTML = "Activate Voice";
    console.log('onend ...', event)
}
recognition.onspeechend = function() {
    recognition.stop();
}
```

Figure 3.27: The screenshot of the codes related to recognition events

- Chatbot: since this application enables the users to command to the application through voice, so the best way to take under control the communication is to design a chatbot. The chatbot that visualized the final transcript commands and providing feedback for the user. The chatbot structured wrote completely by pure HTML and CSS, and the messages inside of is controlling by DOM which is a technique in programming for retrieving the object and changing its characteristic by using "document." Method. The chat bot containing two types of messages as it mentioned before:
  - User response
  - Sys response

  The function for creating the chatbot is called logToBoxArea() which taskes two parameters: the snippet and the input. The purpose of this function is to log a chat message into an HTML element. The function first logs the snippet parameter to the console. Then, it checks the value of the input parameter to determine whether the message is from the user or the system. If input is equal to 'user', the function creates an HTML message string with the snippet parameter enclosed in a span element, along with some other HTML elements for the time stamp and username. If input is not equal to 'user', the function creates an HTML message string with the snippet parameter enclosed in a div element, along with some other HTML elements for the time stamp and system name. After creating the message string, the function appends it to the chatBoxArea HTML element and scrolls the chat-

BoxArea to the bottom to ensure that the most recent messages are always visible. The Figure 3.29. Visualize a screenshot of the code related to the chatbot and the logToBoxArea() function. The other important element inside of the chat bot is the button, which handles the process of the voice recognition. The Function startListening() is developed to pass the accessibility to the application through web speech API for listening and transcription. This function takes two arguments: an "event" object and an object containing "id" and "bmap" properties. The function starts by checking if a variable "recognizing" is true, and if so, it stops the "recognition" process and returns from the function. The Figure 3.28 Is visualizing the chatbot with the button activated for listening and resumed for listening.

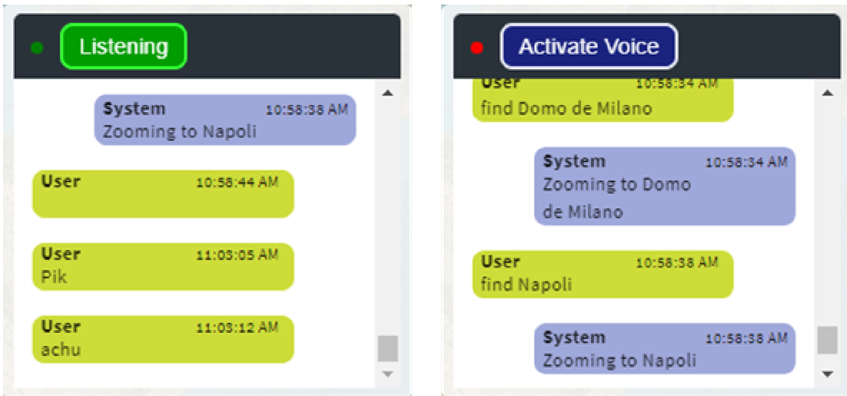

Figure 3.28: Schema of the chatbot and its content

If "recognizing" is false, the function sets a variable "rn" to the value of the "id" property from the second argument. It then checks if the "charts" object has a property with the key equal to "rn", and if not, it adds an empty object to "charts[rn]". The function then sets the "basemap" property of "charts[rn]" to the value of "bmap" if it is not already defined. After that, the function selects an HTML element with an ID of "VoiceButton" concatenated with "rn" and assigns it to a variable named "startButtonElement". It also selects an HTML element with an ID of "chat-history" concatenated with "rn" and assigns it to a variable named "chatBoxArea". Finally, the function starts the "recognition" process and logs a message to the console indicating that the function is listening for voice input. Figure 3.30 Shows the code of the startListening() function in details.

- Function interpret() : in the Web Speech API, a “snippet” typically refers to a segment of a speech or text that has been recognized or transcribed by the speech recognition engine. For example, if the user said “Hi, my name is John”, the Web Speech API might recognize this as the following transcript with two snippets:
  - “hi”

– “my name is John”

The function interpret() is the main function of this script, since as its name, it is responsible to interpret and comprehended the received command and do some actions, regarding to them. The function begins by logging the input snippet into the console and passing it to the function logToBoxArea() which described previously. It then proceeds to parse the snippet to determine which command to execute.

```
function startListening(event, {id, bmap}) {
    if (recognizing) {
        recognition.stop()
        return
    }
    console.log('listening ...')
    rn = id
    if (!(rn in charts)) charts[rn] = {}
    charts[rn]["basemap"] = charts[rn]["basemap"] ?? bmap
    startButtonElement = document.getElementById(`VoiceButton${rn}`);
    chatBoxArea = document.getElementById(`chat-history${rn}`);
    recognition.start()
}
```

Figure 3.29: The stratListening function handling with DOM the process of voice recognition in the chatbot

```
function logToBoxArea(snippet, input) {
    console.log('>>>> ', snippet)
    const msg = input == 'user'
    ? `
    <div class="chat-message clearfix">
        <div id="chat-message-content-user" class="clearfix">
           <span class="chat-time-user">${new Date().toLocaleTimeString()}</span>
          <h5 class="subtitle-chat-message-content-user">User</h5>
         <span id="final_span" class="final">${snippet}</span>
        </div>
    </div>`
    : `
    <div class="chat-message clearfix">
        <div id ="chat-message-content-sys" class="clearfix">
            <span class="chat-time-sys">${new Date().toLocaleTimeString()}</span>
            <h5 class="subtitle-chat-message-content-sys">System</h5>
            ${snippet}
        </div>
    </div>
    `
    chatBoxArea.innerHTML += msg
    chatBoxArea.scrollTop = chatBoxArea.scrollHeight - chatBoxArea.clientHeight
}
```

Figure 3.30: The screenshot of the code for the logToBoxArea()

The function first checks for the basemap commands, such as changing, removing, or showing the base map. If the snippet matches any of these referenced commands, the function will execute the appropriate action, such as changing the basemap, hiding it or showing it again. The references have been enriched with all the variables that are derived from the survey. Regular expression technique in JavaScript, which also known as the regex, is a very powerful tool for searching and manipulating texts. The “?” quantifier matches zero or one occurrence of the preceding element. The “

```
if (/(change|alter|modify)( the)? color of( the)? marker( to)?/i.test(snippet)) {
```

Figure 3.31: the example of the regular expression that uses in the interpret function

```
//* ZOOM TO LOCATION
if (/^(zoom|zuma|zumba|go|find) (to)?/i.test(snippet)) {
    console.log('> zoom to')
    const result = /^(?:zoom|zuma|zumba|go|find) (?:to )?(.+)$/i.exec(snippet)
    const location = result?.[1]
    console.log('location ....', location)
    logToBoxArea(`Zooming to ${location}`, 'system')
    const url = `https://api.mapbox.com/geocoding/v5/mapbox.places/${encodeURI(location)}.json?country=it&access_token=${token}`
    console.log('url ....', url)
    const newXHRRequest = new XMLHttpRequest()
    newXHRRequest.open( "GET", url, true)
    newXHRRequest.onload = function () {
        if (this.status === 200) {
            const response = JSON.parse(this.responseText);
            const feature = response?.features?.[0]
            console.log('--> ', response)
            const [lng, lat] = feature?.center ?? [null, null]
            let placeType
            if (/^(zoom|zuma|zumba)/i.test(snippet)) placeType = feature?.place_type?.[0] ?? null
            window[`panTo${rn}`]({lat, lng, placeType})
        } else {
            console.log('err...')
        }
    }
    newXHRRequest.send()
} else
```

Figure 3.32: special code for Geocoding function and finding the location with voice

The code includes the commands for managing the marker son the map. It can remove or show the clusters markers, as well as increase or decrease the radius of the cluster markers. The Figure 3.33. Shows the codes that are programmed inside of the interpret() function for the markers. The first condition checks if the user wants to remove markers or a cluster of markers. If the command matches the regex pattern “r̂emove (markers|cluster)(marker)?/i” , then the function removeCluster$rn is called to remove the markers.
The second condition checks if the user wants to add or show markers or a cluster of markers. If the command matches the regex pattern “(̂add|show) (markers|cluster)(marker)?/i”, then the function addCluster$rn is called to add the markers.
The third and fourth conditions check if the user wants to increase or decrease the radius of a cluster of markers. If the command matches the regex pattern “(̂increase|rise|expand)(the)?(cluster)? radius/i”, the function increaseClusterRadius$rn is called. If the command matches the regex pattern “(̂decrease|shrink)(the)?(cluster)? radius$/i”, the function decreaseClusterRadius$rn is called.

The fifth condition checks if the user wants to change the format, type, shape, style, or symbol of a marker. If the command matches the regex pattern "ˆ(change|alter|modify)(the)?(format|type|shape|style|symbol)?(of)?(the)? (marker|pin|pins)$/i", the function changeMarker$rn is called.
The last condition checks if the user wants to change the color of a marker. If the command matches the regex pattern "/(change|alter|modify)(the)? color of(the)? marker(to)?/i", the code extracts the color from the command using a regex pattern, and if the color is valid, the function changeMarkerColor$rn is called to change the color of the marker.
Then the code is checking if the user input snippet contains a specific command related to showing or hiding a legend in the map. If the command is to show the legend (e.g. "show the legend", "display legend"), then the function showLegend$rn will be called, which likely displays a legend on the map. If the command is to hide the legend (e.g. "hide the legend"), then the function hideLegend$rn will be called, which likely removes the legend from the map.

```
//* MARKERS
// Remove markers
if (/^remove (markers|cluster)( marker)?/i.test(snippet)) {
    logToBoxArea("Remove markers", 'system')
    window[`removeCluster${rn}`]()
} else
// Show markers
if (/^(add|show) (markers|cluster)( marker)?/i.test(snippet)) {
    logToBoxArea("Remove markers", 'system')
    window[`addCluster${rn}`]()
} else
if (/^(increase|rise|expand)( the)?( cluster)? radius/i.test(snippet)) {
    logToBoxArea("Increase cluster radius", 'system')
    window[`increaseClusterRadius${rn}`]()
} else
if (/^(decrease|shrink)( the)?( cluster)? radius$/i.test(snippet)) {
    logToBoxArea("Decrease cluster radius", 'system')
    window[`decreaseClusterRadius${rn}`]()
} else
if (/^(change|alter|modify)( the)?( format|type|shape|style|symbol)?( of)?( the)? (marker|pin|pins)$/i.test(snippet)) {
    logToBoxArea("Change marker symbol", 'system')
    window[`changeMarker${rn}`]()
} else
if (/(change|alter|modify)( the)? color of( the)? marker( to)?/i.test(snippet)) {
    const result = / (\w+)$/.exec(snippet)
    const color = result[1].toLowerCase()
    if (colors.indexOf(color) === -1) {
        logToBoxArea("INVALID COLOR", 'system')
        return
    }
    logToBoxArea(`Change marker color to ${color}`, 'system')
    window[`changeMarkerColor${rn}`](color)
} else
```

Figure 3.33: the codes for the commands related to the marker

voicemapChart.js This script is an extended code, for the skeleton of the voice map chart. As it described the general prospective of the maps on BStreams platform in the section script, this script must follow the same architecture of both maps. It means it must be extended as a child from the main parent skeleton.js and chart.js. Basically, this script containing all the functionalities and features for both the marker and graduated color map but programmed specifically for the voice interaction. The main functions and other important features are already as the same that it explained on the script section. To adopt the visualization with voice commands, the functions must be rewritten to accommodate the defined commands and callbacks in MainSpeechAPI.js. The adaptation process involves recognizing all existing objects in the code by initializing them as variables or using objects already defined in the map code. Some of the variables already exists on the code of both maps, but some of the new variables needed to be redefined and in order to not raise a conflict with other existing variables their names should be chosen wisely and uniquely. The following items, are explaining the most important structures and functions of the script voicemapChart.js which already is used as a callback in the other MainSpeechAPI.js. Noteworthy, that all the mentioned function here, assign themselves to the global properties of the "window" object. By declaring the window variable with global type, it ensures that the variable is compatible with both the browser and server-side environments, and it can be used to access global variables and functions in a consistent way:

- mapLayer: This function takes two parameters, "layer" and "oldlayer". It first checks if the "oldLayer" is defined using a conditional statement. If it is, it removes the corresponding layer from the map object using the removeLayer method of the Leaflet library, which takes the layer as an argument. Next, it adds the new layer to the map object using the addLayer method of the Leaflet library, which also takes the layer as an argument. This function is used to switch between different map layers on a Leaflet map.
- removeLayer: It takes one parameter, layer, and removes the corresponding layer from the map object using the removeLayer method of the Leaflet library.
- zoomIn: It takes no parameters and calls the zoomIn() method of the Leaflet library on the map object. This method increases the zoom level of the map by one step.
- zoomOut: also takes no parameters and calls the zoomOut() method of the Leaflet library on the map object. This method decreases the zoom level of the map by one step.
- panBy: This function takes one parameter, "direction", which is a string representing the direction to pan the map. The function uses a switch statement to check the value of direction and then calls the panBy method of the Leaflet library on the map object with an array representing the distance to pan the map in pixels along the x and y axes.
- panTo: it takes one object parameter containing lat, lng, and placeType properties. If lat is defined, the function calls either the flyTo or panTo method of the Leaflet

library on the map object with the specified latitude and longitude coordinates. If placeType is defined, it uses the placeZoom object to determine the zoom level to fly to, otherwise it pans to the specified location at the current zoom level.

- placeZoom: declares a constant variable named “placeZoom”, which is an object with keys representing different place type (e.g., “country”, “region”, “postcode”, etc) and values representing different zoom level to be used when the map is centered on a location of that type.
- removeClusters: This function takes no parameters and removes the leaflet marker cluster group from the map by calling the “remove” method on the “cluster” object.
- addCluster: Also, this function takes no parameters, and it adds the leaflet marker cluster to the map by calling the “addTo” method on the “cluster” object and passing in the map object.
- showLegend: This function takes no parameters and adds a Leaflet control object named legend to the map by calling the addTo method on the legend object and passing in the map object as an argument.
- hideLegend: On the opposite this function has the revers functionality of the previous function. it also takes no parameters and removes the legend control from the map by calling the remove method on the legend object.
- increaseClusterRadius: It is a function that can control and modify the cluster radius of the markers by voice. It takes no parameters and increases the maxClusterRadius variable by 20, removes the current cluster object from the map by calling the remove method, and then calls the makeMarkerCluster function to create a new cluster object with the updated maxClusterRadius value and adds it to the map.
- decreaseClusterRadius: This function takes no parameters and decreases the maxClusterRadius variable by 20 if it is greater than 20, removes the current cluster object from the map by calling the remove method, and then calls the makeMarkerCluster() function to create a new cluster object with the updated maxClusterRadius value and adds it to the map.
- changeMarker: function cycles through a predefined list of marker shapes and updates the marker.shape property of the map’s layout object with the next shape in the list. After updating the property, the function removes the existing marker cluster group from the map and generates a new marker cluster group with the updated markers.
- changeMarkerColor: function takes a color parameter and updates the marker.color property of the map’s layout object with the specified color. Similarly, the function removes the existing marker cluster group from the map and generates a new marker cluster group with the updated markers.

- filterAggregates(): This code is checking if a given snippet contains certain keywords to perform different actions. The first conditional statement checks if the snippet contains the keywords "filter" and "less/greater than" (ignoring case) and then extracts the method (less/greater) and value (a number or word representing a number). It removes commas and spaces from the value and replaces the word "million(s)" with "000000" if it exists. If the value is not a number, it checks if it exists in a variable that has predefined which called "mapNumbers" and sets the value accordingly. It then logs a message indicating the filter criteria and calls a function named "filterAggregates" with the region number and the method and value parameters.
- Resetfilters(): this functions checks if the snippet ends with the phrase "reset filters" (ignoring case) and logs a message indicating that the filters are being reset. It then calls a function named "resetFilters" with the region number parameter.
- Help commands: For letting the user to be acknowledge about all the xisting commands a help command also implemented. Users can say “help” or “commands” and a set of feedback with different categories will appears on the chatbot. The code is a function that takes a user input, snippet, and uses regular expressions to check if the input matches certain patterns related to help or commands for a map interface. If the input matches any of the patterns, the code will generate a response text, with a list of available commands related to the specific topic. Then, it will call a function, logToBoxArea(), with the response text and a string value representing the system log, which will display the list of available commands in a box or area on the map interface. Here are the topics and commands that are available based on the code:
  - help basemap: Displays a list of commands related to changing, hiding, or showing the basemap.
  - help movement: Displays a list of commands related to moving the map up, down, left, or right.
  - help zoom: Displays a list of commands related to zooming in or out, or going to a specific location on the map.
  - help cluster: Displays a list of commands related to showing or hiding markers or clusters, changing the shape or color of markers, and increasing or decreasing the cluster radius.
  - help regions: Displays a list of commands related to showing or hiding regions and increasing or decreasing the opacity of the map.
  - help filters: Displays a list of commands related to filtering data by values greater than a certain number, adding or removing regions from the filter, or resetting the filters.

– help segments: Displays a list of commands related to changing the number of steps, color palette, or legend visibility of the map.

Figure 3.34 shows the help function after using the command help on the application and related categories.

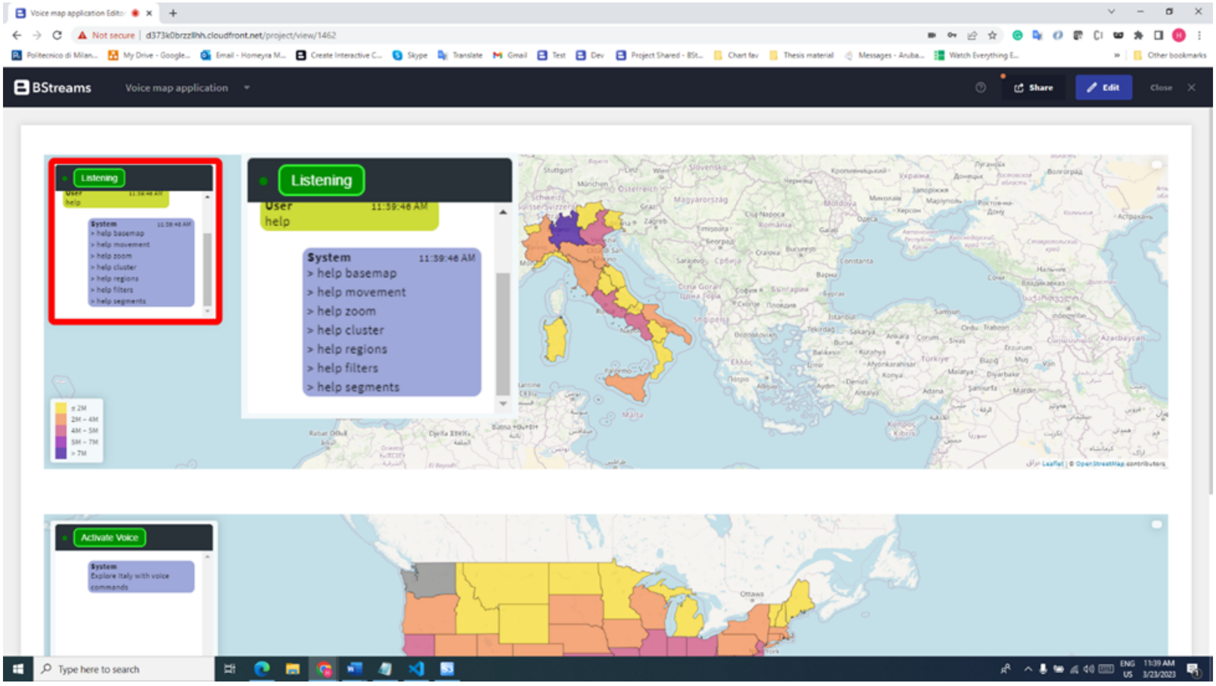

Figure 3.34: screenshot of the application with help command

In the appendix you can find the code for all the voice commands that is been used on the voicemapchart.js. As a matter of developing an open-source application all the codes related to the mentioned script is available on a Github repo. The final application is available on the platform BStreams, and everyone can access to it.

4 | Testing and debugging

The visualization and analysis of the geospatial data is becoming increasingly important and crucian in various domains. The use of a web-based geospatial application has become more prevalent due to ease of access and sharing of spatial data across and sharing of spatial data across different platforms. Additionally, the integration of voice command features in web-bases geospatial applications has the potential to enhance the user experience and improve accessibility for individuals.

The objective of this chapter is to evaluate the usability of the voice map application, that can visualize, process and analyses geospatial data and enable the users to navigate through different places with voice commands. The application included also a chatbot which acts as a virtual assistant and enables the user to interact with the application and receives feedback. As its already discussed in the previous chapter, the whole structure of this application is implemented already on the BStreams platform, and its available to public.

Previous research has shown that the use of voice command features in web-based applications can improve the efficiency and effectiveness of user interactions [56],[12]. Moreover, the use of chatbots has gained popularity due to its ability to improve the user experience by providing personalized and immediate feedback[31]. However, there is limited research on the usability of web-based geospatial applications that integrate voice command features and chatbots.

To address this gap, this study aims to evaluate the usability of a web-based geospatial application that includes voice command features and a chatbot. The study will focus on evaluating the effectiveness, efficiency, and satisfaction of the application through user testing and feedback.

The purpose of the testing was to evaluate the usability of a web application that enabled geospatial data visualization with voice commands. The testing aimed to assess the navigation and satisfaction of users with voice virtual assistants in the context of geospatial data, identify usability issues and opportunities for improvement in the application, and gather feedback on the users' experience of using the application with voice commands. The testing involved 10 participants with experience in using web applications and geospatial data visualization tools, who were diverse in terms of age, gender, and profession. The testing was conducted remotely using video conferencing tools, and participants were asked to think aloud while performing the assigned tasks. Completion rate, task time, error rate, user satisfaction, age, field of study, gender, and mother tongue were evaluated as metrics to analyze the data gathered from the testing. The results were analyzed to identify common issues and opportunities for improvement in the application, and a report was prepared with recommendations for enhancing the application's usability. The following are the details of the usability test is mentioned:

4.1. Usability test

4.1.1. Objectives

- To evaluate the usability of the web application that enables geospatial data visualization with voice commands.
- To evaluate the navigation, and satisfaction of the users regarding the voice virtual assistants in context of the geospatial data.
- To identify usability issues and opportunities for improvement in the application
- To gather feedback on the users' experience of using the application with voice commands

4.1.2. Participants

- 5-10 participants who have some experience in using web applications and geospatial data visualization tools
- Participants should be diverse in terms of age, gender, and profession Materials:
  - Computer with internet access
  - The link of the application on the web to be tested
  - Instructions for tasks which reads by the supervisor (it is accessible to the user in format of the PDF at the beginning of the test)

4.1.3. Tasks

- Go to the link of the application.
- Asking for help from the application by saying the command help.
- Navigate through different places and let users choose their own words/verbs to find and explore different spots.
- Asking user to explore the area and changing the view of the map and moving to different directions.

- Display different layers of the map and change the base map.
- Switch to the marker map.
- Change the color and shape of the marker.
- Increase/decrease the cluster radius.
- Remove the marker map.
- Switch to the graduated color layer of the map.
- Control the opacity of the map.
- Control the appearance of the legend.
- Change the color palette.
- Filter the regions.
- Add/Remove the regions.
- Reset the filters.
- Exit the application

4.1.4. Methodology

- The test will be conducted remotely using a video conferencing tool.
- The test will be conducted one-on-one with each participant.
- Participants will be asked to think aloud while performing the tasks.
- The supervisor observes the participants and take notes from their comments.
- The supervisor should observe the emotions and feelings of the participants, and register the success of each task, and level of the contribution of the user.
-

4.1.5. Metrics

- Completion rate: the percentage of participants who successfully completed each task
- Task time: the time taken by participants to complete each task
- Error rate: the number of errors made by participants while performing each task
- User satisfaction: feedback gathered from participants through a post-test questionnaire
- Age: the age of the users might affect their efforts to correct their errors and being insistent to try again the voice commands
- Field of study: the diversity of the domain knowledge effects the interaction between the user and the application
- Gender: This is a general metric to access the other existing metrics and finding some correlation

- Mother tongue: Usage of this metric enable us to apply some analysis regarding the accent of the users.

4.1.6. Data analysis

- Task completion rates, task time, and error rates will be calculated and analyzed for each task.
- Feedback from the post-test questionnaire will be analyzed to identify common issues and opportunities for improvement in the application.
- A report will be prepared summarizing the results of the test and outlining recommendations for improving the application's usability.

4.2. Results and analysis

The study involved 10 participants with diverse backgrounds and experience levels in web applications and geospatial data visualization tools. The participants were asked to complete 18 tasks, and their task completion rates, task time, and error rates were measured, along with their feedback on the usability of the application. The data analysis aimed to evaluate the effectiveness of the methodology used, as well as to identify the common issues and opportunities for improvement in the application. The Table 4.1. Shows the results that driven from testing of the voice map application.

- Task Completion Rate: The overall task completion rate was 91%, with a range of 75% to 100%. Participant 10 had the lowest task completion rate at 75%. This metric indicates the effectiveness of the application in enabling users to complete tasks using voice commands. A 91% completion rate suggests that the application is generally usable, but improvements may be required to ensure that all users can complete tasks successfully.
- Task Time: The average task time was 14.2 minutes, with a range of 9 to 22 minutes. Participant 8 had the fastest completion time at 9 minutes, while participant 10 had the slowest completion time at 22 minutes. This metric indicates the efficiency of the web application in enabling users to complete tasks using voice commands. The average task time suggests that the application is reasonably efficient, but improvements may be required to reduce the completion time for some users.

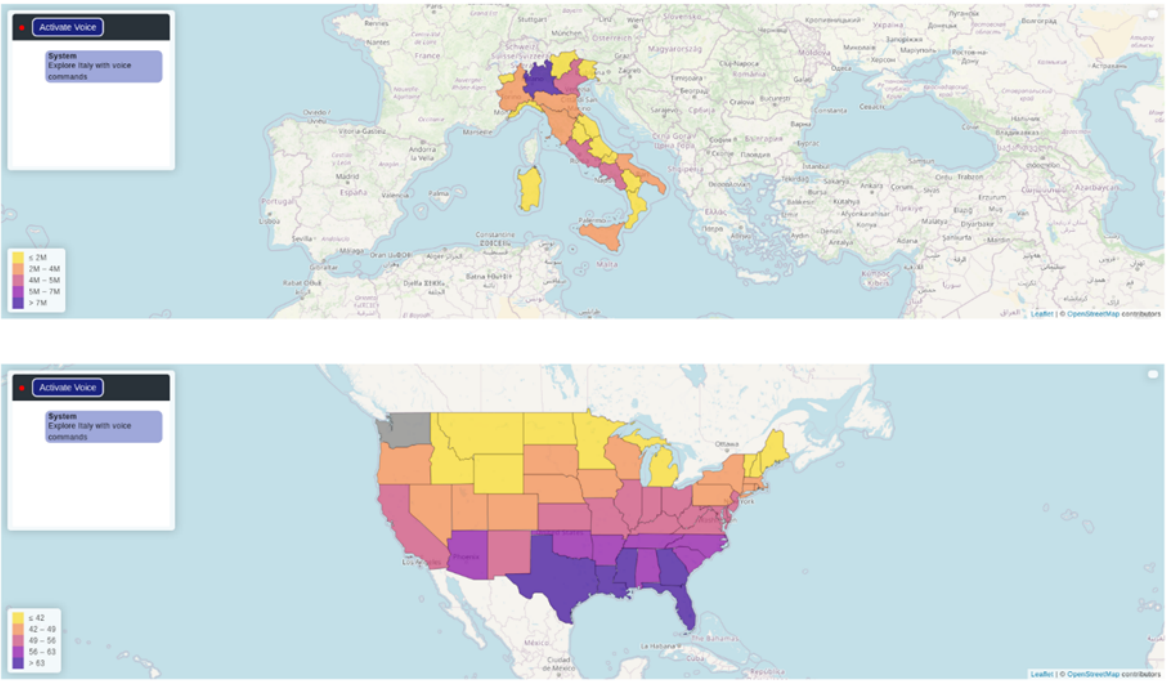

Figure 4.1: Screenshot of the voice map application that tested by the users with two already build map from Italy and United States

- Error Rate: The overall error rate was 3.8, with a range of 2 to 5 errors per participant. Participant 2 had the lowest error rate at 2, while participants 3, 6, and 8 had the highest error rate at 5. This metric indicates the accuracy of the web application in interpreting voice commands from users. A lower error rate suggests that the application is more accurate in interpreting voice commands.
- User Satisfaction Score: The overall user satisfaction score was 7.8 out of 10, with a range of 6 to 9. Participants 3 and 9 had the highest user satisfaction score at 9, while participants 5 and 10 had the lowest user satisfaction score at 6. This metric indicates the level of user satisfaction with the web application in terms of usability, ease of use, and user experience. An average score of 7.8 suggests that the application is reasonably satisfactory, but improvements may be required to increase user satisfaction.

Overall, the data suggests that the web application that enables geospatial data visualization with voice commands is generally usable, efficient, and accurate. However, improvements may be required to ensure that all users can complete tasks successfully, reduce the completion time for some users, and increase user satisfaction. Based on the feedback gathered, recommendations could be made to improve the application's usability, including improvements to the voice recognition software, clearer instructions for users, and

User	Age	Gender	Mother tongue	Field of Study	Task Completion Rate (%)	Task Time (minutes)	Error Rate	User Satisfaction Score (out of 10)
1	25	Female	Persian	Geomatics Eng.	100	10	4	8
2	30	Male	Persian	Computer Science	90	15	2	7
3	22	Male	Italian	Computer Science	100	12	5	9
4	28	Female	Italian	Environmental Science	95	18	3	8
5	37	Male	Italian	Urban Planning	85	20	4	6
6	20	Male	Italian	Mechanical Eng.	100	11	5	9
7	26	Male	Italian	Data science	90	14	4	7
8	24	Male	Italian	Data science	100	9	3	8
9	31	Female	Persian	Architecture and Landscape	95	16	4	9
10	23	Male	Italian	Law	75	22	2	6

Table 4.1: Result of the testing based on the metrics

better feedback for users regarding the application's response to their voice commands.
The Table 3. In the appendix summarized all the detail of the completion of the tasks by the users. We can start by analyzing the completion rate of each task. The data shows that all participants completed the first task, which was to go to the link of the application. The completion rate was also high for tasks 2, 3, 5, 6, 9, 12, 13, 14, 17, and 18, with all participants completing these tasks. Tasks 4, 7, 8, 10, 11, 15, and 16 had a lower completion rate, with some participants not completing them.
Going into details 50% of the users found task number 8 hard and unnecessary. The words were sophisticated and hard to memorize, for commanding to the system. Moreover, than that, while users were performing the command "Increase the cluster radius", they did not find any differences after implementation of the task.
Tasks number 15 and 16 had a low completion rate with respect to other tasks. In general, 30% of the respondents could not complete this task or they required more time to complete the tasks. Mainly the users were unhappy about these two tasks since they did not have enough geographical feature and they discovered that by hovering the mouse on the map they can see the name of the provinces or states that they could remove or add to the map after filtering. Another matter that leads to high uncompletion of this task was that the accent of the users influenced the process of the transcription. For the Italian native speakers, the could easily procurance correctly the name of the provinces in Italy for the thematic map of Italy. However, for the map of the United States they faced difficulties for pronouncing the name of the states correctly. The situation was opposite for the Persian native speakers. In overall, the matter of accent does not consider as an issue since the predefined library of the Web Speech API is based on the verity of languages and their original accents. Moreover, we already assumed the language of the application English both from Great Britain and United states. Therefore, the accent of the speakers is a matter which inevitable on every voice virtual assistant system.
During the performance of the test, users tend to use the help command to undrestand the voice commands, since they do not wish to use their own commands. For example, the first 4 tasks as the Table 3. Shows as a high completion rate, with respect to other tasks since the task and their characteristics were quite intuitive and obvious for the users. As the task required the domain knowledge for understanding their nature, the completion rate also dropped.
In terms of the level of difficulty, tasks with a higher error rate and lower completion rate can be considered more difficult. Based on this, tasks 4, 7, 8, 10, 11, 15, and 16 can be considered more difficult than the other tasks. These mainly related to the complexity of the verbal commands that has been considered for these tasks on the application. Moreover, these tasks are highly related to domain knowledge and being familiar with

geospatial data visualization and processing. For example, task 11 is related to control the opacity of the thematic map that exists on the application and the associated commands to this task are:

- Increase the opacity of the map
- Decrease the opacity of the map

However, there are synonyms for these commands on the application, the users did not understand the characteristic of the task and they must ask for the help command from the application to find the commands related to this task.

In conclusion, we conducted a user study to evaluate the usability of a mapping application. We recruited 10 participants with diverse backgrounds to complete 18 tasks while we collected data on their task completion rate, task time, error rate, and user satisfaction score.

Overall, the results showed that most participants were able to complete most tasks successfully, with an average task completion rate of 92%. However, there were variations in the level of difficulty for each task, with some tasks being easier than others. For example, Tasks 1 and 2, which required basic navigation and asking for help, were completed successfully by all participants, while Tasks 4 and 15, which required more complex navigation and filtering, were more challenging and had lower completion rates.

In terms of user satisfaction, participants generally reported a positive experience with the application, with an average satisfaction score of 8 out of 10. However, there were also some concerns raised by participants, such as difficulty in finding certain features, and some technical issues that caused frustration.

Our analysis also revealed some interesting trends based on participant characteristics. For instance, female participants tended to have higher completion rates and lower error rates compared to male participants. Additionally, participants with backgrounds in data science and computer science tended to complete tasks faster and have lower error rates compared to participants with backgrounds in law and urban planning. Graph 4.1. A combo chart that summarizes the information regarding the error rate, percentage of the completion and gender of the participants.

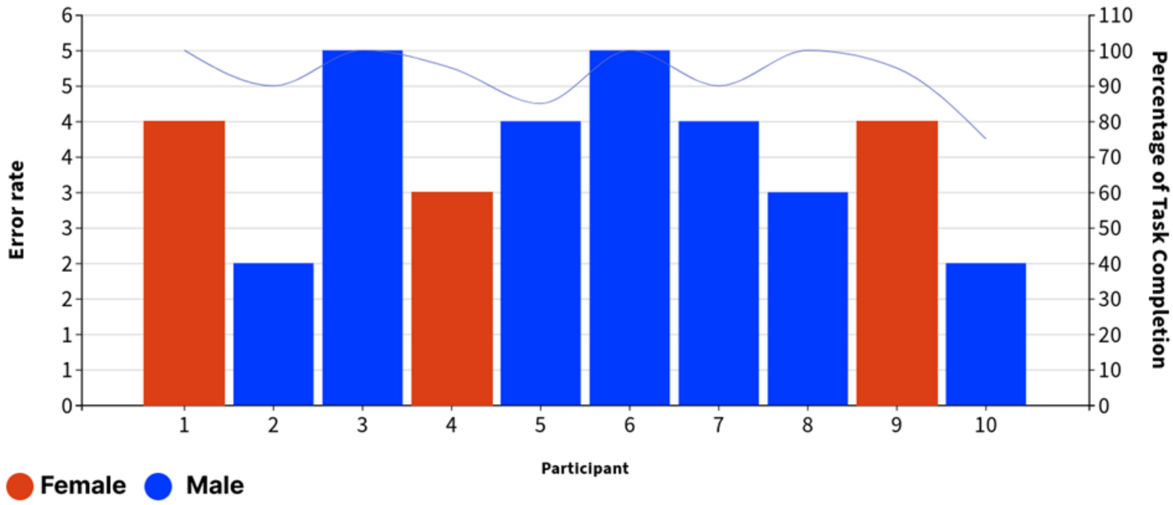

Figure 4.2: combo chart regarded the Error rate and completion rate of the test result

One possible explanation for this discrepancy could be that some users were able to recover from their errors and still complete the task, while others were not able to recover and had to abandon the task. Additionally, some errors may have been minor and did not significantly affect the user's ability to complete the task, while others may have been more significant and led to a longer task completion time.

Overall, it's important to take both the completion percentage and the error rate into account when evaluating user performance on a task. The two metrics can provide different insights into the user experience, and a combination of both can help to identify areas for improvement in the design of the application or the usability of the interface.

In conclusion, our findings provide valuable insights into the usability of the mapping application and can help guide future improvements to enhance user experience and increase task completion rates.

5 | Conclusions and future developments

According to *Dodge[21]*, visual representations possess the ability to stimulate one's creativity for the purpose of investigating and addressing problems, as well as to simplify the identification of patterns within intricate geographical data. In recent years, the rise of web-based platforms for creating and sharing geospatial visualization has made this technology more accessible to wider range of the people. Embedding such applications on web, enable users to have their own contribution on creation and visualization of geospatial data, only with an internet connection. This democratization of geospatial data allows people to uncover insights that could have significant societal impacts.
Embedding a voice virtual assistant withing a web application that let users visualize and process geospatial data can provide an additional layer of accessibility and interactivity. Voice assistants can offer a natural and intuitive way of interacting with data and can make the visualization experience more accessible to users who may have difficulty navigating the interface using traditional methods, such as mouse and keyboard.
Removing the boundaries of domain knowledge for geospatial data visualization and rising interest to this matter, still lacks efforts. According to the Gartner's 2018 Hype cycle virtual assistants were expected to become highly productive and have new become a common feature in everyday life, acting as a new gateway to technology [28]. However, their effectiveness is still limited in certain domains. One of such a domain is in the field of geospatial data visualization. In this context, the problem lies in the difficulty that virtual assistants face in understanding and processing the semantics and context behind the user's commands, leading to inaccurate results. Moreover, the cognitive load of the users regarding the tasks for geospatial data visualization and processing mainly related to complexity of the tasks and their domain knowledge for geospatial data. To address these issues, research was conducted to collect data that could help to enhance human-computer interaction for geospatial data visualization and developing guidelines for human-computer interaction that meet the requirements of users who need smart assistants in the GIS field.
This study details the initial efforts aimed at developing specialized technology for voice

user interfaces on a platform designed to support users with geospatial data visualization and processing, utilizing the English language. The initial stage involved the conceptualization of all the possible tasks already present on the platform into a discourse methodology or framework specialized for geospatial data. To this end, a methodology that was tested and embedded on a GeoDialogue in a previous study was also considered for this research case.

The subsequent phase involved collecting the best possible corpus and terminologies for voice commands regarding the characteristics of the considered tasks. To achieve this, a survey was conducted with the aid of visual contents to convey the different tasks to users and extract lexical choices based on the natural language of the users. Metrics were utilized to determine the relationship between the chosen words. The collected verbs and words, as well as the most frequently used terminologies, were then considered as voice commands.

An unexpected finding from the survey was that people tend to speak to a map with fewer and shorter words. Additionally, the real concept of visualization remains dependent on words and verbs mainly related to it. Based on the findings, the verbs "show" and "change" had the highest frequency among the used terminologies, implying a relationship between the visualization context and the functionality of natural language in human interaction with voice virtual assistant applications.

The last step involved developing the application on the platform and testing it. The frameworks chosen had to be coherent with the existing language and architecture of the application and be open-source and free of charge. The primary core of the development was the Web Speech API, which enabled the entire application to realize the main concept of a voice assistant. Other frameworks and libraries adoptable for geospatial data visualization, such as leaflet, Mapbox geocoding, and the JavaScript language, were utilized to develop the application.

During the testing phase of the application, challenges emerged, mainly related to aspects of natural language and accents. One of the challenges was the application's dependency on domain knowledge. Although this matter was revealed during the survey analysis, in practice, users had difficulty understanding tasks after issuing complex commands to the application. Commands that were easy to understand and more general in terms of usage, such as navigating to different cities, zooming in/out, movement and exploration, changing the color of features, and changing the basemap, were very easy to understand and use, even without using the help command of the application. In contrast, commands such as filtering maps, changing the steps of classification, and increasing or decreasing the cluster radius had a lower rate of comprehension and a high error rate. This implies that processing geospatial data and querying it with a voice assistant is still a topic that

is highly dependent on users' domain knowledge, and efforts must be taken to ease this matter.
Technical challenges were another impact regarding the testing of the application. Users have the tendency and expectation of AI from the application. Users prefer to issue short, general, and easy commands that the application must understand. However, this challenge is highly related to the fact that users come from different backgrounds and languages, and among the users who attended the testing phase, English was their second language. Therefore, users had difficulty expressing commands with the correct accents to be transcribed correctly by the application, and as a result, the application did not execute the desired action.
Here are some possible future work perspectives for studies this:

- Incorporating machine learning algorithms to enhance the natural language processing capability of the voice assistant and enabling it to undrestand complex commands.
- Developing multilingual voice assistants for geospatial data visualization and processing to cater to broader user base.
- Exploring the use of augmented reality technologies to supplement voice assistants and improve geospatial data visualization and processing.
- Investigating the use of voice command technology in remote sensing applications, such as detecting and monitoring environmental changes.
- Developing and improving existing programs and applications that are specialized for the smart home assistants such as Alexa, specifically for geospatial data query in context of smart cities and smart home.
- Conducting user studies to evaluate the usability and user satisfaction of voice assistants for geospatial data visualization and processing.

In conclusion, the development of specialized technology for voice user interfaces that can support users with geospatial data visualization and processing is an exciting and rapidly evolving field. The use of natural language and voice-based systems can greatly enhance the user experience and improve accessibility to geospatial data. However, challenges such as domain knowledge and natural language processing continue to persist and require further research and development. As technology continues to advance, there are numerous opportunities to apply voice-based systems to various fields, such as smart cities, environmental monitoring, and more. Overall, the potential benefits of voice-based systems for geospatial data are significant and hold great promise for the future.

Bibliography

[1] Chatgpt. https://chat.openai.com/chat.

[2] Datawrapper. https://www.datawrapper.de/.

[3] Flourish. https://flourish.studio/.

[4] Openai. https://openai.com/.

[5] Qwhery. URL https://qwhery.com/.

[6] *Proceedings, 2009 International Conference on Computational Intelligence and Software Engineering (CiSE 2009): December 11-13, 2009, Wuhan, China*, 2009. IEEE.

[7] Hypertext transfer protocol (http/1.1): Semantics and content. http://www.rfc-editor.org/info/rfc7231, 2014.

[8] Openstreetmap, Accessed: 2023-03-30. URL https://www.openstreetmap.org/.

[9] V. W. Anelli, T. Di Noia, E. Di Sciascio, and A. Ragone. Anna: a virtual assistant to interact with puglia digital library (discussion paper). 2017. URL http://data.bnf.fr.

[10] J. Austerjost, A. Nordmann, D. Reiser, A. Thünemann, N. Ochse, T. ten Brink, M. Biskup, and D. Staiger. Introducing a virtual assistant to the lab: A voice user interface for the intuitive control of laboratory instruments. *SLAS Technol*, 23(5): 476–482, Oct 2018. doi: 10.1177/2472630318788040.

[11] Y. Belinkov and J. Glass. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72, 2019. doi: 10.1162/tacl_a_00254. URL https://www.mitpressjournals.org/doi/abs/10.1162/tacl_a_00254. Action Editor: Marco Baroni. Submission batch: 10/2018; Revision batch: 12/2018; Published 3/2019. c 2019 Association for Computational Linguistics. Distributed under a CC-BY 4.0 license.

[12] D. S. Bhattarai, G. R. Ghimire, and B. Bhattarai. The role of chatbots in enhancing user experience. *International Journal of Advanced Computer Science and Applications*, 11(10):279–285, 2020.

[13] T. Blanco, S. Martín-Segura, J. L. de Larrinzar, R. Béjar, and F. J. Zarazaga-Soria. First steps toward voice user interfaces for web-based navigation of geographic information: A spanish terms study. *Applied Sciences (Switzerland)*, 13(4):2083, Feb. 2023. doi: 10.3390/app13042083.

[14] S. Bodker. Scenarios in user-centred design - setting the stage for reflection and action. In *Proceedings of the Hawaii International Conference on System Sciences*, page 124, 1999. doi: 10.1109/hicss.1999.772892.

[15] G. Cai, A. M. Maceachren, H. Wang, and S. Fuhrmann. Natural conversational interfaces to geospatial databases. *Transactions in GIS*, 9(2):199–221, 2005. doi: 10.1111/j.1467-9671.2005.00213.x.

[16] D. Calì, A. Condorelli, S. Papa, M. Rata, and L. Zagarella. Improving intelligence through use of natural language processing. a comparison between nlp interfaces and traditional visual gis interfaces. In *Procedia Computer Science*, volume 5, pages 920–925, 2011. doi: 10.1016/j.procs.2011.07.128.

[17] L. Clark, P. Doyle, D. Garaialde, E. Gilmartin, S. Schlögl, J. Edlund, M. Aylett, J. Cabral, C. Munteanu, and B. Cowan. The state of speech in hci: Trends, themes and challenges. 10 2018. doi: 10.13140/RG.2.2.17331.07202.

[18] P. J. Clarkson, R. Coleman, S. Keates, and C. Lebbon. *Inclusive Design: Design for the Whole Population*. Springer, London, 2003rd edition, 2003.

[19] P. R. Cohen and S. L. Oviatt. The role of voice input for human-machine communication. *Proceedings of the National Academy of Sciences*, 92(22):9921–9927, Oct 1995. Colloquium Paper presented at the National Academy of Sciences, The Arnold and Mabel Beckman Center, Irving, CA, February 8-9, 1993.

[20] P. R. Cohen, M. Johnston, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Glow. Quickset: Multimodal interaction for distributed applications. In *Proceedings of the Fifth International Multimedia Conference*, pages 31–40, 1997. doi: https://doi.org/10.1145/266180.266328. URL `https://dl.acm.org/doi/pdf/10.1145/266180.266328`.

[21] M. Dodge. Mapping and geovisualization. `http://en.wikipedia.org/wiki/The_Geographer`.

[22] M. Escalona and N. Koch. Requirements engineering for web applications - a comparative study. *J. Web Eng.*, 2:193–212, 01 2004.

[23] European Parliament and Council of European Union. Directive (EU) 2016/2102,

2016.
https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016L2102&from=EN.

[24] T. Fieldid. Architectural styles and design of network-based software architectures. 2000.

[25] S. Fuhrmann, A. MacEachren, and G. Cai. Geoinformation technologies to support collaborative emergency management. In *Collaborative Geographic Information Systems*, pages 287–307. Springer, 2009.

[26] T. Gilbert. Vocalgeo: Using speech to provide geospatial context in the classroom, 2020.

[27] C. Granell, J. Huerta, E. Klien, and X. Ye. A scoping review on the use, processing and fusion of geographic data in virtual assistants. *Transactions in GIS*, 25(4):1784–1808, 8 2021. doi: 10.1111/tgis.12720.

[28] M. Haklay. How good is volunteered geographical information? a comparative study of openstreetmap and ordnance survey datasets. *Environ Plann B Plann Des*, 37(4): 682–703, 2010. doi: 10.1068/b35097.

[29] A. Hunter and C. Tao. Ubiquitous gis, data acquisition and speech recognition. 01 2002.

[30] R. Jacobson and K. Sam. Multimodal webgis: Augmenting map navigation and spatial data visualization with voice control. In *Proceedings of the 9th AGILE International Conference on Geographic Information Science*, 2006. URL https://www.researchgate.net/publication/307205445.

[31] P. H. M. Jain, S. A. Ansari, and A. Agarwal. Voice interaction for web applications: Review of literature. *International Journal of Advanced Research in Computer Science*, 9(2):14–18, 2018.

[32] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Draft of Third Edition, 3rd edition, 2019. URL https://web.stanford.edu/~jurafsky/slp3/. Accessed on March 31, 2023.

[33] I. C. Kemper. 959-voice, a new interface to GIS. In *Proceedings of the 2003 ESRI International User Conference*, pages 1–9, 2003. URL https://proceedings.esri.com/library/userconf/proc03/p0959.pdf.

[34] P.-C. Lai and A. Degbelo. A comparative study of typing and speech for map metadata creation. *International Journal of Geographical Information Science*, 29(11): 2022–2043, 2015. doi: 10.1080/13658816.2015.1053182. URL https://doi.org/10.1080/13658816.2015.1053182.

[35] P.-C. Lai and A. Degbelo. A comparative study of typing and speech for map metadata creation. *AGILE: GIScience Series*, 2:1–12, June 2021. doi: 10.5194/agile-giss-2-7-2021.

[36] B. N. Larson. Gender as a variable in natural-language processing: Ethical considerations. *Association for Computational Linguistics*, 2017.

[37] Leaflet. Leaflet js, Accessed: 2023-03-30. URL https://leafletjs.com/.

[38] E. D. Liddy and J. H. Liddy. An nlp approach for improving access to statistical information for the masses, 2001. URL http://istweb.syr.edu/~tables/.

[39] I. Lokuge and S. Ishizaki. Geospace: An interactive visualization system for exploring complex information spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, page 409–414, USA, 1995. ACM Press/Addison-Wesley Publishing Co. ISBN 0201847051. doi: 10.1145/223904.223959. URL https://doi.org/10.1145/223904.223959.

[40] D. G. M. Shaw. *Software Architecture: Perspectives on an Emerging Discipline*. Pearson, April 2, 1996.

[41] Mapbox. Mapbox geocoding, Accessed: 2023-03-30. URL https://docs.mapbox.com/api/search/geocoding/.

[42] N. Medvidovic and R. N. Taylor. A classification and comparison framework for software architecture description languages. *IEEE Transactions on Software Engineering*, 26(1):70–93, 2000.

[43] C. Mellish, D. Scott, L. Cahill, D. Paiva, R. Evans, and M. Reape. A reference architecture for natural language generation systems. *Nat Lang Eng*, 12(1):1–34, 3 2006. doi: 10.1017/S1351324906004104.

[44] V. Morocho, R. Achig, J. Bustamante, and F. Mendieta. Virtual assistants to bring geospatial information closer to a smart citizen. In *6th IEEE Ecuador Technical Chapters Meeting, ETCM 2022*, 2022. doi: 10.1109/ETCM56276.2022.9935761.

[45] Mozilla Developers. Web speech api, 2022. URL https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API.

[46] N., B. C., M. A., and S. G. Sardella. An approach to conversational recommendation of restaurants. 2019.

[47] P. C. of Earth and M. Science. Overview of programming languages for gis. *Geospatial System Analysis and Design*, 2020.

[48] S. L. Oviatt. Multimodal interfaces for dynamic interactive maps. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'96)*, pages 95–102, New York, 1996. ACM Press.

[49] K. Panetta. Gartner. 5 trends emerge in gartner hype cycle for emerging technologies, 2021. URL `https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies/`.

[50] A. C. Robinson, I. Muehlenhaus, R. E. Roth, F. Harvey, X. Ye, F. Kessler, C. Ludwig, P. D. Boersma, M. Woods, and M.-J. Kraak. Geospatial big data and cartography: Research challenges and opportunities for making maps that matter. *International Journal of Cartography*, 3(sup1):32–60, 2017. doi: 10.1080/23729333.2016.1278151.

[51] B. Shneiderman, C. Plaisant, M. S. Cohen, S. J. Jacobs, N. Elmqvist, and N. Diakopoulos. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson, 2016.

[52] J. Sodnik, C. Dicke, S. Tomažič, and M. Billinghurst. A user study of auditory versus visual interfaces for use while driving. *International Journal of Human-Computer Studies*, 66(5):318 – 332, 2008. ISSN 1071-5819. doi: https://doi.org/10.1016/j.ijhcs.2007.11.001.

[53] W. Thompson. Annual digital assistant iq test–siri, google assistant, alexa, cortana, 2018. URL `https://loupventures.com/annual-digital-assistant-iq-test/`.

[54] D. A. Tyfa and C. Howes. Speech recognition for command entry in multimodal interaction. *International Journal of Human Computer Studies*, 52(4):637–667, 2000. doi: 10.1006/ijhc.1999.0355.

[55] H. Wang, G. Cai, and A. M. MacEachren. Geo dialogue: A software agent enabling collaborative dialogues between a user and a conversational gis. In *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, volume 2, pages 357–360, 2008. doi: 10.1109/ICTAI.2008.15.

[56] Z. Wu, Y. Wang, and C. C. Shi. Evaluating the effectiveness of voice command in web applications. In *Proceedings of the 10th International Conference on Information Technology and Science*, pages 171–177, 2019.

[57] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13(3):55–75, Aug 2018. doi: 10.1109/MCI.2018.2840738.

[58] C. Zhang, T. Zhao, and W. Li. Automatic search of geospatial features for disaster and emergency management. *International Journal of Applied Earth Observation and Geoinformation*, 12(6):409–418, 2010. doi: 10.1016/j.jag.2010.05.004.

A | Appendix A

Name	Definition
API	An API (Application Programming Interface) is a set of rules, protocols, and tools for building software applications. It allows different software applications to interact and exchange information with each other, providing a way for developers to access and use specific features or functionalities of another application or service.
SDI	SDI stands for Spatial Data Infrastructure. It refers to a framework of technologies, policies, and standards that enable organizations to manage, share, and use geospatial data and services efficiently and effectively.
Frontend	Frontend is part of creating websites or apps that users can see and interact with. It includes making buttons, menus, and other visual elements, as well as making sure everything is easy to use and looks good.
Backend	Backend is part of creating websites or apps that is not visible for the users. It involves building and managing the behind-the-scenes infrastructure, like databases and servers, that support the visible parts of an application. It ensures everything works smoothly, handles large amounts of data, and keeps things secure.
Geospatial data	Geospatial data is information that describes the location, shape, and characteristics of objects, features, and phenomena on the Earth's surface. It is typically represented as coordinates, such as latitude and longitude, and can be displayed on maps or other visualizations.

Geoinformatics	Geoinformatics is an interdisciplinary field that combines geospatial data analysis and visualization techniques with computer science, statistics, and other related fields.
Cartography	Cartography is the study and practice of creating maps. It involves the design and production of maps that represent and communicate spatial information and relationships, including features such as physical and cultural characteristics, boundaries, and locations.
GIS	GIS stands for Geographical Information System, and it refers to a system (computer-based tool) that enables users to visualize, store, analyze, and process spatial information.
HCI	Human-Computer Interaction (HCI) refers to the study of how people interact with computers and other technological devices.
Multimodal interface	A multimodal interface is an interface that allows users to interact with a computer system using multiple modes of communication, such as voice, gesture, touch, and gaze.
Metric	A metric refers to a quantitative field, that is aggregated to the original dataset, and enables the user to visualize data.
Dimension	A dimension can be both qualitative and quantitative fields, allowing users to break and categorize their data for data visualization.
.JSON	JSON (JavaScript Object Notation) is a lightweight data-interchange the format that is easy for humans to read and write, and easy for machines to parse and generate.
.SVG	SVG stand for Scalable Vector Graphics, and it is an XML-based the vector that is used to display the graphics and images on the web.
.geo.json	GeoJSON is an open standard format which is used to represent the Geographical features and their properties. Geospatial data are standardized in the machine and human readable format of the JavaScript Object Notation.
.topo.json	TopoJSON is a file format for encoding geographic data, similar to GeoJSON, but with a more compact representation.
DOM	In JavaScript, the Document Object Model (DOM) is a programming interface that represents a web page as a structured tree of objects. The DOM provides a way for JavaScript code to interact with and manipulate the contents of a web page.

CRS	It is a standardized way to describe and define the locations of features on the Earth's surface. A CRS provides a framework for specifying the position of a point or other geographic feature using a set of coordinates.
WGS 84	WGS 84 stands for World Geodetic System 1984. It is a standard coordinate system used to represent positions on the Earth's surface. The WGS 84 system is widely used in GPS (Global Positioning System) devices, as well as in other applications that require accurate geographic positioning information.

Table A.1: Table of Acronyms and definitions

Task	Task Description	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
1	Go to the link of the application	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	Asking for help from the application by saying the command help	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	Navigate through different places and let users choose their own words/verbs to find and explore different spots	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

4	Asking user to explore the area and changing the view of the map and moving to different directions	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
5	Display different layers of the map and change the base map	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
6	Switch to the marker map	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
7	Change the color and shape of the marker	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
8	Increase or decrease the cluster radius	✓	✓	✓	×	✓	×	×	✓	✓	×
9	Remove the marker map	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
10	Switch to the graduated color layer of the map	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
11	Control the opacity of the map	✓	✓	✓	✓	×	✓	✓	✓	✓	✓

12	Change the steps to preferred number of the user	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
13	Control the appearance of the legend	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
14	Change the color palette	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
15	Filter the regions	✓	✓	✓	✓	×	✓	✓	✓	×	×
16	Add/Remove the regions.	✓	×	✓	✓	×	✓	✓	✓	✓	×
17	Reset the filters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
18	Exit from the application	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table A.2: Table of result of the testing

B | Appendix B

```
function createSegments (palette,steps,data,method,shuffleColors) {
  var colors = generatePalette(palette,steps, shuffleColors)
  let limits
  switch (method) {
    case "quantile": {
      const quantile = d3.scaleQuantile()
        .domain(data)
        .range(colors)
      limits = quantile.quantiles()
      console.log("i got the values")
      break;
    }
    case "logarithmic": {
      colors = []
      const range = 2.83 / steps
      for (let i=range; i<=2.8301; i+=range) {
        let v = Math.floor(Math.E ** i) - 1
        if (v > 14) v = 14
        colors.push(palette[v])
      }
      //this.logg && console.log('COLORS',colors)
      const quantize = d3.scaleQuantize()
        .domain(d3.extent(data))
        .range(colors)
        .nice()
      limits = quantize.thresholds()
      break;
    }
    default: {
      const quantize = d3.scaleQuantize()
        .domain(d3.extent(data))
        .range(colors)
        .nice()
      limits = quantize.thresholds()
      //console.log('LIMITS',limits)
    }
```

```
  }
  limits = [0 , ...limits.map(el => Math.floor(el))]

  for (let i=0; i<steps; i++) {
    const item = {
      from: limits[i],
      to: limits[i+1],
      color: colors[i],
      // label: '',
    }
    segments.push(item)
  }
  console.log(segments)
  console.log(limits)
  //this.logg && console.log('segments',JSON.parse(JSON.stringify(
   segments)))
  return segments
}
```

Listing B.1: Function createSegments

```
window[`mapLayer${rn}`] = (layer, oldLayer) => {
            oldLayer && map.removeLayer(Layers[oldLayer])
            map.addLayer(Layers[layer]) // Initial Layer
        }
        window[`removeLayer${rn}`] = layer => {
            map.removeLayer(Layers[layer])
        }
        window[`zoomIn${rn}`] = () => {
            map.zoomIn()
        }
        window[`zoomOut${rn}`] = () => {
            map.zoomOut()
        }
        window[`panBy${rn}`] = direction => {
            switch(direction) {
                case 'north':
                    map.panBy([0, -100], { speed: 1 })
                    break
                case 'south':
                    map.panBy([0, 100], { speed: 1 })
                    break
                case 'west':
                    map.panBy([-100, 0], { speed: 1 })
                    break
```

```
                case 'east':
                    map.panBy([100, 0], { speed: 1 })
                    break
            }
        }
        window[`panTo${rn}`] = ({lat, lng, placeType}) => {
            if (lat) {
                if (placeType) {
                    map.flyTo(new L.LatLng(lat, lng), placeZoom[
placeType] ?? 5)
                } else {
                    map.panTo(new L.LatLng(lat, lng), { speed: 1 })
                }
            }
        }
        window[`removeCluster${rn}`] = () => {
            //map.removeLayer(cluster)
            cluster.remove()
        }
        window[`addCluster${rn}`] = () => {
            //map.addLayer(cluster)
            cluster.addTo(map)
        }
        window[`removeRegions${rn}`] = () => {
            layerGeo.remove()
            //map.removeLayer(layerGeo)
        }
        window[`addRegions${rn}`] = () => {
            layerGeo.addTo(map)
            //map.addLayer(layerGeo)
        }
        window[`increaseOpacity${rn}`] = () => {
            layerGeo.setOpacity(1)
        }
        window[`showLegend${rn}`] = () => {
            legend.addTo(map)
        }
        window[`hideLegend${rn}`] = () => {
            legend.remove()
        }
        window[`increaseClusterRadius${rn}`] = () => {
            cluster.remove()
            maxClusterRadius += 20
            makeMarkerCluster()
```

```
        }
        window[`decreaseClusterRadius${rn}`] = () => {
            cluster.remove()
            if (maxClusterRadius > 20) maxClusterRadius -= 20
            makeMarkerCluster()
        }
        window[`changeMarker${rn}`] = () => {
            let i = markerShapeNames.indexOf(this.layout.marker.shape)
            if (++i >= markerShapeNames.length) i=0
            this.layout.marker.shape = markerShapeNames[i]
            cluster.remove()
            makeMarkerCluster()
        }
        window[`changeMarkerColor${rn}`] = color => {
            this.layout.marker.color = color
            cluster.remove()
            makeMarkerCluster()
        }
```

Listing B.2: Voice commands on the voicemapChart.js

List of Figures

List of Tables

Listings

Acknowledgements

I would like to express my deepest appreciation to Professor Maria Antonia Brovelli and Professor Silvana Camboim for their contagious passion for the topic of my thesis and their invaluable patience and feedback.
I would like to dedicate this thesis to my beautiful sister "Ghazal" who is not here anymore, but her memory always remains with us as a joyful child and a fighter sister.
The greatest appreciation will go to my father and my family, that helped me to reach to this stage of my life. Without their sacrifice and endurance, I could not be here today.
Special thanks to my friends, flatmates, and colleagues in BStreams, Mr Massimiliano di Bartolo and Mr Claudio Veggioti for their support during the development and writing of my thesis. Finally, I would like to thank Nicolò for his faith, and support throughout the whole stages of my thesis.