A Re-Engineered Web Interface to a Global Citizen Science Portal

Onurcan Cirpik
November, 2023

SUPERVISORS:

dr. ir. Rolf A. de By
drs. Barend Köbben

A Re-Engineered Web Interface to a Global Citizen Science Portal

Onurcan Cirpik

Enschede, The Netherlands,
November, 2023

Thesis submitted to the Faculty of Geo-information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: M-GEO

SUPERVISORS:

dr. ir. Rolf A. de By
drs. Barend Köbben

THESIS ASSESSMENT BOARD:

dr. Frank O. Ostermann (chair)
dr. Robert Planqué (external examiner)

Disclaimer

This document describes work undertaken as part of a programme of study at the Faculty of Geo-information Science and Earth Observation of the University of Twente. All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the Faculty.

ABSTRACT

Citizen science has proved its significance by bridging the gap between the public and scientists, encouraging collaborative approaches to research methods and scientific processes. One of the ways to contribute to citizen science projects is through participating in avian sound collection projects such as the Xeno-canto (XC) platform. Effective data visualization is crucial in these efforts, as it transforms raw data into an accessible and interpretable form. This research project aims to improve the geospatial data visualization capabilities of XC. Considering XC's significant contribution to bioacoustic studies and ecological research, this study addresses data visualization and mapping functionality limitations that currently hinder user experience and data exploration. We propose a cost-effective approach to ensure the platform's long-term viability without the burden of additional costs by transitioning from a commercial web map library such as Google Maps to a license-free, open-source alternative such as Leaflet.

We use a combination of user-centered design (UCD) and a re-engineering approach to develop geospatial data visualization solutions tailored to XC and its users. One of the phases in our methodology is dedicated to usability testing, utilizing a structured questionnaire to evaluate the effectiveness, efficiency, satisfaction, and learnability of developed geodata visualization solutions. The outcomes for selected map types yielded encouraging results across these criteria. Key advancements of this research include a strategic move toward open-source development, transitioning to GeoJSON, and the implementation of a dynamic color coding schema.

Keywords

citizen science, xeno-canto, web map application, geospatial data visualization, reverse engineering, leaflet library, user-centered design, user experience

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my heartfelt gratitude to the people and organizations who contributed to the completion of my master's thesis.

I am sincerely appreciative of the opportunities provided by the Ministry of Environment, Urbanization, and Climate Change and the Ministry of Education of the Turkish Government. Their scholarship was a cornerstone in my ITC journey, and I cherish their trust and support in my academic endeavors.

I am deeply indebted to my first supervisor, Rolf A. de By, for his unwavering guidance, patience, profound knowledge, and continuous support throughout this research endeavor. His expertise, regular meetings, and mentorship played a crucial role in shaping my research and overcoming any challenges I encountered along the way. I am genuinely thankful for his invaluable insights and constant encouragement.

I would also like to express my sincere gratitude to my second supervisor, Barend Köbben, for his valuable contributions and guidance.

Special thanks are extended to Robert Ohuru for his indispensable expertise and support in software and coding, playing a significant role in the success of this thesis.

To my parents, I extend my deepest appreciation for their unwavering support and belief in me throughout my academic journey.

Last but certainly not least, I want to convey my gratitude to my dear friends Nishit, Jorges, Maryam, and Kaan for their friendship and moral support.

Thank you to everyone who has been a part of this remarkable journey!

TABLE OF CONTENTS

Abstract i

Acknowledgements ii

1 Introduction 1

1.1 Motivation and importance 1

1.2 Problem identification 3

1.3 Research identification 3

1.3.1 Overall objective 3

1.3.2 Subobjectives 3

1.3.3 Research questions 4

2 Literature Review 5

3 Research Method 9

3.1 Research designs 9

3.2 Strategy stage: Analyzing the questionnaire 10

3.2.1 Questionnaire description 10

3.2.2 Data analysis approach 10

3.2.3 Data analysis 11

3.2.4 Map type: Browse-by-region & taxonomy 12

3.2.5 Map type: Upload 14

3.2.6 Map type: Mysteries 15

3.2.7 Map type: User profile 15

3.2.8 Map Type: Forum 16

3.3 Scope stage: Selection of open-source JavaScript web map library and defining features of XC 16

3.3.1 Selection of the JavaScript Mapping Library 17

3.3.2 Introduction to Xeno-canto: web mapping applications 23

3.4 Structure stage: Analysis of the code base of XC 33

3.4.1 Files that generate core map features 33

3.4.2 Code-wise analysis map types: Browse-by-region, genus, species, subspecies and upload map types 37

3.5 Skeleton stage: Forward engineering approach 39

3.5.1 Introduction 39

3.5.2 Map core configuration and customization 40

3.5.3 Map types 41

3.6 Surface stage: Usability testing 52

3.6.1 Introduction 52

3.6.2 Questionnaire design 53

4 Results and Discussion **57**

4.1 Result of Strategy stage: Analyzing the questionnaire 57

4.2 Result of Scope stage: Selection of JavaScript library and defining features of XC 58

4.2.1 Selection of JavaScript library 58

4.2.2 XC features and functionality 58

4.3 Results of Skeleton stage: Forward engineering approach 59

4.3.1 Browse-by-taxonomy: Genus, species, and subspecies map types 59

4.3.2 Browse-by-region map 63

4.3.3 Upload map type 63

4.4 Results of the Surface stage: Usability testing 65

4.4.1 Section 1: Species and subspecies maps 65

4.4.2 Section 2: Genus map 66

4.4.3 Section 3: Browse-by-region map 66

5 Conclusion and Future Work **69**

A Integration of the Developed Code into the XC Codebase **71**

A.1 Species and subspecies map 72

A.2 Genus map 73

A.3 Browse-by-region map 74

A.4 Upload map 74

List of References **75**

LIST OF FIGURES

3.1 Sequential stages in methodology 9

3.2 User knowledge of browsing the internet and using web maps 12

3.3 Number of questions on Stack Overflow 22

3.4 Demonstration of eight map types on XC 24

3.5 Demonstration of a pop-up 27

3.6 Query on a Netherlands' region 27

3.7 PNG images for specific and unspecified subspecies, and background sound . . . 28

3.8 Demonstration of genus map type for genus *Anser* 28

3.9 Demonstration of Geoffroyus geoffroyi 29

3.10 Demonstration of Geoffroyus geoffroyi cyanicollis 30

3.11 Demonstration of upload map type 31

3.12 Demonstration of Forum: discussion about unknown species 32

3.13 Demonstration of mysteries map type 32

3.14 Demonstration of user profile map type 33

3.15 Overview of PHP scripts for application infrastructure and for each map type . 34

3.16 Demonstration of different frequency of markers 38

3.17 Demonstration of the customized map interface 42

3.18 Demonstrating data intensity levels from light blue to red 43

3.19 Histogram showing the distribution of species by number of subspecies. The x-axis represents the number of subspecies and the y-axis represents the frequency of species. Species with 17-49 subspecies are grouped together. 44

3.20 Illustration of seven base colors and their RGB/HSL values and the corresponding position on the color wheel 45

3.21 Illustration of colors and their corresponding positions on the color wheel . . . 46

3.22 Current map showing the error on the species ranges of *Corvus corax* 47

3.23 Example of the adopted JSON structure 49

3.24 Illustration of a marker as a pie chart 50

3.25 Illustration of a monotypic species 50

3.26 Illustration of polar region projection for Larus hyperboreus 51

4.1 Newly developed genus map type for the genus *Anser* 60

4.2 Illustration of current `species` map for *Corvus macrorhynchos* 61

4.3 Illustration of developed `species` map for *Corvus macrorhynchos* 61

4.4 Pie chart markers on the species map illustrating the distribution of recordings across multiple subspecies for *Vireo pallens* 62

4.5 Illustration of newly developed subspecies map for Corvus macrorhynchos tibetosinensis 62

4.6 Existing XC map showing data by clustering method 63

4.7 Map showing the number of recordings in a grid-based system 64

4.8 Map showing the number of species in a grid-based system 64

LIST OF TABLES

3.1 Responses for frequency of map types 13

3.2 Comparison features: Leaflet vs OpenLayers 19

3.3 GitHub repository information (as of November 2023) 21

3.4 Number of StackOverflow Questions (as of November 2023) 22

3.5 List of English names of bird species classified by Arctic, Antarctic, and Vast-range regions 25

3.6 Parameters for filtering sound recordings via the Xeno-Canto API 36

3.7 Description of Leaflet plug-ins used in map customization 40

4.1 The table displays a comparison of features for different map types, detailing attributes. For each map type, the presence of a feature is indicated with a checkmark (✓) and its absence with a cross (×). Exclusive functions for each map type are also listed in the last column. 59

4.2 File size comparisons before and after compression 60

4.3 User ratings of map features. Summarized user feedback on map functionality features, with average scores. 65

Chapter 1

Introduction

1.1 MOTIVATION AND IMPORTANCE

Citizen science commonly describes the active participation of individuals from the public in scientific research and can be initiated by scientists or arise from a grassroots level. The driving force behind these projects is citizens' interest and initiative, whether motivated by a desire to contribute scientific knowledge or personal interest in the subject matter. citizen science has been pursued for centuries and utilized in various fields, including science, research, politics, and education (Vohland et al., 2021). Especially technological advancements in recent years have made citizen science possible to grow tremendously (Skarlatidou et al., 2019). This growth, which may bring together volunteers from various communities, benefits both volunteers and scientists. While volunteers gain a better understanding of the scientific process, scientists benefit from having participants from varied backgrounds and locations in their scientific research (Yadav and Darlington, 2016).

Citizen science is mainly used as a methodology for data collection and classification in biology, conservation, and ecology research (Kullenberg and Kasperowski, 2016). Over the centuries, natural history museums have made tremendous collection efforts on soils, rocks, plants, and animals to document our planet's (bio)diversity, and collecting has become a high-level profession in itself. These collections have been enhanced in recent years by new technology-based collectibles, such as DNA/RNA sequences, GPS tracking data, and animal sound recordings. These additional collecting methods have helped to detect species and subspecies and their occurrence patterns that were cryptic under older methods.

One of the ways to contribute to citizen science projects is through participating in avian sound collection projects such as the Xeno-canto project[1]. XC is a citizen science project that intends to compile the world's most extensive open-access sound collection on wild birds to contribute to the study of bioacoustics and ecological research ("Xeno-canto", 2023). Through its web mapping application, the project strives to encourage bird sound recording, enhance accessibility, and raise bird sound knowledge. The portal hosts a database of geolocated bird song recordings covering 95 percent of all bird species worldwide and more than 700,000 recordings made by over 8,000 recordists. Since its inception 17 years ago, it has become a valuable resource for scientific analyses and publications and has been cited as an audio source in over 3,000 scholarly studies.

Birdwatching, or birding, is an action that includes activities such as watching birds in their natural habitats, tracking their movements, and identifying species and counting individuals. Birdwatching is useful because birds are regarded as ecological indicators due to their high sensitivity

[1] https://xeno-canto.org

to environmental changes (García-Moreno et al., 2007). Furthermore, birds provide many vital biological services such as nutrient cycling, biodegradation, insect control, plant fertilization, and seed separation (Whelan et al., 2008). Any decrease in bird habitats or numbers can be seen as an indicator that the environment is at risk (European Commission, 2004). Therefore, given all these ecological and biological benefits, bird monitoring is crucial for biodiversity and ecological sustainability research (Priyadarshani et al., 2017). Systematic analysis of bird sounds and recording localities can improve our understanding of bird behavior and identification. These observations form a part of a scientific dataset that enables researchers to monitor changes in species diversity and ecosystems (Lindsey Jean Schueman, 2021).

The use of software systems has become increasingly prevalent in modern society. They play a crucial role in enabling individuals and organizations to accomplish their objectives. As software systems evolve and grow over time, they may become challenging to maintain, extend, or adapt to new environments (Laato et al., 2022). To deal with these challenges, software re-engineering is often employed as a strategy. Software re-engineering can be defined as analyzing and modifying existing software systems to improve their design, functionality, performance, or other qualities. It can also be used to introduce new technology or programming languages or to modernize legacy systems and make them more relevant in today's technological environment. This may involve refactoring the codebase and adding or removing features. It is often necessary when a software system has become outdated or is no longer meeting the needs of its users (Majthoub et al., 2018). This process involves both reverse engineering and forward engineering (W. Lin, 2017).

Reverse engineering is the process of analyzing a system or method in order to understand its design, functionality, and operation. This process involves breaking down the system or method into its components and examining each individually to understand how they work together. It is often used to identify areas for improvement or potential issues or vulnerabilities (van den Brand et al., 1997). Reverse engineering is commonly used in software development to improve or modify existing systems or create new ones based on existing ones. On the other hand, forward engineering is the process of implementing changes to a system based on the insights gained through reverse engineering. It involves taking the results of the reverse engineering process and using these to create new designs, prototypes, or systems (W. Lin, 2017). Forward engineering is typically used to implement changes or improvements to a system.

The construction of web applications from a technical standpoint is not a major concern, as the necessary technology has already been developed and is well understood. Instead, the major issues come with creating a design under strict conditions that satisfy users and allow additional data exploration functions per the user needs/wishes (Garrett, 2005). Software re-engineering is not always a straightforward process. It can be challenging, especially for large and complex systems. It requires careful planning and a thorough understanding of the existing system, users, performance considerations, and stakeholders' requirements. It also requires strong problem-solving and debugging skills, as well as the ability to maintain compatibility with existing data and systems. In order to successfully re-engineer software, it is important to prioritize functional continuity and carefully consider the potential impacts of current platform modifications. Software re-engineering can be a challenging but rewarding endeavor that requires careful attention to detail and a thorough understanding of the system and its users. In forward engineering, we develop a set of proposals to change the code base. These proposals provide a comprehensive understanding of the desired modifications, motivations, and goals behind them to allow the software engineering team to understand the scope and implications of the re-engineering process.

1.2 PROBLEM IDENTIFICATION

The Xeno-canto project has been operational since 2005, and citizens and scientists are using it extensively for various purposes. Nevertheless, a number of issues stand in the way of the effective use of the Xeno-canto platform and its information contents. The first issue is that present mapping functions are quite limited, do not utilize modern cartographic techniques, and do not fully use data exploration capabilities. These issues result in an unnecessarily restricted user experience. Secondly, current mapping features in XC are generated by Google Maps. Although it is a powerful library, Google Maps API has become commercial in 2018. As a non-profit organization, XC currently does not pay for the usage of the API. However, there is a risk that Google may change its pricing or terms of service in the future, which would make running the project potentially costly. Due to potential financial risks associated with the Google Maps API, it is important for citizen science platforms to seek free and open-source software (FOSS) solutions.

While the collection of hundreds of thousands of sound files is a central aspect of this MSc project, the focus is not on acoustic analysis but on studying the metadata associated with sound recordings. The data that the platform collects is beyond just sound files. It also includes information about the in situ collection conditions, such as location, time of day, weather, characterization of the natural environment, distance to the microphone, the position of the vocalizer, and equipment used. In addition, the platform gathers data on the classifications of the sound types produced by the vocalizers (the birds). These birds vocalize for different purposes, and the characteristics of interest include the purpose of the vocalization such as song, mechanical sound, begging call, alarm call, contact call, duet, etc., the number, sex, and age of the vocalizers, the identification of the (sub)species. The platform also collects derivative information, such as the altitude at the location and the time of year (an indication of the season). The high dimensionality of this data offers numerous opportunities for users to explore and optimize their analysis. Individuals can gain a more comprehensive understanding of the underlying trends and relationships within the data by analyzing the parameters and variables within the dataset. Time and place are key variables in this study because they are potentially powerful explanatory factors that may enhance our understanding of temporal and spatial patterns in bird sounds. Therefore, this MSc project focuses on analyzing time and place metadata.

1.3 RESEARCH IDENTIFICATION

1.3.1 Overall objective

This MSc project aims to support a citizen science platform by developing improved geodata visualization solutions to enhance the capability of the Xeno-canto portal for exploring high-dimensional data. Additionally, this MSc project addresses the need for a more financially sustainable model by replacing the Google Maps API with an open-source and license-free alternative.

1.3.2 Subobjectives

- To analyze the questionnaire responses and understand the user requirements for map types on the XC platform, especially in explorative and analytical map-based functionality.

- To evaluate the suitability of open-source JavaScript mapping libraries for compatibility with XC.
- To analyze the current base code of XC and determine its functions.
- To implement new geodata visualization solutions to addressspecific needs and requirements of users.
- To assess the usability of the new geodata visualization solutions.

1.3.3 Research questions

RQ1: What are the primary motivations of users of the Xeno-canto portal?

RQ2: What challenges and requirements do users have when using the Xeno-canto web mapping stack for analytical purposes?

RQ3: What are the open-source JavaScript mapping libraries capable of in meeting the needs of users of the Xeno-canto web mapping stack?

RQ4: What is an appropriate re-engineered web stack for XC that considers innovative map-based exploration?

RQ5: How effective are the new geodata visualization solutions in meeting the needs of the users of the Xeno-canto portal?

Chapter 2

Literature Review

Cartography was once limited to the generation of static maps on paper; however, interactive maps began to appear with the development of the Internet, particularly with the advancement of Web 2.0 technology (M.-H. Tsou, 2013). Along with Internet developments, there has been a surge in the number of frameworks for web mapping (Edler and Vetter, 2019). The majority of these frameworks now use JavaScript (Netek et al., 2019) because its mapping libraries enable various functions and controls that are essential for interactive maps (Velkoski et al., 2017) and ensure enhanced responsiveness, improved interaction capabilities, and comprehensive user interface (Casteleyn et al., 2014). These characteristics allow web-based map interfaces to more closely resemble maps in desktop GIS applications in terms of interaction (Haklay et al., 2008). Consequently, it has become feasible to create advanced web mapping applications that allow the exploration of geospatial information without extra software (Machwitz et al., 2019).

XC is a valuable source for bird research. Over 3,000 studies have used recordings of XC for their research. For instance, one research (Ortega and Sarria-Paja, 2022) used a subset of 3689 audio recordings from XC to identify and classify the 30 species of the Thamnophilidae family in the western mountain range of Colombia using bioacoustics. Similarly, another research (Zulian et al., 2021) estimates the global abundance of the Vinaceous-breasted Parrot. XC was one of the four datasets used in this study. The authors found that combining citizen science data provided better results in analyzing the distribution and abundance of species. Moreover, XC is a useful resource for bird vocalization recognition research. In research (Xie and Zhu, 2019), the recordings are used as a source of labeled data for training and testing bird vocalization recognition models.

As web mapping applications became widespread, scientific studies on user interface design, the content of the dynamic map, and mapping technology increased. As a result, web mapping applications have become increasingly complex, with a range of functions and features that can be challenging to design in a user-friendly manner. This highlights the importance of considering user needs and requirements in developing web mapping applications, which can be achieved using the user-centered design (UCD) method (M. H. Tsou, 2011). The UCD method focuses on putting users at the center of development and on continuously incorporating users' needs, data, and feedback into the product design process. It aims to strongly emphasize the user demands to tailor a product to the exact wishes of its users (Mülazimoğlu and Başaraner, 2019; Agarina et al., 2019). A successful interface from a cartographic standpoint has implications beyond just programming and debugging; it also necessarily involves a detailed analysis of user requirements and the integration of modern insights into cartography and map-based information exchange (Cybulski and Horbinski, 2020). In other words, the interface must offer adequate capabilities for users to interpret the map and engage with the information. Therefore, regardless of the application's power, it is seen as ineffective if it cannot fulfill the requirements.

There are existing guidelines for conducting user-centered activities. Still, they do not advocate a specific approach because it involves a flexible and adaptable design approach that considers the unique needs and preferences of the users. The flexibility and adaptability of the UCD approach make it a popular choice for projects that intend to tackle real-world problems.

The UCD approach was applied in several projects to create effective web mapping applications that fulfill user needs. One project (M. H. Tsou and Curran, 2008) applied the five-stage UCD method (Garrett, 2005) to the functions and content of a web mapping application for the U.S. Geological Survey (USGS) hydrologic data collection. The authors suggested that following this five-stage UCD method can assist developers in creating effective web mapping applications that fulfill the needs of users. Another project (Roth et al., 2015) combined the UCD method with the Usability and Utility approach (Nielsen, 1993; Holzinger, 2005) for their web mapping application that visualizes criminal activities in place and time. Overall, in this case study, a successful crime-monitoring web application was built. The authors concluded that following the UCD method is helpful for initial planning and defining the overall goals and objectives of the project, as well as identifying the specific features and functions that will be included.

Yet another project (Hancock et al., 2022) followed a similar methodology for creating interactive web-based atlases for visualizing ecological data. The UCD method was utilized to collect feedback on the usability and utility stages. The authors stated that continuous feedback from users during the development process proved invaluable in creating a product that effectively addressed their needs and desires, ultimately saving time and resources that would have otherwise been spent on post-release revisions. Additionally, researchers assume the products developed using user-centered design methodology will increasingly benefit the biodiversity community.

Various citizen science projects collect biodiversity observations, providing researchers and general users with data. One of these projects is the eBird[1] platform, which allows users to analyze specific bird species. It also provides a hotspot map showing which areas are hotspots of a diversity of species by performing regional searches. eBird does not use clustering for markers. Instead, it visualizes using a grid-based representation to indicate the number of markers it contains. As the user zooms in, more detailed visualization is provided with raster grids of the first 100km and then 20km. A similar approach is also followed by

iNaturalis[2], another platform for geodata visualization, employs a similar approach. The size of the grids changes depending on the zoom level, and markers are shown when it reaches the high zoom level. This way, it provides a general overview of the regional distribution of species observations. On the other hand, GBIF[3], another important citizen science project, combines markers as the zoom level decreases to create a single marker, visualizing a bubble map that grows as more markers merge. Both the eBird and iNaturalist platforms use Google Maps, while GBIF prefers OpenLayers for its web mapping applications.

There are other web map applications that are built with Leaflet. For example, The Cornell Lab[4], developers of eBird, use Google Maps as their web mapping library for eBird; however, for their other project, BirdNet[5] was built with the leaflet library. BirdNet is a website and mobile application used to identify bird sounds. Users can upload bird sound recordings, and it analyzes

[1] https://ebird.org
[2] https://www.inaturalist.org
[3] https://www.gbif.org
[4] https://www.birds.cornell.edu/home/
[5] https://birdnet.cornell.edu/

those sounds to determine which bird species they belong to. Instead of creating a complex map, it shows the location of the uploaded records with a marker at each zoom level.

Another web map application of a citizen science project is OBIS Mapper[6]. It is a tool offered by the Ocean Biogeographic Information System (OBIS)[7], which is a global platform that collects and shares marine biodiversity data. After users search for a species, it creates a grid-based representation of points, as in eBird. Another project is The IUCN Red List of Threatened Species, which is a website that provides detailed information on threatened species and includes distribution maps of these species. Both platforms decided to build their application with Leaflet library.

[6] https://mapper.obis.org
[7] https://obis.org

Chapter 3

Research Method

3.1 RESEARCH DESIGNS

This MSc project combines user-centered design (UCD) and software re-engineering methods to develop geodata visualization solutions for XC. The UCD method, as outlined in the *Elements of User Experience* (Garrett, 2002), is a systematic approach to design that focuses on understanding the needs and preferences of the users to create products and services that meet their needs. It involves five stages: strategy, scope, structure, skeleton, and surface. The software re-engineering method is incorporated into the UCD process to implement the system changes in a structured and systematic way. The methodology flowchart for project development is shown in Figure 3.1.

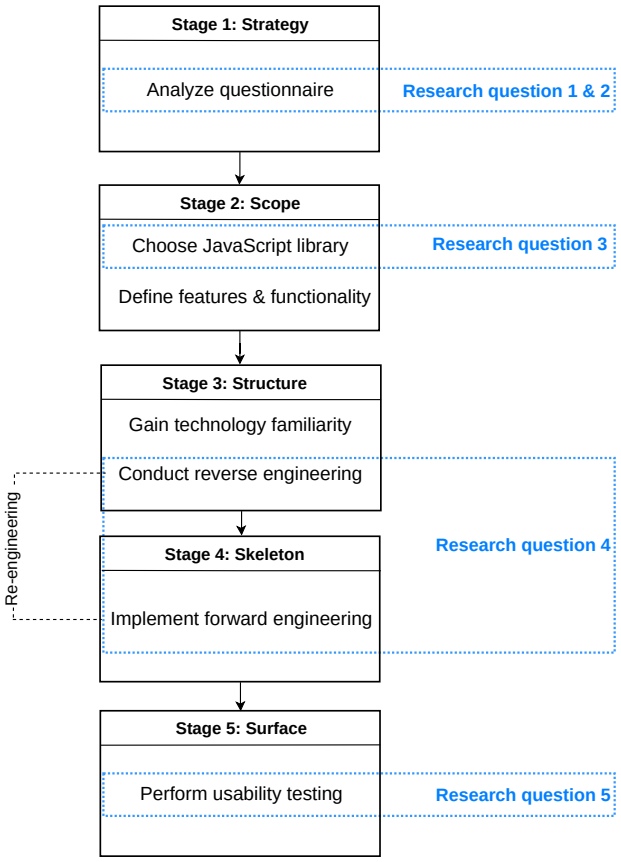


Figure 3.1: Sequential stages in methodology

3.2 STRATEGY STAGE: ANALYZING THE QUESTIONNAIRE

The strategy phase, as defined in *Elements of User Experience* (Garrett, 2002), is the first phase of the UCD process. This section presents a detailed examination of the questionnaire responses to understand the needs and motivations of Xeno-canto (XC) users. Through the analysis of responses to the questionnaire, we explore opportunities to enhance the user experience and data exploration capabilities of XC web map applications. The methodology for analyzing the responses to the questionnaire is mentioned below, and the results are discussed in Section 4.1.

3.2.1 Questionnaire description

The questionnaire was designed and executed in 2020 by a former student with the objective of getting insights from individuals who had interacted with XC maps. In this context, it successfully collected responses from 41 participants. Considering that there is no other research, this questionnaire presents a unique and invaluable opportunity to gain insight into the needs and motivations of XC users.

XC is a comprehensive platform that offers various tools and resources, including different map types that are created for certain purposes. Therefore, the questionnaire was designed in a way that questions were asked for each map type separately. The former student's decision was to divide these maps into five different types, namely (1) Explore region & taxonomy, (2) Upload recording, (3) Mysteries, (4) User profile, and (5) Forum. Users were asked to provide feedback on these five map types. In addition, they were asked to share their skills and motivations for using the XC platform.

The questionnaire consists of five questions for each map type. The first is a multiple-choice question stating how often this map type is used. Other questions are open-ended questions and expect a written answer from the participants. The second question asks about the users' purpose for using that map type. The third and fourth questions ask about which map elements they like, which should be kept, and which should be removed. Lastly, for each map type, it asks if they have any additional comments. Besides, participants also provided input to increase data exploration and analyze capabilities for map types `browse-by-region` & taxonomy. In addition to mapping types, the questionnaire also gathered information on users' web map applications, computer knowledge skills, and overall reasons for using the XC platform. Although a former student previously conducted the questionnaire, this study represents the first analysis of the responses.

3.2.2 Data analysis approach

An important point that requires emphasis is that the former student grouped `browse-by-region` and `taxonomy` map types under the same question title in the questionnaire. These map types are, however, used for different purposes. `browse-by-region` map lets users query a specific area on the map to display all recordings. On the other hand, `taxonomy` map type is used as the general name of the genus species and subspecies map types. And it displays recordings of a specific genus, species, or subspecies. As a result, `browse-by-region` and `taxonomy` maps require separate consideration as they have different potential for data exploration capabilities. Still, in the design of the original questionnaire, this distinction was not made, and both `browse-by-region`

and `taxonomy` map types were grouped under the same question title. We think it is unfortunate because these maps are different in nature. Even though the questionnaire is not well designed according to separating the map types, conducting a new questionnaire considering the limited time was not feasible. It would also be unwanted for users to fill out the survey again. Instead, the responses are analyzed to examine `browse-by-region` and `taxonomy` map types separately.

3.2.3 Data analysis

The data analysis process began with the examination of the responses to the user knowledge of web applications and browsing the internet to understand the users' familiarity in these areas. Subsequently, responses to the open-ended question, *Why do users use XC?* is examined. These initial steps aimed to understand the knowledge and motivation of users behind the use of XC. Next, an analysis was conducted on the responses to the question, *How often is each map type used?* to determine the frequency of usage for each map type. This question provided respondents with the option to select from multiple options. Later, the responses to the open-ended questions regarding the purposes for using each map type, suggestions for map elements to be included or excluded, and comments were analyzed.

Questions: Knowledge of internet and web map applications of users

The analysis began with the examination of the user's knowledge of web mapping applications and browsing the Internet. The participants stated their knowledge level of web mapping applications and internet browsing on a scale of 1 to 5, with 1 being the lowest and 5 being the highest.

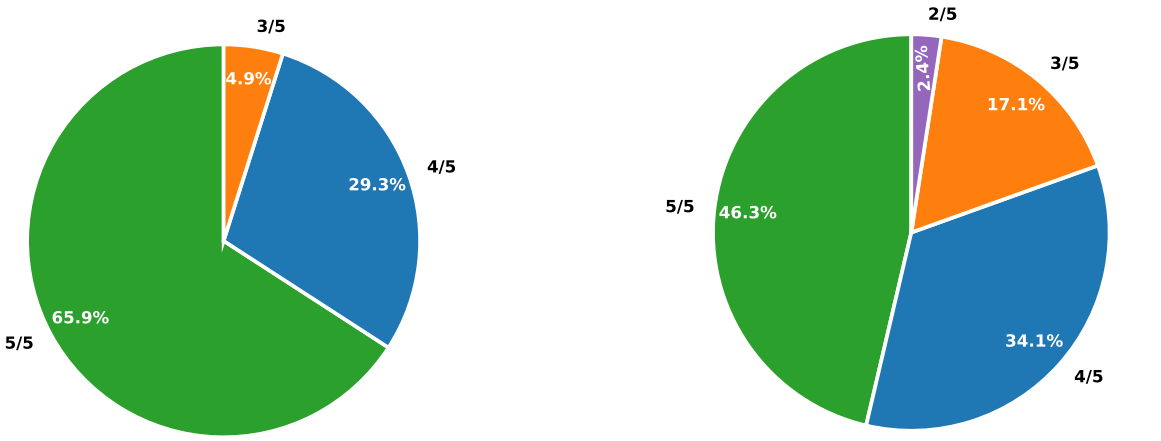
Users rated their knowledge of using web map applications, with 19 individuals considering themselves highly knowledgeable (rating of 5), 14 rating themselves as good (4), 7 considering their knowledge average (3), and only 1 person rating her/his knowledge as poor (2). For browsing the internet, 27 individuals rated themselves as highly knowledgeable (5), 12 rated themselves as good (4), and 2 individuals considered their skills to be average (3). Figure 3.2 represents this information in pie charts.

Overall, high numbers of users have strong confidence in using web mapping applications and browsing the internet. Their proficiency in these areas suggests they can easily comprehend complex geodata visualization-based solutions.

Question: Why do you use Xeno-canto?

The analysis of responses to the question *Why do you use XC?* provides an understanding of the motivation and purpose behind the use of the platform. From a thematic analysis of the open-ended responses, several common themes emerge.

One prominent theme is the wish to *upload & share* bird recordings, indicating that users actively contribute and share their recordings with the XC community. This helps XC to grow with more and more bird sounds over time. The theme of *education & science* highlights the platform's value as an educational and scientific resource. Researchers and scientists utilize the platform to study and analyze bird behavior and taxonomic changes, and they can learn about different species,



(a) Browsing internet

(b) Using web maps

Figure 3.2: User knowledge of browsing the internet and using web maps

their distribution, and vocalizations. The theme of *listen & compare* indicates that users use XC to listen to bird sounds to compare and identify different vocalizations. Users also utilize XC to *download* bird recordings for various purposes, such as creating sound libraries, using them in research, educational projects, or for personal enjoyment. Lastly, the theme of *enjoyment* points out that some individuals engage with XC for the pleasure of listening to bird sounds.

Question: How often do you use map types?

To assess the frequency of usage for each selected map type in the XC platform, participants were asked to indicate their usage frequency on a scale from *very frequently* to *very infrequently*.

Analysis shows that the map type `browse-by-region` and `taxonomy` is the most commonly preferred type and is used very frequently by 61.0% of users. It is followed by the `upload` map type, preferred very frequently by 46.3% of users. The map types `user profile` and `forum` have moderate popularity among users. Some 39% of users use the `user profile` very frequently or somewhat frequently, while this proportion is 29.3% for the `forum`. On the other hand, `mysteries` has a lower popularity among users. It was used very frequently or somewhat frequently by only 7.3% of users.

In summary, the findings show that `browse-by-region`, `taxonomy` and `upload` map types are more popular than map types related to the community, such as `mysteries`, `user profile`, and `forum`. The distribution of responses is shown in Table 3.1.

3.2.4 Map type: Browse-by-region & taxonomy

As mentioned in Section 3.2.2, it is important to highlight that the `browse-by-region` and `taxonomy` map types serve different purposes. Therefore, in our analysis, these map types are examined separately to increase the interpretation of responses.

Table 3.1 Responses for frequency of map types

Map type	Very frequently	Somewhat frequently	Occasionally	Somewhat infrequently	Very infrequently
Browse-by-region & taxonomy	**25**	8	7	0	1
Upload	**19**	11	7	1	3
Mysteries	2	1	7	8	**23**
User profile	6	10	10	2	**13**
Forum	8	4	13	2	**14**

Question 1: "For which purposes do you use these map types?"

Browse-by-region map

`browse-by-region` is used to search and explore a specific region. Users do this to locate recordings, plan birding trips, and assess the diversity of bird species in those areas.

Taxonomy: Species and subspecies map types

Mainly, this map type has been used to analyze the distribution range of species. Another reason is studying and comparing the vocalization of bird species and subspecies. Users want to compare their recordings with others to understand vocal variations and dialects among different bird populations. Some users have a firmer scientific background and focus on conducting research on taxonomy and global biodiversity or on assessing the influence of geographic factors in bird sounds.

Question 2–3: Which map elements must be kept/left out?

Overall, users expressed satisfaction with the current map elements. However, there was one specific function that participants found unnecessary and recommended removing, which is the Pegman.

Pegman is a special element used in Google Maps, which resembles an orange-colored human figure (Chandra Mishra and Dutta, 2023). This tool allows users to switch to street view. By dragging the Pegman icon on the map, users can explore a 360-degree photograph of the selected location.

Question 4–5: What additional mapping functionality would help you to explore further and analyze the data? & Additional comments?

Overall improvements

Firstly, participants highlighted the importance of mouse-controlled zooming capabilities and mobile-friendly map functionality. A smooth and intuitive zooming feature would enhance the

usability of the map. Furthermore, recordists expressed the desire for a wider range of basemaps, including options that represent land types and elevation levels. The ability to select among these options would allow users to customize their map view based on their specific needs and preferences, enabling a more comprehensive understanding of species distribution and habitats.

Browse-by-region map

Region map type presents a valuable exploration tool for bird species distribution. However, users have limitations to utilize this map type on mobile devices. Since the current functionality requires holding the shift button while drawing a bounding box, it is impractical to draw a bounding box on mobile devices due to the lack of a physical keyboard.

Species map

Participants emphasized the importance of filtering options to facilitate more focused and customized data exploration. They recommended incorporating filters based on month or season and sound type. These filtering options would enable recordists to analyze specific subsets of data, providing a more targeted and efficient research experience. While it is possible to filter the sound type using the search bar, users desire to see the sound type on the map. One possible way to enhance this feature is to utilize a visualization technique on the markers of recordings to indicate the sound type. In this way, users can read the sound type without making a query. Similarly, filtering results for specific months is possible, but visually representing months or seasonal trends on the map instead of relying solely on filtering is wanted from users.

In addition, users indicated the importance of toggling map elements such as species ranges or markers of recordings. This feature would enhance interactivity and provide a more personalized web map application experience.

Also, while the current map is limited to displaying a single species or a higher-order taxonomic entity like a genus, there is a strong user interest in having the capability to display multiple species simultaneously on the map.

Respondents also emphasized the need for a map projection type that facilitates more specific views. This requirement arises due to the limitations of using the Web Mercator, the default web map projection. It is a conformal projection that preserves angles and shapes but distorts the area. Thus, it has difficulties in accurately representing regions like the Polar regions (Battersby et al., 2014). It is desirable to provide an accurate area representation for citizen science portals focused on nature observations. Therefore, to accurately represent species distribution in the polar areas, using a polar-specific projection method helps make more meaningful analyses for users.

3.2.5 Map type: Upload

Question 1: “For which purposes do you use this map type?”

Recordists use `upload` map type to upload and verify the location of their bird sound recordings.

Question 2–3: Which map elements must be kept/left out?

The majority of respondents expressed a preference for keeping all map elements intact and suggested leaving out Pegman.

Question 4: Any further comments about this map type?

Many recordists emphasized the importance of zooming in and out using the mouse scroll wheel, as it offers a convenient and intuitive way to adjust the map view.

3.2.6 Map type: Mysteries

Question 1: "For what purposes do you use this map type?"

The majority of participants indicated that they have never used this map type or do not find it useful for their needs. However, several participants reported using the map to locate mystery recordings and explore unidentified sounds in unfamiliar locations. Additionally, some users used the map to upload their own mystery recordings and search for similar sounds from the same region.

Question 2–3: Which map elements should be kept or left out?

Users mentioned they do not use this map type frequently enough to have a strong opinion. However, like other map types, users suggested leaving out Pegman.

Question 4: Any further comments about this map type?

The common suggestion was to make the map more user-friendly by implementing easier zooming in and out capabilities.

3.2.7 Map type: User profile

Question 1: "For which purposes do you use this map type?"

Primarily, `user profile` map type is used to gather information about the locations and geographic coverage of recordings made by a specific user. Users visually explore the distribution of recordings by a specific user and discover the places that the user has been to desire to learn more about the visited locations, explore common fieldwork sites, and discover places that have been visited by the user.

Additionally, `user profile` is used for managing personal recordings. Users can review their recording lists, identify any blind spots or areas they are most familiar with in the world, and gain

a comprehensive view of their recording activities.

Question 2–3: Which map elements must be kept/left out?

As in other answers, leaving out Pegman was suggested. However, users either did not provide a clear opinion or had not explored this map type.

Question 4: Any further comments about this map type?

First, users emphasized the need for additional statistical information related to recordings, such as the number of recordings per location or the overall distribution of recordings by the user. These statistics would provide users with a deeper understanding of their recording patterns. Users expressed the need for a contact mechanism to enable communication with a recordist via email. This functionality would enhance collaboration and knowledge sharing among users.

3.2.8 Map Type: Forum

Question 1: "For which purposes do you use this map type?"

The responses indicate that users use this map type to verify the location where a recording was made.

Question 2–3: Which map elements must be kept/left out?

Users responded as with other map types and found all items useful except Pegman.

Question 4: Any further comments about this map type?

The responses were not detailed, with many users expressing confusion or not answering the question, resulting in limited insights.

3.3 SCOPE STAGE: SELECTION OF OPEN-SOURCE JAVASCRIPT WEB MAP LIBRARY AND DEFINING FEATURES OF XC

The `scope phase`, as defined in *Elements of User Experience* (Garrett, 2002), involves determining the specific requirements and features of the project. This section focuses on assessing the advantages and suitability of free and open-source JavaScript web mapping libraries. During the `scope phase`, the first task was to select an open-source JavaScript library that aligns with the goals and requirements of XC. Besides fulfilling technical and functional requirements, it is also important to take into account the economic issues highlighted by the XC team. The team pointed out the

need to move away from the current Google technology to lighten economic burdens. The selection of the JavaScript library was taken at this stage because we aimed to determine whether the selected library meets the required functions that would emerge in the subsequent stages.

3.3.1 Selection of the JavaScript Mapping Library

Criteria for library selection

Free and open-source software (FOSS) license is our main criterion for the selection of the library. This choice is motivated by both economic considerations demanded by XC teams and the desire to provide users with enhanced customization options.

In recent years, web-based geospatial data visualization has grown rapidly (Milić, 2022). The emergence of open-source JavaScript mapping libraries has revolutionized the creation of interactive web mapping applications. The use of an open-source library directly impacts the platform's sustainability, customizability, and usability (AlMarzouq et al., 2005). Developers now have the freedom to design and publish their web mapping applications that suit their unique needs without restrictions and financial burdens. Moreover, open-source libraries have the ability to allow problem-solving and knowledge-sharing within their communities (Newman et al., 2012). The free accessibility of open-source libraries leads to their widespread use, creating an environment for sharing ideas and knowledge. Developers can consult with other developers in these communities and find solutions when they encounter an issue. Furthermore, the dynamic and flourishing nature of this web ecosystem increases the development of third-party plug-ins. This allows developers to supply a broader range of functionality (Khitrin, 2017).

When it comes to selecting a free and open-source JavaScript mapping library that can handle big geodata visualizations without usage limits, there are two options worth considering. These libraries are Leaflet and OpenLayers (Ledur et al., 2016; Farkas, 2017; Zunino et al., 2020; Gede, 2023).

The most important factor is that Leaflet and OpenLayers are released under the BSD-2 Clause. This permissive license allows unrestricted use, modification, and distribution without additional costs (Ballhausen, 2019) and ensures that developers can benefit from their features and capabilities.

Leaflet and OpenLayers are the most popular web mapping libraries, and neither surpasses the other in every aspect. So, our analysis was not about finding an absolute winner but determining to what extent the effectiveness of these libraries responds to the specific XC user needs. In this regard, comparisons were made based on their features, ease of use, performance, documentation, and community support. Those criteria were chosen because our top priority was meeting user needs. To do this, first, we compared libraries for supporting all the necessary features. Secondly, we considered how easy the libraries are to use. Choosing a library that is easy to use allows beginner developers to quickly become proficient. This can result in a more efficient development process and also help reduce maintenance time at later stages. Thirdly, we compared their performance because it is another important factor in sustaining a responsive and interactive user experience, especially considering a large dataset like XC's. To keep users engaged, displaying data without any lag or freezing is crucial. Next, we evaluated the quality of documentation of the libraries. Documentation is the bridge between the creators of a software library and its users, especially developers. A well-documented library guarantees that its full potential is fulfilled. Lastly, we

value having strong community support. A well-supported library is important for troubleshooting, keeping up-to-date with the latest developments, and ensuring long-term stability. An active community provides access to valuable knowledge and resources, allowing developers to address challenges more effectively and efficiently.

Comparison of Leaflet and OpenLayers: features, ease of use, performance, and community support

While the below subsections provide a detailed comparison between Leaflet and OpenLayer, the selection of the library for implementation is further discussed in Section 4.2.1.

Feature comparison

As a result of analyzing responses to the questionnaire, the features that users expect from XC maps were determined. To address them, the selected library should support web browser and mobile compatibility features, offer custom basemaps, different projections, and geocoding services, and display data in KMZ, KML, and GeoJSON formats.

Leaflet and OpenLayers can function properly on any popular browser, such as Mozilla Firefox, Google Chrome, Safari, Opera, or Microsoft Edge (Leaflet, 2023; OpenLayers, 2023). Additionally, they are designed to work smoothly on tablets and smartphones and provide accessibility regardless of device.

The default projection of both libraries is Web Mercator, and neither library provides a default basemap, but they have the flexibility to select a desired basemap, including commercial and free map tile providers. The leaflet-providers[1] extension has good documentation for free map tile providers. These basemaps can be used for both libraries by giving credit to the web tile providers. Users can choose suitable basemaps for their needs, whether they would like to see a road map, terrain visualization, or satellite imagery. The ability to choose a basemap makes the mapping experience more personalized and engaging for users.

Both libraries have the ability to work with different projections using the Proj4js[2] library. This integration with Proj4js empowers developers to incorporate and manipulate geospatial data in a range of coordinate systems and projections within their web mapping applications.

Neither library provides an integrated geocoder, but they can be extended to support geocoding and reverse geocoding by using third-party services. These features are required to obtain the address information of the selected coordinates when users utilize the `upload` map for uploading audio recordings.

The XC API returns distribution ranges in KMZ files. OpenLayers natively supports KML formats, but working with KMZ files, which are compressed versions of KML, necessitates the use of a plug-in to unzip them. While Leaflet does not support KML initially, it does have a plug-in that can directly handle KMZ or KML files. Moving away from Google dependency also requires transitioning from the KMZ file format. In this case, GeoJSON files provide an alternative solution to KML and KMZ files. Both libraries provide built-in functionality to handle GeoJSON

[1] https://leaflet-extras.github.io/leaflet-providers/preview
[2] http://proj4js.org

and allow developers to display, style, and interact with geospatial features encoded in GeoJSON. GeoJSON files may not be the preferred format due to their large size. Instead, it is possible to use these files in a more compact format by compressing them in .zip format.

Table 3.2 Comparison features: Leaflet vs OpenLayers

Feature	Leaflet	OpenLayers
Web browser and mobile compatibility	Integrated	Integrated
Custom basemaps	Integrated	Integrated
Different projections	Via plug-in	Integrated
Geocoding services	Via Plug-in	Via plug-in
Display KML	Via plug-in	Integrated
Display KMZ	Via plug-in	Via plug-in
Display GeoJSON	Integrated	Integrated

In conclusion, the two libraries have similar feature capabilities to address user needs. OpenLayers has more built-in functions than Leaflet; however, it is also possible to have those features on Leaflet by getting support from its rich plug-in ecosystem. Therefore, both present effective options for developers to create customized web mapping applications. Feature comparison for both libraries is shown in Table 3.2.

Ease of use

In terms of ease of use, Leaflet is known for its simplicity and user-friendly API, making it an ideal choice for beginners. It provides a straightforward way to create interactive maps with just a few lines of code (Gong et al., 2019). On the other hand, OpenLayers has a more complex API that can be challenging to learn for someone who is not familiar with coding (Farkas, 2017). As a result, it may take more time and effort for developers to learn and get comfortable with OpenLayers than with Leaflet.

To compare the ease of use and simplicity of Leaflet and OpenLayers showcases[3] were created by implementing a simple map with features that are needed for current map types on XC, such as marker and pop-up, marker clustering, displaying a KMZ file, and making API call and displaying markers from GeoJSON response. The goal of those showcases was to compare the required code to achieve the same functionality in each library. Results showed that the initialization of the map, marker, and pop-up is straightforward in Leaflet. On the other hand, OpenLayer's initialization process has more steps, and the code is more wordy because adding a marker requires creating a feature, setting its style, and adding it to a vector layer.

Additionally, pop-ups in OpenLayers require creating an overlay and handling click events, which adds complexity. In Leaflet, marker clustering is achieved using the Leaflet markercluster plug-in, while OpenLayers supports marker clustering initially without the need for an external plug-in. With a plug-in, Leaflet provides a more straightforward way to handle KMZ files, but this introduces a dependency on external plug-ins. On the other hand, OpenLayers has built-in support for KML files. Leaflet's approach to fetching and rendering data appears more direct. The code structure is linear: fetch data, process it, and add it to the map. While OpenLayers involves

[3] https://onrcrpk.github.io/comparelibs

more steps: fetch data, process it into features, add features to a source, create a layer from the source, and then add the layer to the map.

We concluded that both Leaflet and OpenLayers have well-readable and expressive code. However, the required code to initialize the map with added features is significantly shorter and more straightforward in Leaflet in every case that we looked at. This is because OpenLayers does not provide default values for its variables, so developers need to explicitly specify them in their code. This leads to longer and more verbose code in OpenLayers. For developers who seek a library that prioritizes simplicity, Leaflet offers a more straightforward and beginner-friendly experience than OpenLayers. However, it is worth noting that OpenLayers inherently provides more advanced capabilities, making it more customizable without needing third-party plug-ins. This can be advantageous for projects that require a more extensive set of built-in features right from the start. The conclusion addresses the trade-off between starting simple and using plug-ins to advance the capabilities or starting more with a more complex code base to not depend on plug-ins.

Performance

When it comes to rendering performance, both Leaflet and OpenLayers offer smooth map interactions.

One aspect of comparing performance is the size of libraries. Leaflet has a smaller core file size than OpenLayers. For instance, the current JavaScript file of Leaflet version 1.9.4 is about 148 kB, while OpenLayers version 8.2.0 is around 800 kB. The size difference is primarily due to the latter having a more extensive set of initial features and functions, while Leaflet relies on third-party plug-ins to extend its capabilities. For example, OpenLayers supports various geospatial data formats, drag-and-drop, or advanced drawing features without the need for additional plug-ins, but Leaflet uses plug-ins to support those features. This plug-in based architecture of Leaflet allows developers to pick and choose the functions they need, keeping the core library lightweight. On the other hand, OpenLayers initially provides a wide variety of functions.

Currently, there are approximately 750,000 recordings on XC, and this number continues to increase. Although it is not necessary to show all those recordings for most scenarios, Leaflet or OpenLayers do not have the capacity to display such a large number of recordings in a single map. A study by Netek et al., 2019 expresses that Leaflet can handle up to 100,000 while OpenLayers can handle up to 500,000 points. The study also provides a website[4] for conducting these experiments. The information provided in this study might have been accurate at the time of its publication in 2019. However, in recent testing on their showcase, it was observed that Leaflet could successfully cluster 250,000 points. Though it took around 1 minute to load, it worked smoothly afterward. Additionally, even with 500,000 points, though it took approximately 3 minutes to load, the map worked smoothly afterward. This suggests that advancements in communication and computing technology and web mapping library optimizations have progressed in handling larger datasets.

Regarding the XC API, the maximum number of recordings displayed for the `browse-by-region` map type is limited to 2,000. There is no limit to displaying the number of markers for the `taxonomy` map types. The genus with the highest number of records contains about 8,000 records, while a single species has maximally about 2,500 records. When comparing the performance of Leaflet and OpenLayers, they are similar for the first 10,000 points. According to the study, Leaflet

[4] https://geoinformatics.upol.cz/app/bigdata/

renders the data at an average of 1572 ms, while OpenLayers renders an average of 1388 ms. The performance of libraries depends on technological improvements. Libraries themselves and hardware capabilities are also affecting this, and they are constantly changing. We can assume this performance will improve in the future.

Documentation & community support

Comprehensive documentation and an active community can provide productivity, assistance, knowledge sharing, and collaboration opportunities.

Documentation

Documentation plays a vital role in guiding developers by helping them use the library with ease. Both libraries have extensive documentation to explain their functions and features. They provide a quick start guide, documentation for their API, and tutorials of their functions and features with an interactive demo. They also address common user concerns through the FAQ section. Furthermore, both prioritize backward compatibility through documentation for their older API versions.

Leaflet suggests starting by simply introducing its HTML tag to the page and then setting up the map. On the other hand, OpenLayers encourages developers to use the npm package manager and Node.js[5] environments, which may require more time and effort to comprehend for those unfamiliar with this technology. Both libraries provide example usage with final maps and their implementation code. Overall, the extensive documentation provided by both Leaflet and OpenLayers empowers developers to use the full potential of these libraries.

Community support

To assess the size and activity of the communities, we can analyze various indicators such as GitHub and StackOverflow statistics.

Table 3.3 GitHub repository information (as of November 2023)

Repository	Stars	Forks	Contributors	Watchers
Leaflet	39,000	5,800	800	900
OpenLayers	10,500	3,000	350	400

The GitHub statistics demonstrate that Leaflet has a more extensive and engaged community relative to OpenLayers. The increased number of stars, forks, and contributors suggests interest and involvement among developers.

Leaflet displays a higher number of questions on Stack Overflow, potentially indicating a broader community and more widespread usage. A high number of questions may also indicate the difficulty of the library. To investigate this, StackExchange Data Explorer is used to extract

[5] https://nodejs.org/en

Table 3.4 Number of StackOverflow Questions (as of November 2023)

Library	Questions
Leaflet	14,000
OpenLayers	5,100

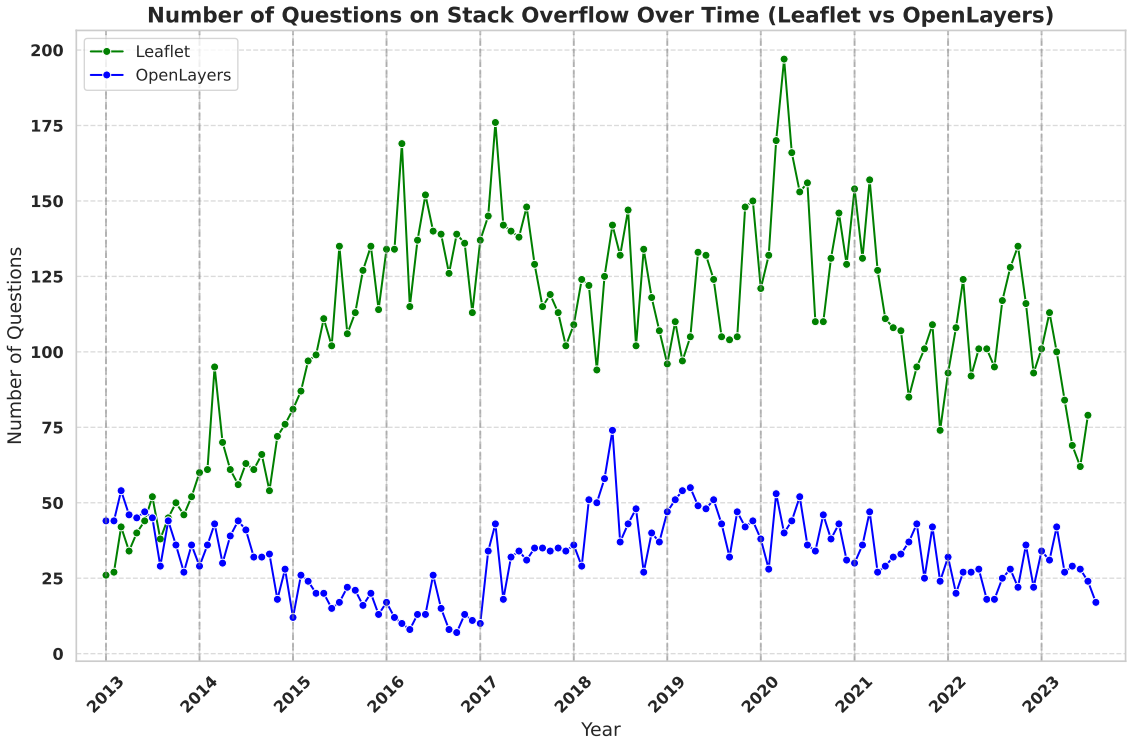


Figure 3.3: Number of questions on Stack Overflow

an overview of the number of questions for the last decade. Figure 3.3 shows that while there were more questions for OpenLayers in 2013, they were on a declining trend until 2017. On the other hand, Leaflet encounters more and more questions every year until 2017. Then, it continues with a fluctuating graph. Since Leaflet was created in 2011, it may have reached more and more users every year. On the other hand, OpenLayers was already more mature and stable since it was founded in 2005. This comparison shows that while the number of questions fluctuates over time, Leaflet gets about three times more questions overall than OpenLayers on StackOverflow. This high number of questions does not mean that Leaflet is harder to use than OpenLayers because when we examine the questions and their tags on StackOverflow, it shows that the questions were more about the use of the library with other frameworks, such as Vue, Angular, or React, than on basic use of the library. Considering the GitHub data, it can be concluded that Leaflet has a more active community and users.

3.3.2 Introduction to Xeno-canto: web mapping applications

This chapter explores web mapping applications within XC, providing comprehensive documentation for each map type. Map types and APIs are introduced briefly. A code-wise analysis of those map types and APIs is presented in Chapter 3.4. Results are discussed in Section 4.2.2.

Introduction of map types of XC

Xeno-canto is a collaborative online database and community platform for sharing bird sounds globally. Over time, this collective effort has resulted in a valuable resource for ornithologists, researchers, and anyone interested in bird vocalizations. In 2023, the XC platform enhanced its scope by involving grasshoppers and bats sounds in the portal. However, this study's focus is only on bird species.

The homepage of XC provides a toolbar menu that allows users to navigate and select their desired map type. Within the portal, there are eight different sections where users interact with a map. These map types have distinctive features and serve different purposes. Figure 3.4 demonstrates eight map types, representing their relative positions on the toolbar. According to the questionnaire responses, the first five map types are favored by users. Additionally, when we examine the available map types, the first four map types offer more potential for data exploration and analysis.

Configuration and customization of map

This section covers the core features of all map types. The similarities among all web map applications in XC include the use of the Web Mercator projection, the availability of two basemaps (a street map and satellite view from Google Maps), the inclusion of common map features such as zoom controls and a full-screen setting, and access to the Xeno-canto API v2.

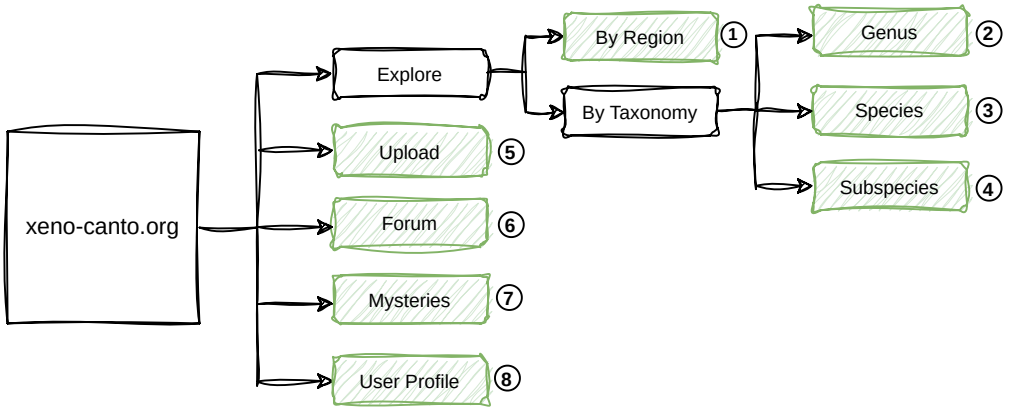


Figure 3.4: Demonstration of eight map types on XC

Map projection

The Web Mercator projection is employed across maps within the XC portal. It's worth noting that the portal only utilizes this projection and does not provide alternative projection options. The Web Mercator projection, also known as Google Web Mercator, became a standard for web mapping applications when it was adopted by Google Maps in 2005 (Battersby et al., 2014). While this projection is widely adopted in web map technology, it can distort geographical features, particularly in the polar regions (Farkas, 2017). This distortion exaggerates the apparent size of polar regions, resulting in a visual representation where these regions appear larger than their actual size. For example, Greenland appears much larger on a Web Mercator map than it actually is in relation to continents like Africa (Battersby et al., 2014). To address this, an alternative projection specifically tailored for these regions should be offered to enhance the maps in such areas.

XC contains recordings from all around the world, including polar regions. The Arctic Circle is defined as the latitude at which the sun does not set at the summer solstice and does not rise at the winter solstice. Although the exact location of the Arctic Circle changes slightly over time, its approximate location is 66 degrees, 34 minutes North, or 60 degrees (Varnajot, 2019). Similarly, the Antarctic Circle, defining the boundary of the Antarctic region, is approximately located at 66 degrees, 34 minutes South. In defining north polar species, our criteria included species ranges extending northwards of 60°N but not below 45°N. This specific threshold of 45°N was chosen not to include species with wide ranges, ensuring that our criteria included only those found in the Arctic region. Applying a similar logic for the south polar species, we considered those whose ranges extend southwards of 60°S but not above 45°S. There are 27 species inside the Arctic Circle. In the Antarctic Circle, there are three species in the XC database. 26 species have ranges in the south or north pole, but they also have ranges outside of those areas. By exploring alternative projections, XC has the opportunity to improve the accuracy and reliability of analyses conducted in these regions. Table 3.5 shows the species at the north pole and south pole and species with vast ranges.

Table 3.5 List of English names of bird species classified by Arctic, Antarctic, and Vast-range regions

Arctic Region	Antarctic Region	Vast-Range
Tundra Bean Goose	South Polar Skua	Grey Plover
Brant Goose	Emperor Penguin	Bar-tailed Godwit
Greater White-fronted Goose	Antarctic Shag	Ruddy Turnstone
Rock Sandpiper		Red Knot
Glaucous Gull		Sanderling
Iceland Gull		Broad-billed Sandpiper
Little Auk		Curlew Sandpiper
Arctic Redpoll		Pacific Golden Plover
Yellow-billed Loon		American Golden Plover
Steller's Eider		Bristle-thighed Curlew
Spectacled Eider		Eskimo Curlew
King Eider		Slender-billed Curlew
Emperor Goose		Great Knot
Lesser White-fronted Goose		Temminck's Stint
Pink-footed Goose		Red-necked Stint
Little Curlew		Baird's Sandpiper
Surfbird		Little Stint
Sharp-tailed Sandpiper		White-rumped Sandpiper
Purple Sandpiper		Buff-breasted Sandpiper
Ivory Gull		Western Sandpiper
Ross's Gull		Long-billed Dowitcher
Great Skua		Red Phalarope
Red-breasted Goose		Wandering Tattler
Barnacle Goose		Spotted Redshank
Siberian Nuthatch		Harris's Sparrow
Scottish Crossbill		Spoon-billed Sandpiper
McKay's Bunting		

Map basemaps

Regarding basemaps, two primary options are offered to users. First, the Google Maps default street map with an additional terrain option to highlight hills, mountains, and other physical features of the landscape. This option can be toggled as wished. Second, XC offers a satellite view, both with and without labels, providing aerial imagery. The basemap switch button is placed on the top left.

Map features

Zoom functionality using a mouse wheel is not allowed for any map type. Instead, users can zoom in using the zoom control buttons at the bottom right of the map or double-clicking the left mouse button. However, as outlined in Section 3.2.4, users highly desire to navigate the map with a mouse wheel. Inversely, Pegman is available for all maps, but users found it unnecessary. Additionally, the platform provides a full-screen button at the top-right of the map.

Xeno-canto API v2

The XC platform has provided a comprehensive RESTful JSON-based API since 2013. This API empowers users with custom queries that allow them to filter their searches based on more than 37 parameters. Whether researchers want to explore the calls of certain species, their behavior at different life stages, or temporal variations in vocalizations over time, the API gives them the power to filter their queries as they wish. This enables applications ranging from ecological research to educational tools showcasing bird vocalizations' rich diversity.

API for pop-up content

When a user clicks on the marker, it brings a pop-up. This pop-up's content is generated by the API, which delivers HTML code to compose the pop-up's contents. The pop-up consists of the id of the recording, a download button, licensing information, an image of a sonogram, an embedded audio player, species and recordist name, and location. ID serves as a unique identifier for the specific recording, allowing users to reference or share the recording. The download button allows users to download the recording file to obtain the audio for further analysis or personal use. Licensing information shows availability under Creative Commons licenses. Sonograms visualize the frequency and time characteristics of the sound. The audio player allows users to play the recording directly without having to download it. Figure 3.5 demonstrates the pop-up content.

Browse-by-region map

Region map type allows users to explore bird recordings coming from specific geographic areas and study the distribution of bird species in different regions. Users draw a bounding box over the map to select a region of interest. After executing a query, the results are displayed below the map, indicating the number of recordings and unique species. Users can click on the summary

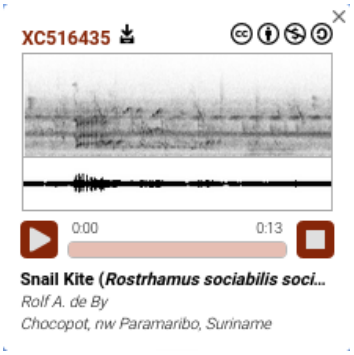


Figure 3.5: Demonstration of a pop-up

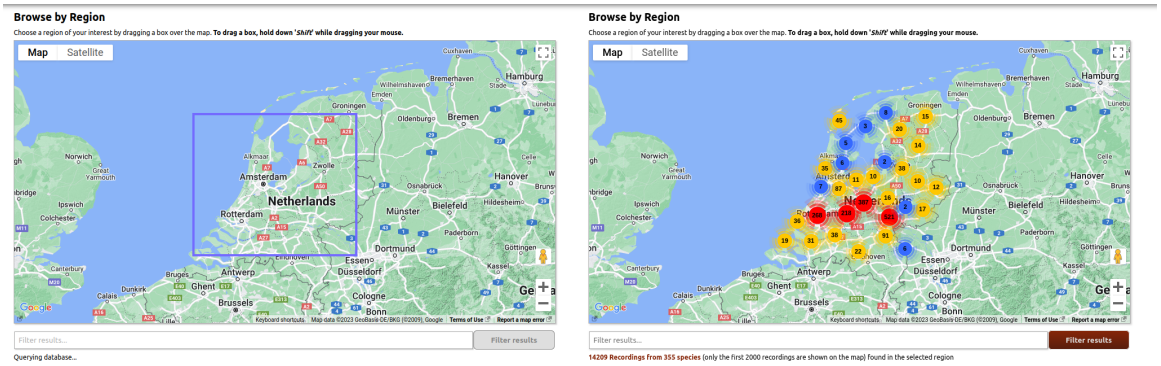


Figure 3.6: Query on a Netherlands' region

and can access the list of recordings. This list of recordings can be further filtered by using the xeno-canto API v2 through the search bar located below the map. This feature enables users to filter their search based on their preferences. Once a marker is clicked, a pop-up is provided with the AjaxPlayer API. In Figure 3.6, a visual demonstration of the `browse-by-region` map type is presented. The left image shows a bounding box drawn over some Netherlands' region, while the right image shows the results of the query.

Explore by Taxonomy: genus, species and subspecies map types

The taxonomy map type provides a taxonomic list from genus to subspecies level. Users can follow the family tree by clicking on the list. Creating a map is possible for genus, species, or subspecies levels. Unlike the `browse-by-region`, `taxonomy` maps do not employ marker clustering. Instead, each recording is shown through individual markers. When a user clicks on a marker, a pop-up displays content retrieved from the AjaxPlayer API. Additionally, a legend is provided to display the different species or subspecies on the right side. Their marker color, English name, and scientific name are shown in the legend.

In the existing XC approach, each marker is assigned a PNG image. There are 25 set markers, with 23 of them for subspecies and 2 of them for background sounds and subspecies not specified, respectively. Figure 3.7 displays the PNG images used as markers by XC.

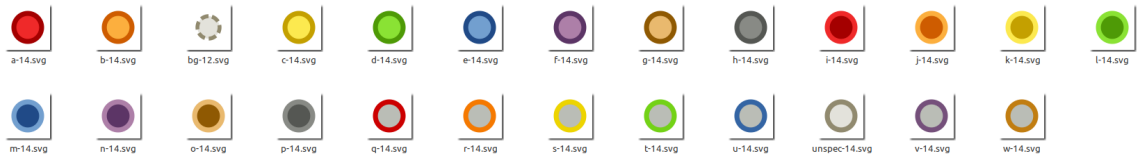


Figure 3.7: PNG images for specific and unspecified subspecies, and background sound

It is important to mention that the XC database contains order information only for the *species* level. Thus, the order of species on the genus map is ordered based on this; however, the order of the *subspecies* on the *species* map is based on alphabetical order.

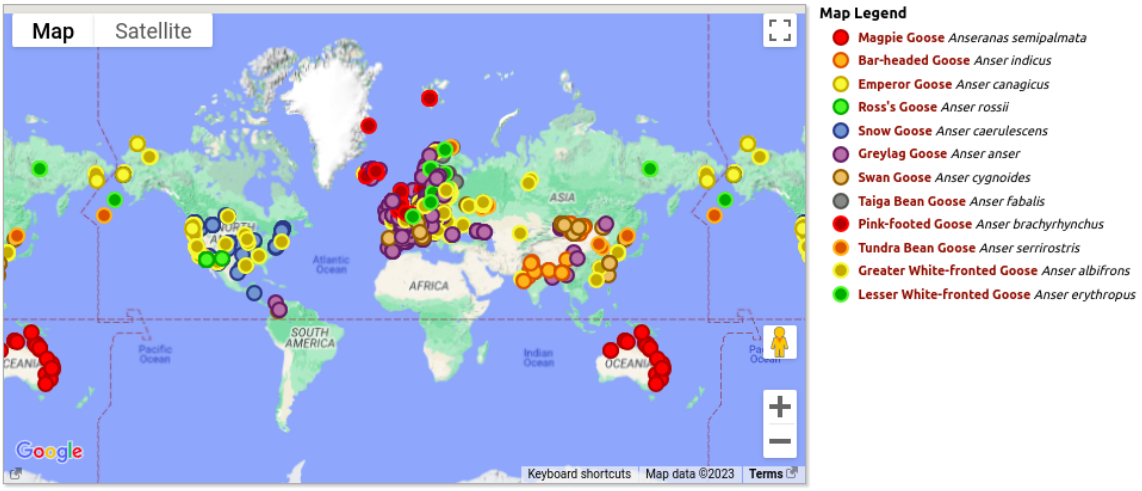


Figure 3.8: Demonstration of genus map type for genus *Anser*

Genus map The *genus* map type only shows recordings belonging to a particular genus of birds. It displays recordings individually marker and assigns different colors to different species. The filtering function and species ranges are not provided for this map type. Below the map, the list of recordings is provided. We would like to point out that currently, it faces problems when dealing with similar genus names, which leads to inaccuracy in the data presented. Some of the genus names are quite similar. For example, the genera *anser* and *anseranas* are different genera, but their names have a similar formation. As shown in Figure 3.8, the current genus map for *anser* shows recordings for *anseranas* as well.

Species map `Species` map provides valuable information to ornithologists, ecologists, bird watchers, and researchers interested in birds and their sounds. Users use this map to explore bird sounds, study bird behavior, or view the distribution range of a particular species or subspecies. After a user selects a bird species from the provided family tree list, if there is subspecies information, markers are assigned colors based on their alphabetic order of subspecies. If there is no subspecies information for a specimen, it shows as *no subspecies specified* with a gray color. This brings two issues, first, the markers do not provide insights by their colors, and secondly, it wrongly labels the monotypic species as *no subspecies specified* with a gray color.

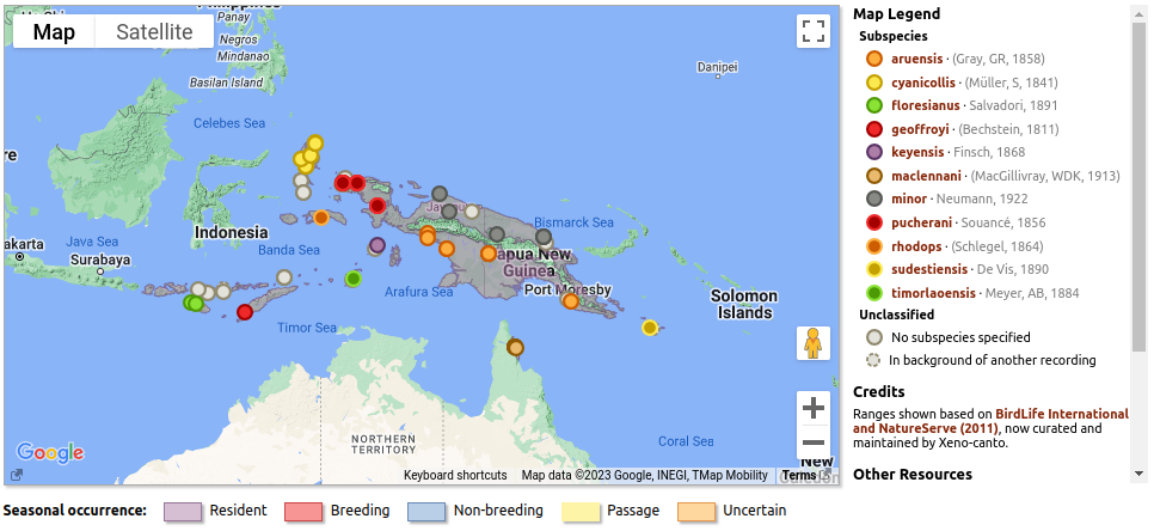


Figure 3.9: Demonstration of Geoffroyus geoffroyi

The map's initial view is determined by the locations of the records. If the recorded locations are close to each other, the map will zoom in on that particular area without considering the distribution range. In this case, users need to zoom out manually if they wish to view the entire distribution range.

The current assigning markers system first checks whether there is a nominate subspecies or not. If there is, it gives the first marker (red) to that subspecies, and then the rest get the markers in alphabetic order.

The most prominent feature of this map type is to provide a distribution range map for species with the use of KMZ files that are provided by XC. Distribution ranges are categorized as having resident, breeding, non-breeding, passage, and uncertain range parts. It is worth noting that it is not possible to toggle this layer.

Similarly to the `browse-by-region` map type, the filtering function is provided below the map. In this way, it is possible to perform the same queries for `species` map types as well. Moreover, connection information to other websites, such as eBird and Avibase, is provided about species or subspecies.

Subspecies map `Subspecies` map type is a sub-section of a `species` map type.

Sometimes, a subspecies is named identically to its species. In this case, these subspecies are considered as *nominate subspecies*. The nominate subspecies always get a red marker. But for other subspecies, the colors do not keep their constancy between species and subspecies map types. When selection is made, this marker is shown as the first possible marker, red or orange.

For instance, Figure 3.9 shows the *Geoffroyus geoffroyi*. The nominate subspecies *geoffroyi* is assigned with red colors, and then the rest gets the marker in alphabetical order. For example, if users make a selection for the subspecies *Geoffroyus geoffroyi cyanicollis*, they will arrive at the map shown in Figure 3.10. Despite the subspecies *cyanicollis* being yellow on the species map, its

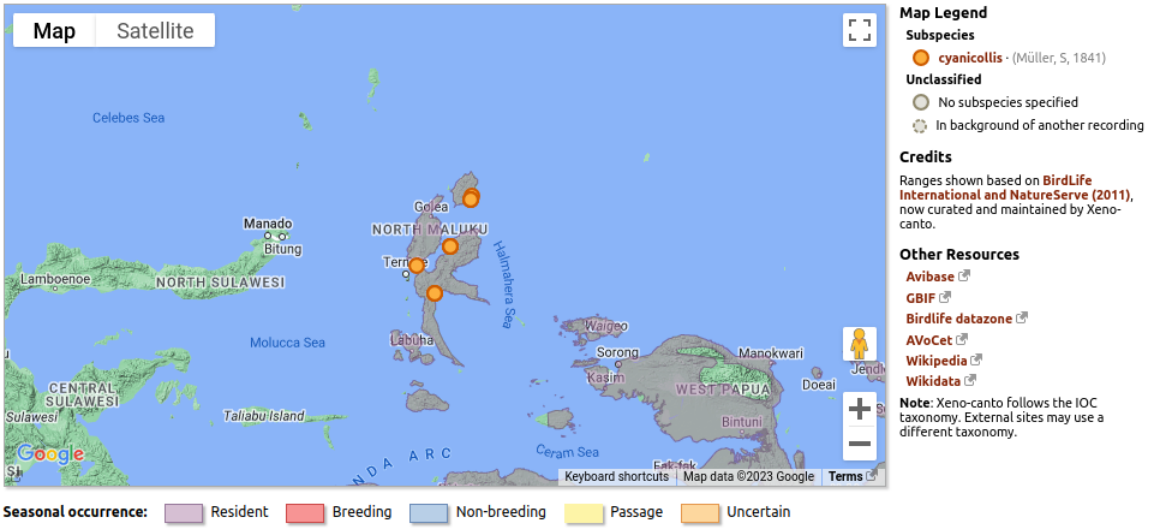


Figure 3.10: Demonstration of Geoffroyus geoffroyi cyanicollis

markers change to orange on the subspecies map.

Upload map

Users apply this map type to select and determine the coordinates of their recording while uploading it to the platform. When a point is selected on the map, the coordinates, address, and altitude of the selected location are displayed on the page. For this purpose, the upload map utilizes the reverse geocoding and elevation API of the Google Maps API to obtain the coordinates, address, and elevation of a selected point on the map. Reverse geocoding is a process that converts geographic coordinates into a human-readable address (Kounadi et al., 2013). Additionally, forward geocoding is provided by the search bar to zoom in on specific locations. Forward geocoding is a process that transforms a written address or a place name into geographic coordinates (Cambon et al., 2021). This feature is important for users who may only know the name of a location or have a specific address but not its coordinates. Users can search for a country, city, or well-known place to locate it on the map. Besides, users can discover nearby recordings by zooming into specific areas on the map.

Forum map

`Forum` is where users can discuss various topics. `Forum` is used to share knowledge and help others rather than data visualization. Users often post their recordings of unknown species and ask for clarification. As shown in Figure 3.12, users can comment on posts to share their thoughts. They can also create posts related to issues or requests about the website, map, taxonomy of recording, hardware, or software.

While users are discussing a particular recording, a map shows with a red marker where the recording was made. The pop-up of the marker shows the location details. Furthermore, if the species of recording is known, it also displays distribution ranges, which aids in assessing whether

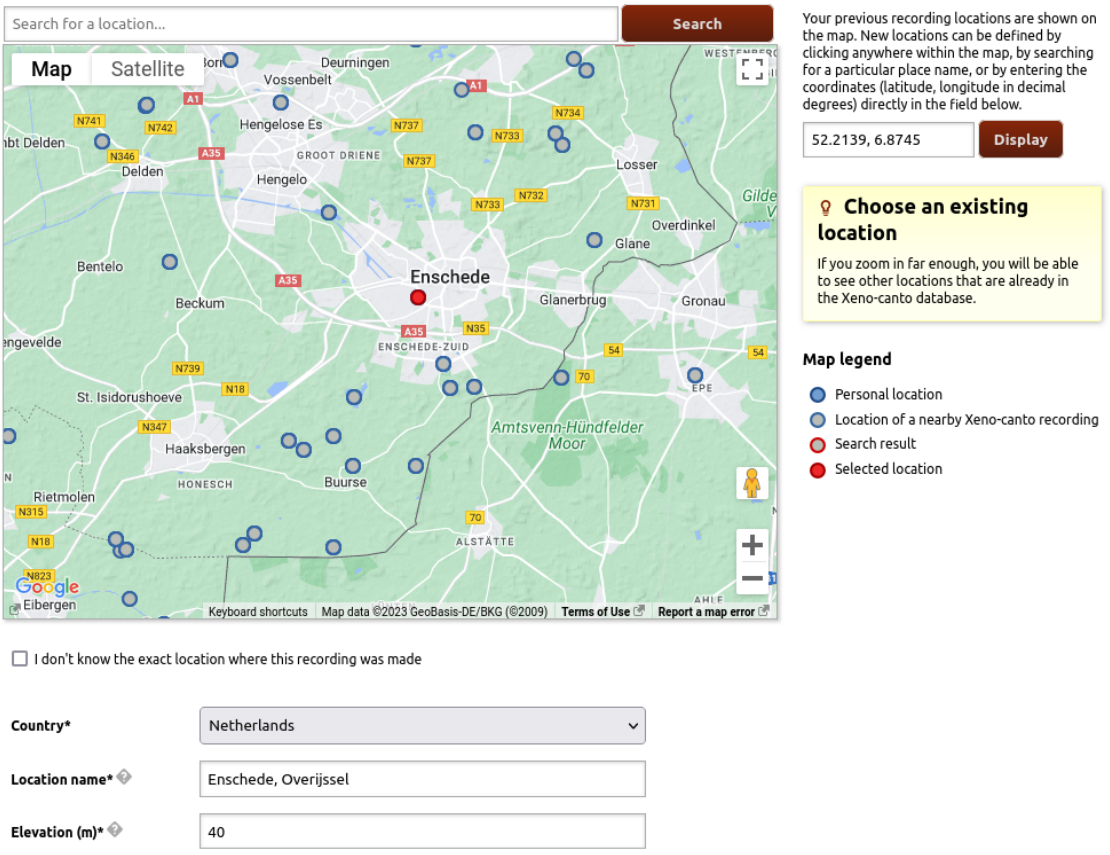


Figure 3.11: Demonstration of upload map type

the recording is in the expected region. However, the map does not provide a legend.

Mysteries map

This map type displays recordings with unknown species to seek help and contribute to identifying bird species. As a default function, it brings all the mystery recordings during the last 365 days. The map employs a clustering technique for its markers. The XC API function for pop-ups is also employed in this map type. Filtering with the XC API makes it possible to define and execute a more customized query.

User profile map

Each recordist has her/his own page that contains information about recordings made. In this section, the map shows the recordings of a specific user. It brings recordings and displays with marker clusters. This type of map also utilizes the XC API for generating pop-ups. Filtering with the XC API makes it possible to make a more customized query.

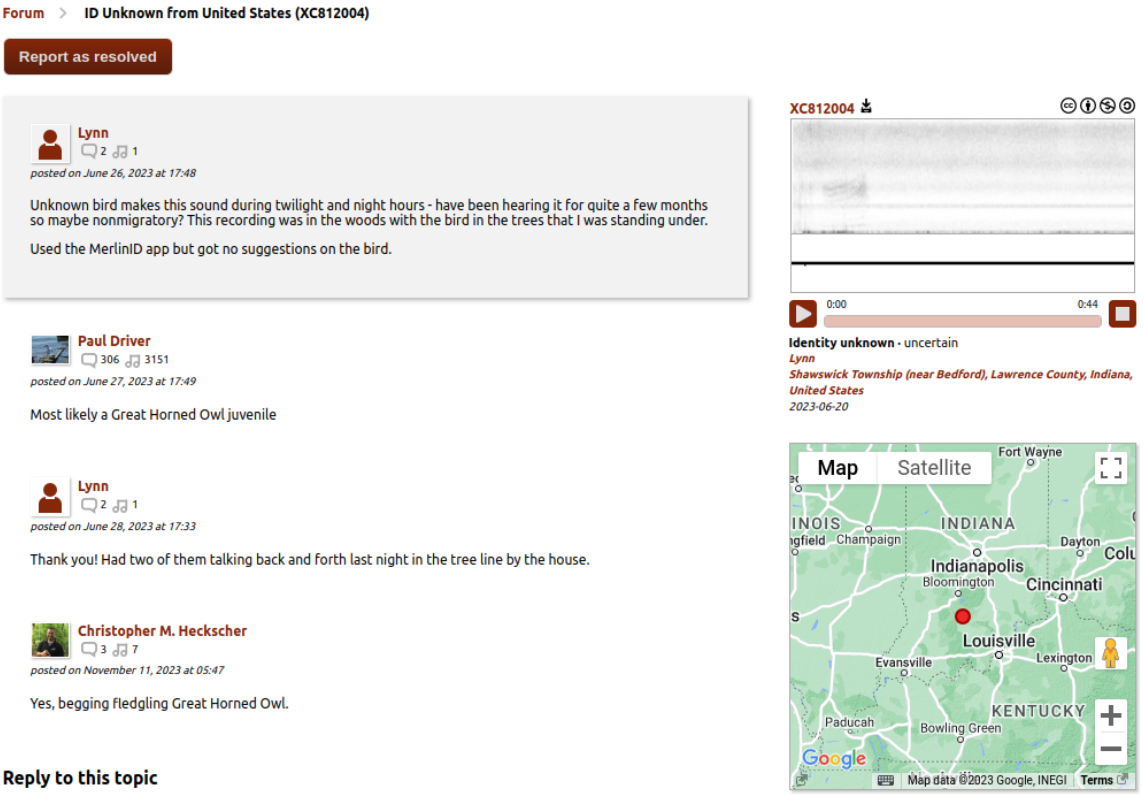


Figure 3.12: Demonstration of Forum: discussion about unknown species

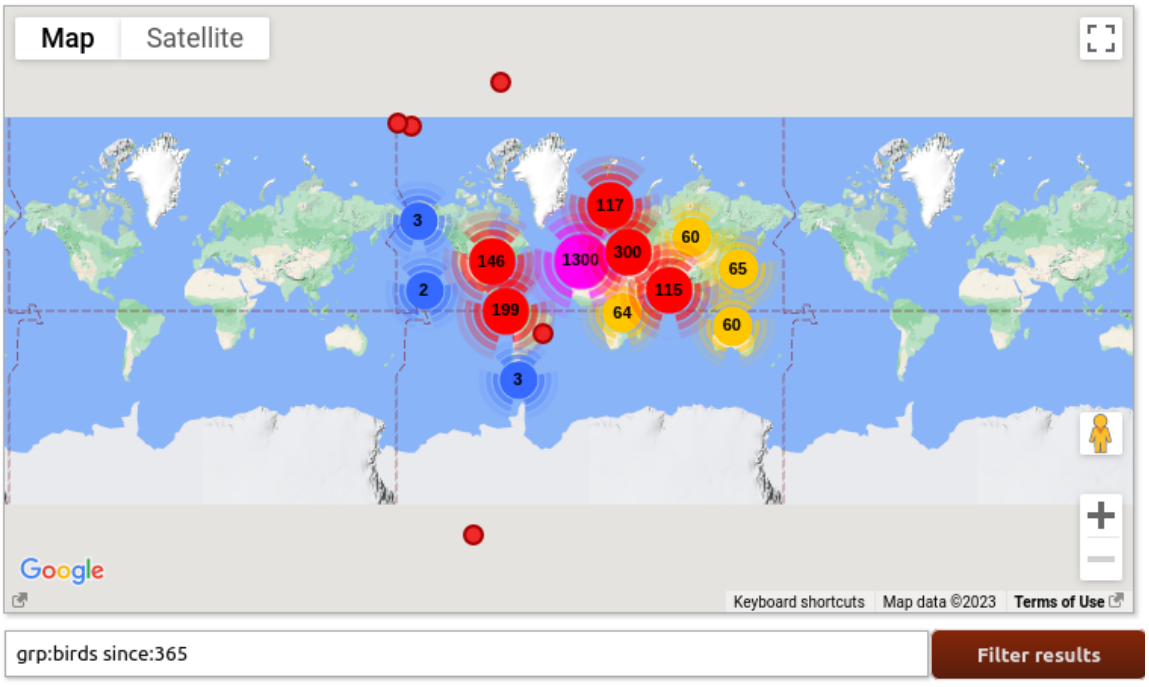


Figure 3.13: Demonstration of mysteries map type

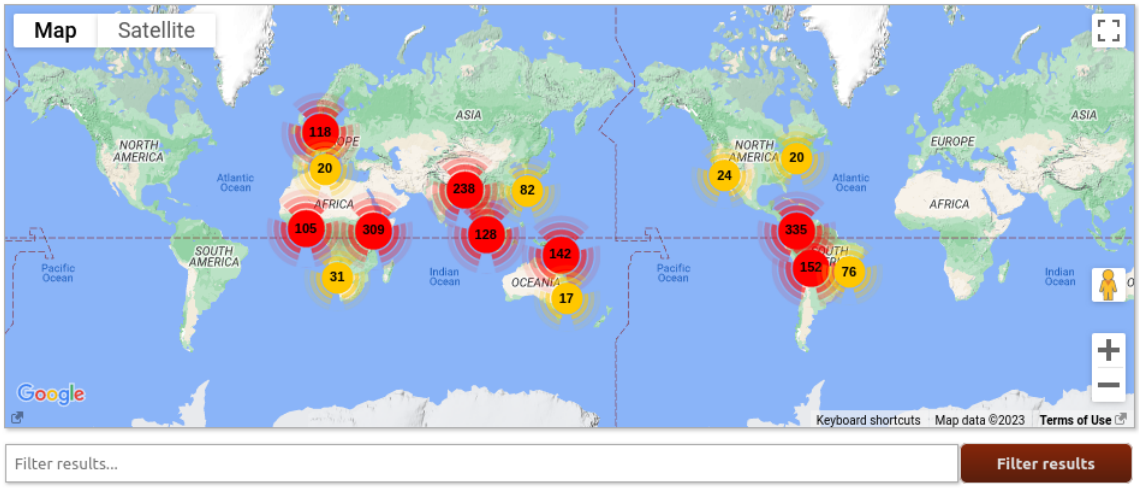


Figure 3.14: Demonstration of user profile map type

3.4 STRUCTURE STAGE: ANALYSIS OF THE CODE BASE OF XC

In the UCD process, as described in the book *Elements of User Experience* (Garrett, 2002), the structure stage involves organizing the components of the system. In this section, we conduct a deeper analysis of the code base of web map applications and APIs. A reverse engineering approach is applied to the XC codebase to understand its functions and remove dependency on Google Maps technology.

The code base for Xeno-canto is hosted on GitLab[6] and operates within a Docker environment. As of the latest version, which is named *develop*, the GitLab repository contains all the necessary files and documentation to set up the development environment.

The XC code base consists of 171 PHP files. These PHP files handle various aspects of the web page and map applications, including the generation of HTML code, controlling web pages, and handling back-end operations. In this study, we only focus on the files that are needed for the generation and use of web mapping applications. Figure 3.15 shows the specific PHP files we have modified.

3.4.1 Files that generate core map features

There are core map features that are responsible for generating HTML web pages and map content. By stating *core map features*, we refer to the key elements that users interact with when exploring maps. These components encompass the underlying code necessary for generating HTML pages, integrating CSS and JavaScript files into the web interface, and facilitating API calls.

[6] https://gitlab.com/naturalis/bii/xenocanto/xeno-canto

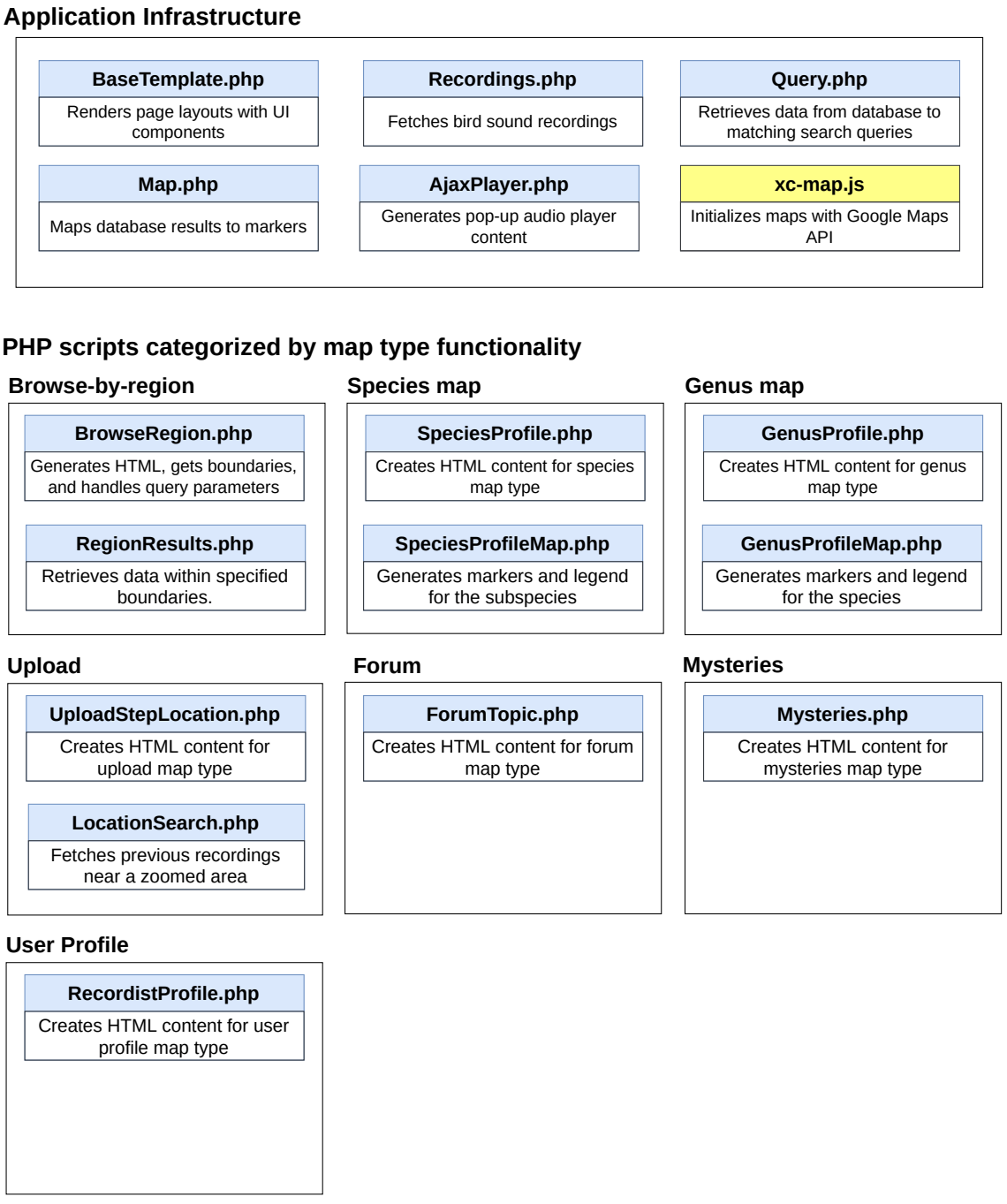


Figure 3.15: Overview of PHP scripts for application infrastructure and for each map type

BaseTemplate.php script

The code of `BaseTemplate.php` generates the overall structure of an XC web page. It takes content and parameters as input and constructs the complete HTML page using the necessary CSS tags. It also generates the header of the web page, including the site's logo, quick search bar, and main menu. Moreover, it includes footer information, such as links to social media, language options, and copyright details. Lastly, It includes JavaScript files required for various functionalities, such as jQuery (for handling dynamic interaction and various UI elements) and jPlayer (for audio player).

Map.php script

The code of `Map.php` is responsible for processing database results into map markers with specifications and providing the necessary JavaScript code for map functionality and interactions. While this file manages map-related tasks, it does not directly perform database queries; instead, its role involves processing database results and converting them into map markers. It structures database results in a format that can be used to create markers via JavaScript on the client side.

Recordings.php script

Starting in 2013, XC offered an extensive RESTful API that uses JSON. It is named Xeno-canto API v2[7] and enables users to filter their searches using over 37 parameters. Its primary purpose is to provide functionality for searching and retrieving bird sound recordings from the XC database through a RESTful API. It's called when users make API requests to search for recordings. An example of a query API call for adult male recordings in the Netherlands in the month of June for instance is: https://xeno-canto.org/api/2/recordings?query=cnt:netherlands+sex:male+stage:adult+month:6

`Recordings.php` interacts with the database to fetch and respond to bird sound recordings. The code PHP file defines three functions. First, the search function checks for the user-generated queries and validates the page number against the total number of result pages available. It performs a query execution based on the given parameters and handles possible database errors when an API request for recordings is received. The *query* refers to the user's search request, which the function uses to ask the database for specific recordings matching the user's criteria. The results of these queries are encoded as JSON objects and form the basis of the response, providing users with essential information such as the total count of recordings, species count, pagination details, and an array containing recordings.

Secondly, the recordingObject function assigns values to 37 parameters, including links for different sonograms, and provides information about the license of the recording. The function's purpose is to structure the recording data for the response. A parameter, in this context, refers to a query filter, while its value corresponds to the specific criteria users wish to apply. This function is called when constructing the response object for a specific bird sound recording. It organizes the data in a structured format for presentation to users.

[7] https://xeno-canto.org/explore/api

Lastly, the showRecording function finds a recording using its unique *xcid* to retrieve detailed information about a single bird recording. When the recording is located, it provides its details. When the recording is found in the database, the function provides details, including both metadata about the recording and links to the audio file. When not, it indicates that the recording wasn't found.

The 37 parameters that can be queried are shown in Table 3.6. These parameters are used as query criteria when users search for bird sound recordings using the XC API. They are not created by the functions; rather, they are predefined attributes that users can use to filter their searches.

Table 3.6 Parameters for filtering sound recordings via the Xeno-Canto API

Paramethers	Description	Paramethers	Description
id	the catalogue number of the recording on xeno-canto	**file-name**	the original file name of the audio file
gen	the generic name of the species	**sono**	the urls to the four versions of sonograms
sp	the specific name of the species	**osci**	the urls to the three versions of oscillograms
ssp	the subspecies name	**lic**	the URL describing the license of this recording
group	the group to which the species belongs	**q**	the quality rating for the recording
en	the English name of the species	**length**	the length of the recording (minutes:seconds)
rec	the name of the recordist	**time**	the time of day that the recording was made
cnt	the country where the recording was made	**date**	the date that the recording was made
loc	the name of the locality	**uploaded**	the date that the recording was uploaded to xeno-canto
lat	the latitude of the recording in decimal coordinates	**also**	an array with the identified background species
lng	the longitude of the recording in decimal coordinates	**rmk**	additional remarks by the recordist
type	the sound type of the recording (call, song, etc)	**bird-seen**	indicates whether bird seen or not
sex	the sex of the animal	**animal-seen**	indicates whether animal seen or not
stage	the life stage of the animal (adult, juvenile, etc.)	**playback-used**	indicates whether playback used or not
method	the recording method (field recording, in the hand, etc.)	**temperature**	temperature during recording
url	the URL specifying the details of this recording	**regnr**	registration number of specimen
file	the URL to the audio file	**auto**	automatic (non-supervised) recording
dvc	recording device used	**smp**	sample rate (Hz)
mic	microphone used		

For greater flexibility, there are five additional parameters. The *Area* parameter facilitates geographical searches, permitting users to focus on specific regions such as Africa, America, Asia, Australia, and Europe. By utilizing the *since* parameter, users can discover recordings uploaded after a given date, with support for both integer values (e.g., since:3 for the past three days) and dates in YYYY-MM-DD format (For example, 2023-09-16). The *year* and *month* parameters enable searches for recordings created on particular dates, even accommodating $>$ (after) and $<$ (before) operators. Developers have stated that they will not change or delete the current response properties to guarantee compatibility with earlier versions. By *existing response properties*, we refer to the specific attributes and features within the software that are currently in use and will remain unchanged.

AjaxPlayer.php script

`AjaxPlayer.php` is responsible for handling AJAX (Asynchronous JavaScript and XML) requests for the media player. When a request is made to this endpoint with a given nr, it returns an HTML snippet that embeds a player for that recording. This HTML code creates the pop-up content.

Query.php script

This PHP code is designed to handle complex queries on data. Its primary function is to create and execute structured queries against a database, retrieving information based on search criteria.

This file actually queries the database. Other PHP files set up these requests, but it's actually the `Query.php` script that retrieves the information from the database.

xc-map.js script

This JavaScript code provides a comprehensive framework for integrating and customizing the Google Maps API. It facilitates the creation and management of map instances, markers, and information windows, with special features for handling media playback and marker clustering. It manages multimedia pop-up content for each marker, ensuring users can access the pop-up content generated by `AjaxPlayer.PHP`. This code also incorporates KML layers into the map to display species ranges. The marker clustering is also performed in this code.

3.4.2 Code-wise analysis map types: Browse-by-region, genus, species, subspecies and upload map types

Browse-by-region map

Two PHP files are used to generate the `browse-by-region` map type: `RegionResults.php` and `BrowseRegion.php`. The HTML code for this page, the division for *map-canvas*, and the JavaScript code for getting bounds and making a query on the basis of API calls are generated by `BrowseRegion.php`. The code defines the *queryBounds(bnds, query)* function. This function calls the API using the corner coordinates of the bounding box drawn by the user and displays the results on Google Maps. Once the region is chosen, recordings in that area will be shown on the map with a marker clustering technique.

The color of the markers corresponds to the frequency of recordings: blue for 2 to 9 recordings, yellow for 10 to 99 recordings, and red for more than 99 recordings. Different levels of markers are demonstrated in Figure 3.16.

Markers of single recordings are a single red dot. Notably, XC developers have set a limit to show only the first 2,000 recordings on the map.

An API call for this map type is generated by the api/RegionResults.php.[8] `RegionResults.php` defines a public function get() which is responsible for processing HTTP GET requests. It is responsible for fetching and summarizing the XC database based on a set of coordinates and an optional query filter. This function takes five input parameters: yn, ys, xw, xe, and query. These parameters represent the latitude and longitude boundaries of a bounding box (yn: latitude max, ys: latitude min, xw: longitude min, xe: longitude max) and a search query for filtering the results. Finally, the function compiles a summary of the results, including marker specifications, markers information, and the summary of the results. The responses are formatted as JSON.

For example, to make a query on the *Enschede* region, the API URL is shown as: `https://xeno-canto.org/api/internal/region-results?yn=52.269&xe=6.713&ys=51.653&xw=5.663&query=`

In this example, we are searching for bird recordings within the specified geographical coordinates (Enschede area). This query will return recordings that match within the defined geograph-

[8] `https://gitlab.com/naturalis/bii/xenocanto/xeno-canto/-/blob/develop/xc/Controllers/api/RegionResults.php`

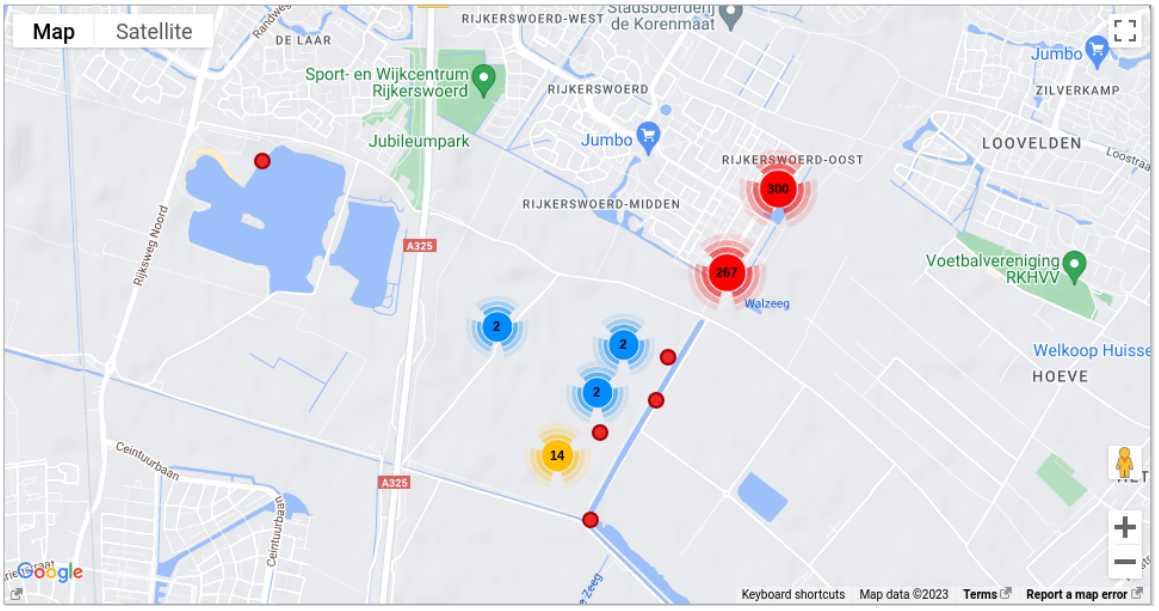


Figure 3.16: Demonstration of different frequency of markers

ical region. The output is returned in JSON format, and it gives three results, which are *specs* that define the marker icon and size, *markers* holding information that includes coordinates and names, and *summary* that provides the total number of species and recordings.

Below the map, query functionality is provided with the search bar. Through this, users can filter results by additional query parameters such as country, coordinates, altitude, type of call, sex, quality, length, date, and time.

Genus map

`GenusProfile.php` creates a `<div>` with the id of *map-canvas* to be the container for Google Maps. It accesses the database to get taxonomic details of the genus. It brings up details such as the genus' associated order and family. `GenusProfile.php`'s main task is to create a visual map that associates bird sound recordings with the genus they belong to. The *getLegendVfunc* function creates the description of the map. It creates an HTML list detailing each species within the genus, complete with common and scientific names and links to relevant profiles.

Species and subspecies map types

`Species` and `subspecies` map types are generated in the same code. Selecting subspecies in the `species` map triggers the display of a `subspecies` map. This map type focuses specifically on subspecies within a broader taxonomic category.

`SpeciesProfile.php` creates a * * with the id of map-canvas to be the container for Google Maps. API call for this map type is generated by the `Recordings.php`. `SpeciesProfileMap.php` is focused on visualizing bird sound recordings on the `subspecies` map. It plays a role in distin-

guishing records according to their taxonomic classification, displaying different markers for foreground species, background species, and those not designated to a particular subspecies. When a user accesses a species profile, this script is called to manage the map's description via the *getLegendVfunc* function to generate the legend. This function creates a custom HTML legend by arranging pointers, subspecies names, and taxonomic authors in alphabetical order.

Upload map

`UploadStepLocation.php` creates a `<div>` with the id of map-canvas to be the container for Google Maps. When users get a higher zoom level than 6, they can see the recordings that have been recorded in that region before. `LocationSearch.php` responsible for fetching these recordings' data.

Address information is obtained based on its latitude and longitude coordinates using the Google Maps Geocoder service. The code first extracts the country and its short name (country code) from the geocoding results using the getAddressComponentType function. Next, the code checks for more specific location details. It attempts to find a locality that often represents the city or town. If a locality is available, it is stored in the locality variable. Then, the code searches for specific types of location details like natural features, establishments, and parks in the geocoding results. If none of these specific types are found, it resorts to using an administrative area level 3 component. This administrative level typically refers to a subdivision of a geographical area that is smaller than a state or province but larger than a locality or city. Additionally, the code checks for and appends administrative area level 2 (county) and administrative area level 1 (state) if present.

3.5 SKELETON STAGE: FORWARD ENGINEERING APPROACH

Within the scope of this project, referring to the insights provided in *Elements of User Experience* (Garrett, 2002), the skeleton phase is characterized by the application of an advanced engineering approach. This involves the practical application of the knowledge gained from the previous stages to the actual development and improvement of the user interface for the XC project. This section describes the implementation of the forward engineering approach. Results are discussed in Section 4.3.

3.5.1 Introduction

The XC platform makes use of eight different map types. Our aim is to leverage the strengths of these map types to maximize user experience and data exploration capabilities. However, given the timeline of this project, it was not feasible to build a different web map application for each. Therefore, the scope of this project is narrowed to cover five map types, such as:

1. Browse region
2. Genus
3. Species

4. Subspecies
5. Upload

The reasons for choosing these map types include user preferences and the functions that the maps have. The results of the questionnaire indicate that users have strong preferences for these map types. Additionally, they offer more data exploration opportunities compared to other map types. Due to their popularity and functionality, our study prioritized these map types. In particular, we made special efforts to create the `species` map type. We think it is the most important map type because it provides visual and spatial context to sound data, allowing users to understand the geographic distribution of bird species. This is important for studies on bird behavior, distribution, migration patterns, conversation, and biodiversity assessments.

3.5.2 Map core configuration and customization

The lightweight nature of Leaflet allows for easy integration of necessary plug-ins. Following the current architectural framework, we placed the JavaScript HTML tag links of the plug-ins in the `Map.php` script. The CSS files of these plug-ins are integrated into `BaseTemplate.php`.

In the initial configuration and customization of the maps, plug-ins that provide additional functionality and aesthetic improvements were used to address the specific needs of the maps. Each plug-in has a unique role in the map's functionality. Firstly, Leaflet is the fundamental library for building interactive maps. The *L.fullscreen* plug-in is integrated to allow for a full-screen map view. *L.switchBasemap* is used to enhance the user interface by offering a visually appealing method to toggle between basemaps. Enhancements such as clustering markers and providing a drawing bounding box with a button for *browse-by-region* are achieved with the *L.markercluster* and *Leaflet.draw* plug-ins. The *leaflet-control-geocoder* plug-in is used to add a feature to let users search for places and addresses right on the map. Additionally, to optimize performance and efficiency, our GeoJSON files are stored in a compressed zip format. To extract the contents of these zipped GeoJSON files, the *JSZIP* plug-in is used. *Chroma.js* is utilized to arrange distinct colors to genus level species ranges in Genus map. Lastly, we created a separate JavaScript file for each of the map types. HTML tags of these JavaScript files and CSS files are also included in the header. The plug-ins used are shown in Table 3.7.

Table 3.7 Description of Leaflet plug-ins used in map customization

Plug-in	**Description**
Leaflet[9]	Core library for building interactive maps
Leaflet.fullscreen[10]	Allows for a full-screen map view
L.switchBasemap[11]	Allows users to toggle between different basemaps
Leaflet.markercluster[12]	Clusters markers
Leaflet.draw[13]	Enables drawing and editing of vector layers on the map
Leaflet-control-geocoder[14]	Adds a search function to find places on the map
JSZIP[15]	Handles the decompression of GeoJSON files
Chroma.js[16]	Makes color conversions and color scales

In configuring the Leaflet map, we first initialize the map instance by setting its view to coordinates [35, 15] and a zoom level of 2, which provides a global perspective. This initialization is

applied to our HTML element.

In Leaflet, when users zoom out on the map, they will see multiple copies of the world map next to each other. By setting the *worldCopyJump* feature to true, we ensure that the basemaps appear continuous. Instead of reaching an edge, users seamlessly transition from one side of the map to the other. The northern and southern borders of the map are defined as borders. To prevent users from exceeding these limits, the *maxBounds* variable is set to [-90, -Infinity], [90, Infinity]. This arrangement prevents moving outside the boundaries of the map when using the zoom and/or pan functions and keeps users always engaged with the map.

In Leaflet, the default position for the zoom control buttons is the upper left corner. In the current maps, however, those buttons are placed in the bottom right corner. To maintain the familiarity of XC users, we place the zoom control buttons in the bottom right corner. In the same manner, the full-screen button is placed in the top right position of the map. Finally, even though current maps do not provide a scale, we set a scale in the bottom right. To support the global usage of XC, both metric and imperial units are shown.

In the current XC platform, users have the ability to zoom in and out using zoom buttons or by double-clicking. Responses to the questionnaire, however, revealed a demand for the ability to zoom using the mouse wheel. The mouse wheel zoom is particularly important as it provides a quick and intuitive way to control the map view.

To improve user experience, we provide users with a visually appealing basemap switch button to change between basemaps. Additionally, as mentioned in Section 3.3.2, we increased the number of basemaps offered. We selected five diverse basemaps: Terrain, OpenTopo, Satellite, OSM, and GBIF. Each basemap offers unique visual features that cater to different user needs and preferences. The WorldStreetMap[17] and WorldTopoMap[18] basemaps present a comprehensive street map, providing detailed information on roads and key landmarks. They provide a rich representation of the Earth's surface, including information on roads, key landmarks, and topographical variations. Additionally, WorldTopoMap provides elevation information. Satellite[19] basemap provides an aerial view of the earth's surface, allowing users to explore bird habitats with real-world visualization. The OpenStreetMap[20] basemap provides detailed maps, supporting users in precise navigation and location identification. Finally, the GBIF- Geyser[21] basemap has been selected for its minimalist and calm gray/white background. It has advantages for users who prefer not to have a visually cluttered basemap. Figure 3.17 illustrates our customized map interface, showcasing the places for zoom control, full-screen buttons, scale, and switch map function.

3.5.3 Map types

Browse-by-region map

The file hosts the HTML code necessary to generate the `browse-by-region` page. This file holds the code for generating `div-id` for the HTML content.

[17] https://www.arcgis.com/home/item.html?id=3b93337983e9436f8db950e38a8629af
[18] https://www.arcgis.com/home/item.html?id=6e850093c837475e8c23d905ac43b7d0
[19] https://www.arcgis.com/home/item.html?id=10df2279f9684e4a9f6a7f08febac2a9
[20] https://operations.osmfoundation.org/
[21] https://tile.gbif.org/ui/

Figure 3.17: Demonstration of the customized map interface

The current `browse-by-region` map type allows users to draw a bounding box only via a shift button. Respondents to the questionnaire, however, indicated that this function leads to poor user experience, especially on mobile devices. To enhance this, an additional button was added to the map to draw a bounding box. This innovation offers a more useful interface available to desktop and mobile web browsers.

On the current map, markers are shown by a clustering method depending on zoom level. This approach provides an overall visualization of the data at lower zoom levels. While this broad approach is effective for a high-level overview, it hides specific details of individual records within these clusters. Additionally, users can see the number of recordings in a region, but they cannot get information about the diversity of species represented in those recordings. To address these issues, a grid-based system was developed. This gives users the option to view the number of records or the number of species. Users can toggle between these two variables. In this way, users can detect regions with high species diversity more effectively.

The grid dimensions are set at 111 km for zoom levels up to 6, reducing to 20 km grids for closer views. This decision was influenced by an extensive study of existing web mapping applications, primarily eBird[22]. eBird uses a grid size of 100 km for its global view and 20 km for larger zoom levels (Zhang, 2020; Kelling et al., 2012). In our case, we chose 111 km for the global view—a conscious decision to adapt to geographical measurement where 1 degree equals approximately 111 km. For higher zoom levels, we set our grids to 20km. We also conduct experiments to find out the optimum solution. Our findings showed that 20km provides an ideal balance, offering detailed data presentation without overwhelming the user interface. After zoom level 9, recordings are shown by utilizing clustered markers.

Yet, this approach also has limitations. The first issue with this map type is that the API implementation only offers the first 2,000 recordings. The involved PHP file is `RegionResults.php`.

[22] https://ebird.org/hotspots

The grid-based presentation approach is, however, designed to provide an overview of large areas and will yield more useful results with a high number of recordings. Another issue is that while our initial design defines the grids as squares, the inherent characteristics of the Web Mercator projection introduce distortions. As discussed in Section 3.3.2, this projection is not an equal-area projection. Consequently, our square grids tend to stretch into rectangles in higher or lower latitudes.

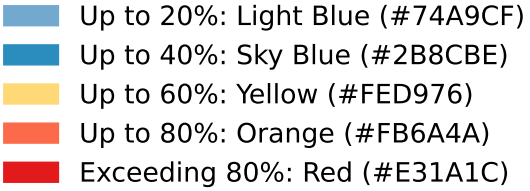


Figure 3.18: Demonstrating data intensity levels from light blue to red

In the visualization of grids, a dynamic color coding scheme is used to represent data variations effectively. Thresholds are determined based on the maximum count value, establishing five distinct grades to categorize the data. Each category is assigned a specific color, which calculates the ratio of individual counts to the maximum count. Ratios up to 20% are denoted with light blue (#74A9CF), up to 40% with sky blue (#2B8CBE), up to 60% with yellow (#FED976), up to 80% with orange (#FB6A4A), and values exceeding 80% are marked in red (#E31A1C). Colors are chosen to be distinct from one another and ensure clear separation. The light lavender (#74A9CF) denotes the lower end of the scale, offering a calm and neutral start. As the scale progresses, the colors become more intense, culminating in red (#E31A1C), which is often associated with alertness and importance (Won and Westland, 2017), highlighting areas of high value. Figure 3.18 shows the color gradients from light blue to red.

For marker clustering, we follow the existing approach, which uses three different thresholds. A single red circle marker for a single recording, blue for two to nine recordings, yellow for 10 to 99 recordings, and red for more than 99 recordings. Additionally, users can place their mouse cursor or click on the grid cell, and it shows the number of recordings and species in the pop-up.

Browse-by-taxonomy: Genus, species, and subspecies map types

In this section, we first discuss a newly developed color coding scheme, and the conversion of species ranges from KMZ to GeoJSON.zip format is discussed. Subsequently, implementations for `browse-by-region`, `genus`, `species` and `subspecies` map types are explained.

Development of dynamic color coding scheme for markers As explained in Section 3.3.2, the current coding system assign the list of static icons as markers. This, rather arbitrary method, does not account for the rich biological relationships between the subspecies. To enhance the accuracy and educational value of XC, we implemented a dynamic marker color coding system that reflects, to the extent possible, taxonomic sequence according to the International Ornithological Congress (IOC) World Bird List ("IOC World Bird List – Version 13.1", 2023). (This is the taxonomy that XC adopts for its species list.) The new color scheme is designed to visually represent

the relationships between subspecies—the closer the relationship, the more similar the colors of the markers.

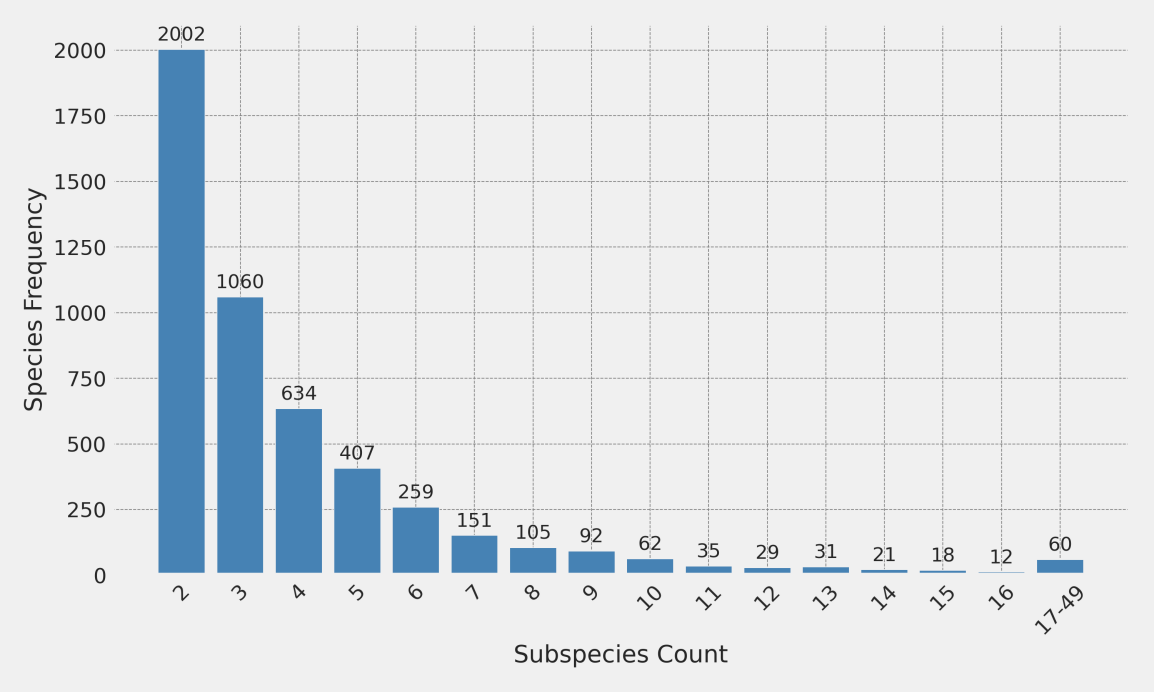


Figure 3.19: Histogram showing the distribution of species by number of subspecies. The x-axis represents the number of subspecies and the y-axis represents the frequency of species. Species with 17-49 subspecies are grouped together.

It is important to choose an appropriate threshold when adjusting colors for visual representation. This threshold determines the strategy used, whether it represents a broad spectrum of many colors or a more limited palette. Insights from the XC database indicate variability in the number of subspecies among species, and as Figure 3.19 illustrates, the general trend among bird species leans toward a lesser number of subspecies. In light of this trend, the decision has been made to design a color coding schema that can effectively differentiate between species using a minimum of 16 distinct colors.

During the initial design phase, a color gradient between 0 and 180° was considered to use maximal contrast between colors for the first and last element. We judged, however, that such a color range did not provide sufficient color diversity for cases with a high number of subspecies. In a second approach, a 240° portion of the color wheel (see Figure 3.20) was chosen to ensure sufficient color diversity for cases also with a high number of subspecies, ranging from red to dark blue (T. Lin et al., 2015). When we next applied a homogeneous distribution over this color scale, it was not easy to distinguish between all the generated colors. Human color perception is nonlinear along this scale. This is why we chose seven base colors from which we derived the marker colors used by interpolating between them. In the application phase, we observed that red hues are distinguished more easily than green hues. To produce the maximum number of distinguishable colors, the brown color was added between red and orange in the seven base colors. When a high number of marker colors is generated, a number of them will be closer to red and will be easier to distinguish. The selected seven base colors are red, brown, orange, yellow, green, blue, and dark blue. These base colors, their RGB and HSL values, and their corresponding position on the color

wheel are shown in Figure 3.20.

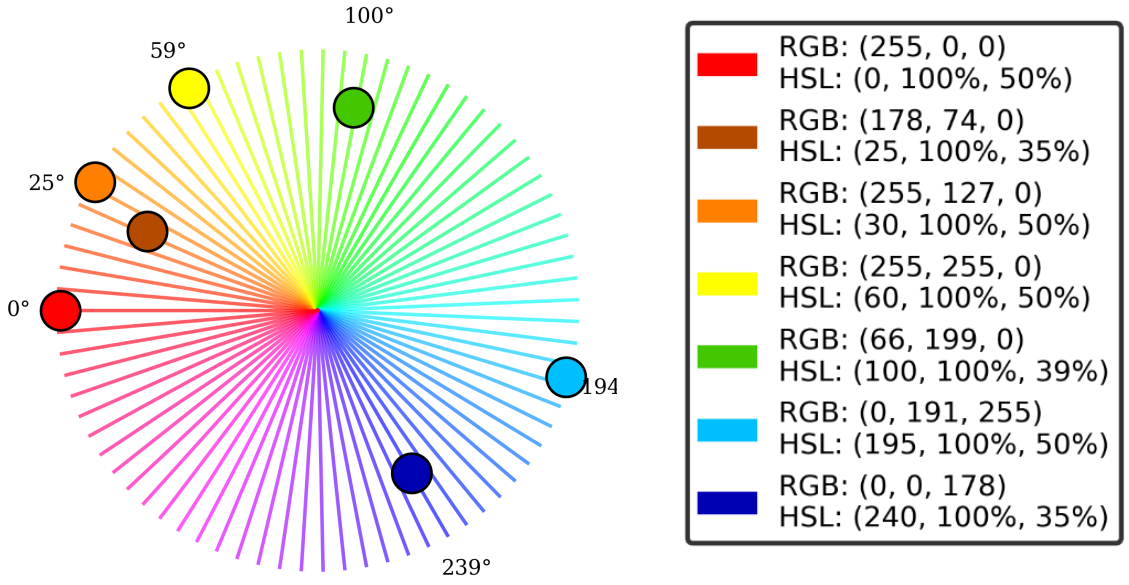


Figure 3.20: Illustration of seven base colors and their RGB/HSL values and the corresponding position on the color wheel

The function is designed to start with red for the first subspecies and to end with dark blue for the last subspecies. Linear interpolation of HSL is employed between color steps to generate continuous and graded transitions across the spectrum for each subspecies. This is achieved by mapping each subspecies' position on the IOC list to a specific point within a gradient of basic colors. To provide a clear understanding, Figure 3.21 shows the the seven base marker colors location on the color wheel.

We must remark here that the sequence of IOC comes with certain constraints. Firstly, while the IOC list orders bird species based on their DNA sequences, the taxonomic trees inherently bring with them only a partial order. This gives us several possible linear orders, and the ranking used by IOC is just one of these possibilities. Secondly, our taxonomic knowledge is not always complete, and the IOC listing introduces a number of assumptions. For example, ranking among species missing certain DNA information may have to be estimated by factors such as physical similarities or geographical proximity of habitats (McClure et al., 2020). This creates a ranking of taxa that is sometimes imprecise and subject to change. In other words, it is not always possible to make a complete and accurate ranking of species with the scientific information we have at present. Consequently, despite the limitations and uncertainties involved in the taxonomic classification process, the IOC list remains a valid and scientific resource for bird taxonomy.

Conversion from KMZ to GeoJSON.zip Moving away from Google map technology would prompt us to get rid of KMZ/KML files because those files are designed for Google's ecosystem, which we are trying to remove the dependency on. As a solution, we decided to use the GeoJSON format because it offers a more universal, open-standard format. It is widely accepted and easily used across various Geographic Information System (GIS) ans spatial database platforms

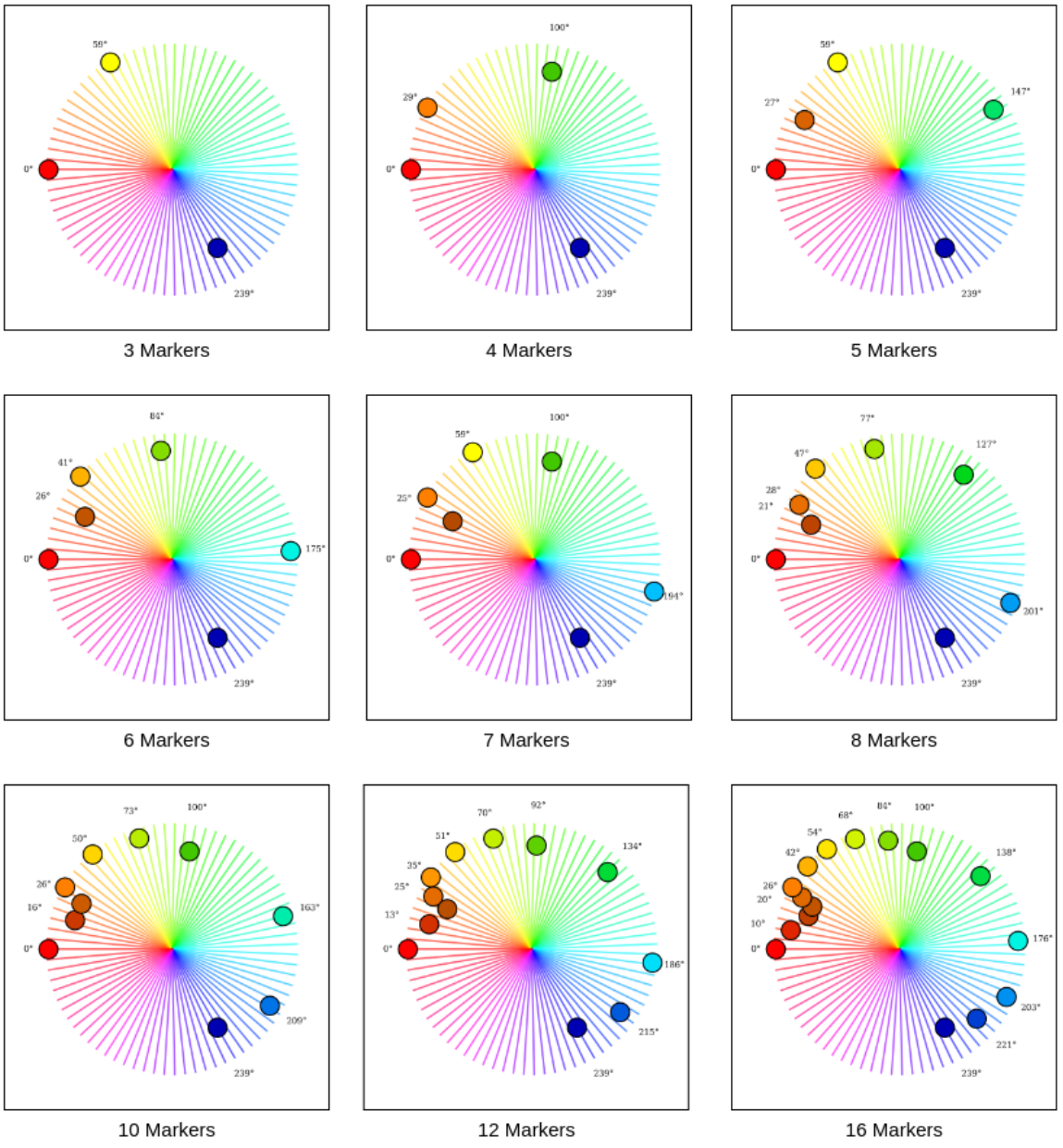


Figure 3.21: Illustration of colors and their corresponding positions on the color wheel

(Butler et al., 2016). Using GeoJSON encourages data sharing and interoperability, enabling XC to participate more actively in the open-source geospatial community.

Another issue is that in some cases, the KMZ configurations are not rendered correctly by Google, leading to incorrect representations of species ranges on the maps. For instance, there have been issues with the plot of the range of *Corvus corax* in current XC maps. Google technology over the years has changed data requirements, leading to map visualization instabilities. The use of GeoJSON files mitigates such conflicts, promoting the accuracy and reliability of species range visualizations in our maps. This approach enables us to bypass the inconsistencies and errors associated with KMZ configurations, ensuring a more dependable presentation of geographical

data. Figure 3.22 shows an example of one KMZ rendering error.

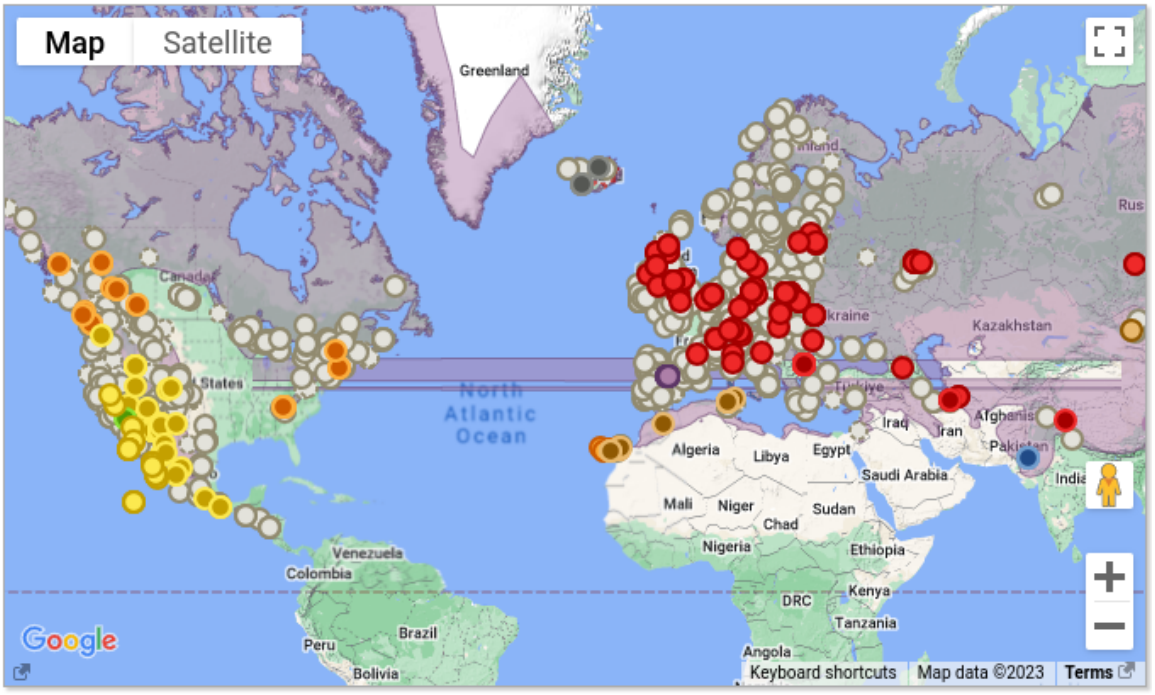


Figure 3.22: Current map showing the error on the species ranges of *Corvus corax*

Genus map The map `<div>` for genus map type is placed by the `GenusProfile.php`. `GenusProfileMap.php` file is responsible for generating the legend.

As explained in Section 3.5.3, we have implemented a color coding schema instead of static icons to provide a visual correlation between species' taxonomic position. We also added two buttons at the bottom of the map interface. They are designed to toggle the visibility of markers and species ranges. With these, users gain the flexibility to customize their map view by showing or hiding markers and species ranges as they wish.

The current genus map type provides only markers of recordings, and it does not show species ranges. In the legend, species are ordered; however, since coloring is based on the static image order, there is no correlation between colors and species. To enhance the visualization of species markers on the genus map, we have employed our newly developed dynamic coloring approach. In this strategy, markers representing species with closer taxonomic affinities are assigned colors that are visually alike, promoting a more intuitive understanding of species similarities and differences.

Additionally, the XC platform provides KMZ files that represent the species ranges for each species. We used these species ranges to obtain a single genus range in such a way that every range part was labeled with the number of species occurring in that range part that belongs to the genus. Next, we output these results as a GeoJSON file. After compressing these GeoJSON files, we incorporated them to display diverse areas on the genus map. In visualizing species ranges, colors are assigned based on species count within each range. The Chroma plug-in facilitates this and assigns various shades of purple to represent different species counts. Purple was chosen as it is distinct from the colors used for markers, preventing any visual confusion or overlap in colors

between markers and species ranges. Enhancing user interaction, the visualization allows users to click on specific areas, triggering a pop-up that provides information regarding the number of overlaying species ranges.

To address the issue of ambiguous search results due to similar but not identical genus names, we refined XC API by enclosing the genus parameter in quotes, ensuring that the query matches only with the exact genus name, thus eliminating unrelated results. This way, our results to plot markers only contain matching genus. Moreover, changes have been implemented in the `Query.php` script to get exact matching genus data. Previously, the script supported two conditions when querying the genus: a direct match or a starting with condition. Now, by eliminating the second one, only exactly matching genus names are queried. The following code snippet illustrates the query code to get two conditions.

```
} elseif ($tag == 'gen') {
    if ($tq->operator == $tq::MATCHES_OPERATOR) {
        $condition = $this->matchesCondition('birdsounds.
            genus', $term);
    } else {
        $condition = $this->startsWithCondition('birdsounds.
            genus', $term);
    }
```

Species map The `SpeciesProfile.php` script constructs the HTML code needed to generate the page, and `SpeciesProfileMap.php` holds code that generates a legend.

The current species and subspecies map types use a static color coding system to color markers of subspecies based on their alphabetic order. This method does not account for the rich biological relationships between the subspecies. To enhance the accuracy and educational value of XC, we have implemented a dynamic color coding system for the markers that reflects taxonomy sequence according to the International Ornithological Congress (IOC) World Bird List. The new color scheme is designed to visually represent the relationships between subspecies—–the closer the relationship, the more similar the colors of the markers.

To integrate this dynamic color coding system, we changed our database. The XC database contains the `taxonomy_ssp` table to retrieve taxonomic data. This table includes columns such as order, family, genus, species name and number, subspecies, and authorship. We added a new column called `ssp_seq` to this table to capture subspecies sequence within a species as found in the IOC List. This new column shows the ranking of subspecies. Then, we created an additional temporary table, `sp_ioc`, with three columns named species, subspecies, and `ssp_seq`, which was filled from a .csv file. By assigning the species and subspecies fields as primary keys, we synchronized this data with the data in the `taxonomy_ssp` table and updated the `ssp_seq` column. This step ensures that each subspecies is correctly ranked. When performing SQL queries on the `SpeciesProfileMap.PHP`, we defined the underlying query using *ORDER BY ssp_seq ASC* to follow the correct taxonomic order. Once the subspecies were sorted correctly, we created dynamic SVGs instead of fixed icons. This ensures consistency between markers in the legend and on the map.

Another issue is that the legend has a problem with subspecies names due to allowing arbitrary user input for this field. Users are free to write any subspecies name they want. Some users have inserted a question mark or a separate note at the end of subspecies names when they are uncertain.

This situation causes complexity. To simplify the process and minimize complexity, we now cross-reference the name of each subspecies with the IOC list. If a name does not match an entry in the list, we classify it as an *unspecified subspecies* and represent the recording with a gray color marker.

Assigning colors to markers based on taxonomic vicinity constitutes an important component of this map type. For this purpose, each identified species and subspecies are ranked using the IOC list. IOC provides Excel files for their data[23], but the Excel file format is not feasible to work with JavaScript. To make a seamless integration with JS, we decided to create a JSON file based on the IOC list. The structure of the created IOC.json file is shown in Figure 3.23.

```
"Genus1": {
    "Species1": ["Subspecies1", "Subspecies2", "..."],
    "Species2": ["Subspecies1", "Subspecies2", "..."]
},
"Genus2": {
    "Species1": ["Subspecies1", "Subspecies2", "..."]
},
"Genus3": {
    "Species1": ["Subspecies1", "Subspecies2", "..."],
    "Species2": [],
    "Species3": ["Subspecies1", "Subspecies2", "..."]
}
```

Figure 3.23: Example of the adopted JSON structure

Additionally, we introduced two buttons below our map container to toggle markers and species ranges. In this way, users can show and hide the view of the markers or ranges and can customize their map interface according to wish.

The complexity of geographical locations is considered for the species map type. There are cases where multiple recordings exist in the same location. In the current map, these markers apply a single color. For example, when two subspecies are recorded in the same location, the current map shows only one color, and users cannot know whether another subspecies recording exists until they click on the marker. As a solution to this, we grouped our recordings based on coordinates. When multiple recordings exist for a location, we now use pie chart markers. This provides users with immediately visible information about the diversity in a location without clicking. Figure 3.24 illustrates one of the pie chart markers. This issue generally arises between specified and non-specified subspecies. As the XC dataset continues to expand, it is possible that a similar issue will emerge among specified subspecies as well. Implementing the pie-chart marker approach will prevent potential conflicts that may occur in the future as users continue uploading recordings.

Another consideration is that not all species have subspecies. For example, in Figure 3.23 Genus3-Species2 has no subspecies, and it is a monotypic species. For quite a number of species, no specific subspecies have been described or are considered valid by IOC. They are consequently called monotypic species. In these cases, the current species map shows the markers of these species in gray (not specified). However, in our study, monotypic species are not ignored in the coloring and classification. If a species is monotypic, then subspecies information is shown as monotypic.

[23] https://www.worldbirdnames.org/new/ioc-lists/master-list-2/

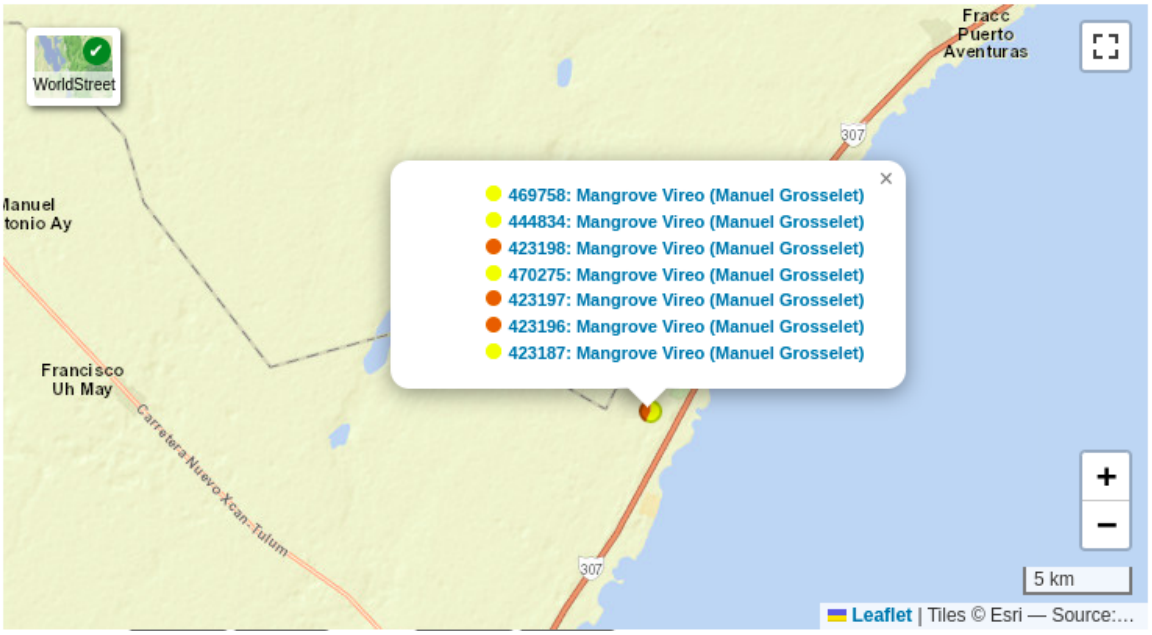


Figure 3.24: Illustration of a marker as a pie chart

Figure 3.25 shows the example for the species *Ibidorhyncha struthersii*.

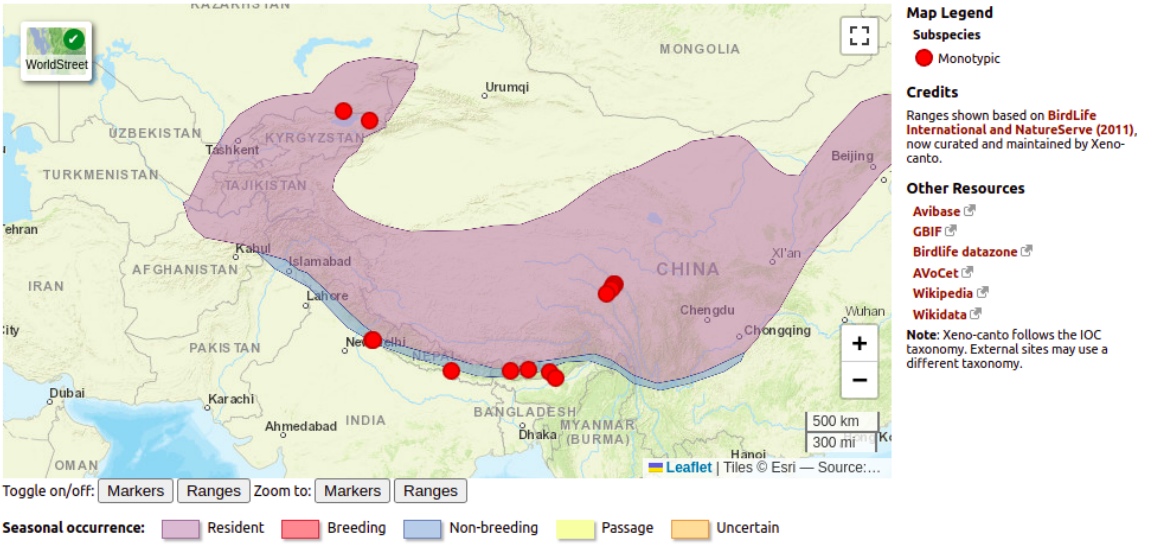


Figure 3.25: Illustration of a monotypic species

We also attempted to provide an alternative projection to Web Mercator for species that occur predominantly in polar regions. The Proj4.js plug-in facilitates the necessary coordinate transformations, but we encountered difficulties finding a functional basemap web tile to serve below the reprojected range maps. We initially tried to use the Polarmap.js[24] plug-in to implement polar region web tiles provided by the ArcticConnect[25] project. However, support for these web tiles was

[24] https://github.com/GeoSensorWebLab/polarmap.js
[25] http://arcticconnect.ca/

discontinued during our project. As a result, we could not find a suitable alternative network tiling for the polar regions. Although GBIF[26] (Global Biodiversity Information Facility) provides a web tiling service, we encountered significant errors with that basemap's panning and zooming functions. These issues made the service inadequate for our purposes. To solve this problem, it may be necessary to create and provide XC's own web tiles for the polar regions. The figure 3.26 shows our initial attempt to provide different projections and web tiles for the species *Larus hyperboreus*.

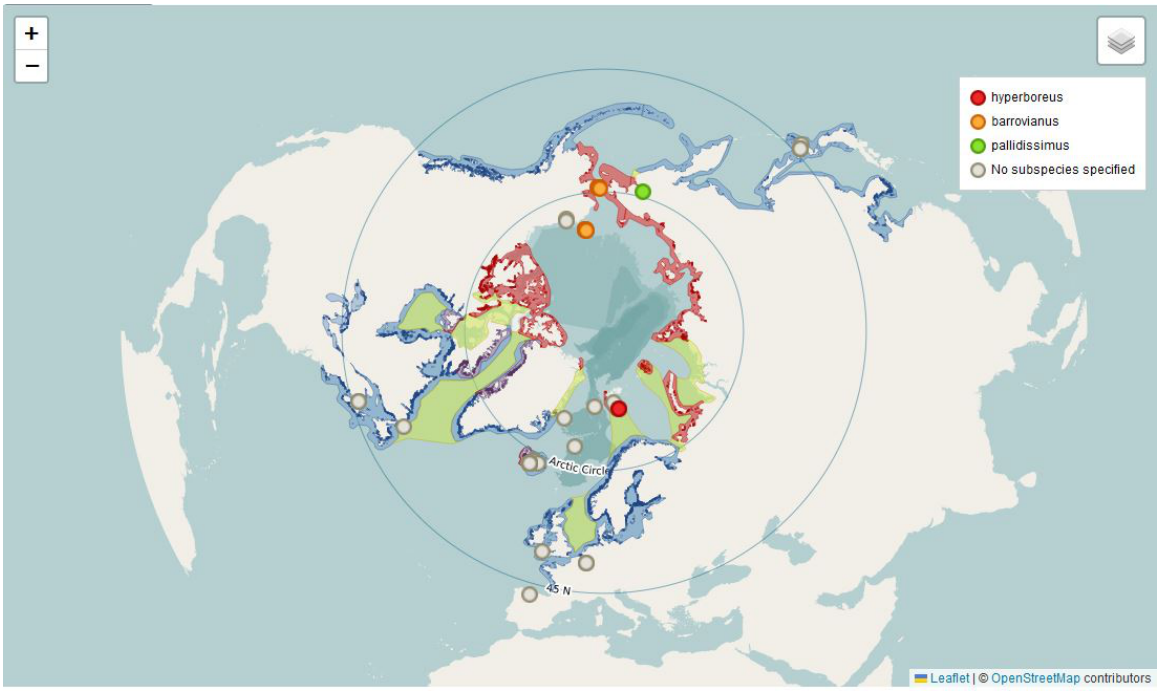


Figure 3.26: Illustration of polar region projection for Larus hyperboreus

Upload map

`UploadStepLocation.php` script constructs the HTML code necessary to generate the Upload map type. This map type utilizes the `LocationSearch.php` script. This API function brings up the previous recordings when the user zooms in on a particular area.

One of the main goals of our project, ending dependence on Google services, plays an important role in this type of map. As discussed in Section 3.3.2, users need to be provided with geocoding, reverse geocoding, and elevation service information. While the current map receives this information from Google, we developed open-source solutions to this.

After thorough research and comparative analysis, we selected Nominatim and GeoNames as our new providers for geocoding and elevation data services. Nominatim, powered by OpenStreetMap data, offers an unrestricted geocoding service. GeoNames, on the other hand, provides elevation data that is both accurate and readily accessible, thus serving as a strong alternative to Google's elevation services (Karmacharya, 2018). It would, alternatively, not be impossible to build up one's own elevation data server, based on open-source data such as Copernicus DEM[27].

[26] https://tile.gbif.org/ui/
[27] https://spacedata.copernicus.eu/collections/copernicus-digital-elevation-model

Additionally, to provide users the ability to search for locations, we used the Leaflet Geocoder plug-in.[28] This plug-in provides a user interface for geocoding directly on the map and can use various geocoding services as its back-end. The plug-in is configurable to utilize either Nominatim or an alternative geocoding service, with Nominatim set as the default option.

3.6 SURFACE STAGE: USABILITY TESTING

The final phase of the UCD process, as outlined in *Elements of User Experience* Garrett, 2002, is the surface phase. This phase shifts the focus from the implementation of interface and interaction designs to the comprehensive evaluation of the final product. This evaluation is important in terms of ensuring that the web map applications developed in the project meet both functional requirements and user experience expectations. The subsequent subsections discuss questionnaire design, while the results of this evaluation are presented in Section 4.4.

3.6.1 Introduction

Our objective is to follow the ISO 9241-11 standard[29] to conduct user-centered usability testing. This standard outlines three criteria such as effectiveness, efficiency, and satisfaction (Nivala et al., 2008; Voldán, 2010; Štěrba et al., 2014). Effectiveness in usability testing describes the users' ability to complete tasks accurately. Efficiency measures how effortlessly users can complete these tasks, and satisfaction reflects users' overall feelings about using the application. Moreover, it has been suggested that cartographic usability testing should also include learnability in the criteria (Štěrba et al., 2014; Roth et al., 2017). Hence, we must ensure that our cartographic solutions are easy to understand for users in order to effectively convey information.

We chose to conduct an online questionnaire as our usability testing method because of its ability to overcome challenges such as limited physical access to users and strict time constraints (Roth et al., 2015). The questionnaire consists of tasks and questions, and they were designed to get insights about four criteria: effectiveness, efficiency, satisfaction, and learnability.

The purpose is to evaluate the usability of our recently developed mapping applications. Our goal is to evaluate the functionality of the maps, not the users' individual skills or knowledge. There are no definitive right or wrong answers; therefore, users are encouraged to openly express their true thoughts and personal experiences. Additionally, participation is anonymous. Responses are kept confidential and are analyzed for the sole purpose of evaluating map features. No personal information is included in the report or in any publications resulting from this study.

To facilitate the publication of the web map applications, a Docker image containing the necessary software and configurations is installed on a server at ITC, University of Twente.

Participants are sent an invitation to complete the survey via email. This invitation includes instructions on how to access the map applications and submit responses.

[28] https://github.com/perliedman/leaflet-control-geocoder
[29] https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-1:v1:en

3.6.2 Questionnaire design

Our questionnaire is designed to collect feedback on the new features added to the application.

The questionnaire consists of four sections. The first section asks questions about the user's device, while subsequent sections focus on specific tasks associated with various map types:`browse-by-region`, `genus`, `species`, `subspecies`. These tasks are carefully crafted to be short and direct, and they are accompanied by a corresponding question.

We included both scaled (1–5), multiple choice, and open-ended questions. Users are able to quickly score their experiences with scaled questions. On the other hand, open-ended questions are posed to obtain deeper user feedback by encouraging users to share their thoughts, experiences, and suggestions.

In this questionnaire, we created three sections to provide feedback on three map types that were re-engineered: A) *species and subspecies map*, B) *genus map*, and C) `browse-by-region`. Each section of the questionnaire contains an introduction text, hyperlinks that provide access to such maps, and some simple questions for users to answer. We also inform users that the interface provided for this test is not an exact copy of the working XC website.

Demographic question

Objective 1: To get a better insight into users, in the first question, we ask a demographics question. Participants were provided with multiple choice options, including Desktop/Laptop, Tablet, Smartphone, or Other.

Question 1: What type of device do you most commonly use to access the Xeno-canto website?

Species and subspecies map

This section of our questionnaire focuses on species and subspecies map types, using the example of *Corvus macrorhynchos* specifically. This species, with its nine subspecies, is an ideal case to showcase the effectiveness of our color coding system.

Scenario: You are exploring the sounds of *Corvus macrorhynchos.*

Task 0: Go to the link: http://owsgip.itc.utwente.nl:82/species/Corvus-macrorhynchos

Task 1: Experiment with the new zoom functionality using your mouse wheel.

Task 2: Use the toggle buttons to show or hide species ranges and markers.

Task 3: Interact with the basemap switcher and explore the various basemap options available.

Task 4: Review the legend and explore how the marker colors correspond to the subspecies order.

Task 5: Explore subspecies map for *Corvus macrorhynchos tibetosinensis.* The provided link:

http://owsgip.itc.utwente.nl:82/species/Corvus-macrorhynchos?query=ssp:%22tibetosinensis%22

Task 6: Use the additional buttons to zoom to the boundary of markers and/or recordings.

After the participants complete the designated tasks, we proceed to the questionnaire phase.

Objective 2: Evaluate the ease of mouse zoom functionality.

Question 2: How helpful do you find the mouse wheel zoom functionality for enhancing your map navigation experience?

Objective 3: Determine the effectiveness of the toggle buttons in enhancing map customization.

Question 3: How useful are the toggle buttons in customizing the map view to your preferences?

Objective 4: Evaluate the ease of use and utility of the basemap switcher.

Question 4: Please share your thoughts on the basemaps switcher. How appealing do you find it? Do the various basemap options align with your preferences?

Objective 5: Evaluate whether the color coding schema is effective and helps in subspecies distinction.

Question 5: How intuitive do you find the color coding schema for the markers in helping you identify and differentiate between subspecies?

Objective 6: Evaluate the impact of keeping the color consistent for each subspecies on the user's experience.

Question 6: How useful do you think it is that a subspecies is always shown with the same color?

Objective 7: Finally, the last question in the section aims to collect information about the overall experience of the users.

Question 7: Do you have any further comments on (Sub)species map type? Are there any features that you particularly liked or disliked? Please share any additional feedback or suggestions you might have.

Genus map

The second section of the questionnaire is for the genus map type.

Scenario: You want to discover Genus: Anser.

Task 0: Go to the link: http://owsgip.itc.utwente.nl:82/genus/Anser

Task 1: Interact with the map to explore species ranges at the genus level.

After the participants complete the designated tasks, we proceed to the questionnaire phase.

Objective 8: Determine the effectiveness of visual representations, their learnability, and overall user satisfaction.

Question 8: How effective do you find the overlay of species ranges in understanding bird diversity at the genus level?

Question 9: How easily do you grasp the meaning of the different shades of purple that represent overlaying species ranges?

Objective 10: Finally, the last question in the section aims to collect information about the overall experience of the users.

Question 10: Do you have any further comments on the Genus map type? Are there any features that you particularly liked or disliked? Please share any additional feedback or suggestions you might have.

Browse-by-region map

Scenario: You want to discover bird species diversity in the area you want.

Task 0: Go to the link: http://owsgip.itc.utwente.nl:82/explore/region

Task 1: Draw a bounding box on the intended area. Try utilizing the drawing button apart from the shift-key method.

Task 2: View and interpret the grid-based system. Toggle the grids between species and recordings.

Task 3: Switch between different basemaps.

Following the completion of the specified tasks by the participants, we then move into the phase of administering the questionnaire.

Objective 11: The first task starts with evaluating the drawing button. We aim to evaluate the effectiveness of whether the implemented feature allows users to successfully complete the task. The user's rating will provide insights into their level of satisfaction with this feature.

Question 11: How useful did you find drawing a bounding box using the button?

Objective 12: Assess the clarity and effectiveness of grid-based visual representations. We aim to measure whether users can effortlessly understand and use the grid-based presentation. This will also give us information about the learnability of this feature.

Question 12: How useful did you find the toggling between species and recordings within the grid?

Objective 13: Evaluate the ease of use and utility of the basemap switcher. In the species map, the basemap switcher is placed on the top left, but for `browse-by-region` map type, we placed it on the bottom left. This selection was made on purpose to provide users with different views to evaluate between them.

Question 13: Which position of the basemap switcher did you find more convenient? Top Left - species map type
Bottom Left - `browse-by-region`
Both positions are suitable for their respective maps

Objective 14: Finally, the last question in the section aims to collect information about the overall experience of the users. Open-ended feedback will be instrumental in uncovering areas that improve user experience.

Question 14: Do you have any further comments on this map type? Are there any features that you particularly liked or disliked? Please share any additional feedback or suggestions you might have.

Chapter 4

Results and Discussion

In this chapter, the results of our reverse engineering approach on the code base of the Xeno-canto platform are presented.

4.1 RESULT OF STRATEGY STAGE: ANALYZING THE QUESTIONNAIRE

As outlined in Section 3.2.2, the design of the initial survey categorized `browse-by-region`, `species`, and `subspecies` in the same category. This made it difficult for us to get more specific user feedback for each map type independently. Nevertheless, to avoid overburdening respondents with repeated questioning, the available responses were interpreted as effectively as possible to get valuable insight into users' knowledge, motivation, and needs.

Participants expressed high confidence in using web map applications and browsing the internet. This indicates that XC users have the necessary familiarity to engage with complex geodata visualization solutions. Nevertheless, we should also note that the survey respondents are not representative of all XC users because these participants chose to be a part of this survey study. A significant portion of the XC community is likely more passively engaged with the platform and uses it as a useful sound library, and they may not have been interested in participating in the questionnaire. We think that the users who participated in the survey may be more active and experienced users.

The analysis of user response preferences suggests that the `species` and `upload` are the most frequently utilized map types. In comparison, community-related map types such as `mysteries`, `user profile`, and `forum` are less favored among users.

Improvements to the XC code base were achieved through the interpretation of user feedback, aiming to meet basic needs in three different areas. First, we found a need for more detailed and informative geographic data visualizations for `species`, `subspecies`, and the `genus` map types. These types can be enhanced to enable geographic data visualizations that reflect taxonomic information, acoustic types, and seasonal patterns. Furthermore, alternative map projections, especially for the polar regions, can be offered alongside the traditional Web Mercator projection for a more realistic visualization. Users also suggested that toggling the recording markers and the species ranges would provide a richer and more customized map experience.

Secondly, for the `browse-by-region` type, suggested improvements include a button that provides an alternative way to draw a bounding box. Additionally, users are interested in discovering areas with bird diversity. The current system only focuses on recordings, but a grid-based

presentation would enrich the user experience of bird diversity across regions beyond a singular focus on recordings.

Lastly, the responses to the questionnaire also suggest improvements that impact all map types. These include mouse wheel-controlled zooming for a more intuitive map navigation experience and diversifying basemap selections to show a variety of land types and elevation information. One common unwanted feature for all map types is the Pegman. It is a proprietary functionality provided by Google, and replacing the JavaScript mapping library will eliminate this function automatically.

4.2 RESULT OF SCOPE STAGE: SELECTION OF JAVASCRIPT LIBRARY AND DEFINING FEATURES OF XC

4.2.1 Selection of JavaScript library

As explained in detail in Section 3.3, both Leaflet and OpenLayers are powerful JavaScript libraries for mapping extensive geographic data. Both offer robust capabilities for handling big geodata visualizations. Leaflet creates an easy-to-use option for developers, while its rich plug-in ecosystem offers the opportunity to provide additional functionality. On the other hand, although OpenLayers offers more built-in functions, it comes with a more difficult learning curve. In the end, the choice between Leaflet and OpenLayers depends on the specific requirements of a project.

After a comprehensive evaluation of open-source JavaScript web map libraries based on their features, ease of use, performance, and community support in Section 3.3, we concluded that Leaflet is the best fit for our needs. The primary reason for this choice is that Leaflet offers a user-friendly and straightforward API, making it easy to use, even for developers with limited experience in coding. Furthermore, Leaflet has an active community with extensive support and resources. The wealth of documentation, tutorials, and examples available from the Leaflet community provides valuable assistance. Leaflet and OpenLayers have similar capabilities in terms of features, performance, and customizability. The ease of use and strong community support provided by Leaflet are the determining factors in our decision.

4.2.2 XC features and functionality

In Section 3.3.2, we examined the types of maps available on the Xeno-canto platform in detail, revealing their different features and the value they offer to users. Although the primary focus of the XC project was bird sounds, it recently expanded its scope to grasshoppers and bats. Our study specifically concentrated on the analysis and enhancement of the ornithological aspects of the project. XC offers eight distinct map types, each serving specific functionality. The `species` map type is the most important because it provides an important resource for ornithologists, sharing useful information for research and studies.

XC maps are equipped with features such as zoom controls and full-screen buttons. It also provides users with a basic street map with detailed terrain and a satellite imagery option for different base map choices.

The map projection of all these is Web Mercator, which is a common default projection in web

Table 4.1 The table displays a comparison of features for different map types, detailing attributes. For each map type, the presence of a feature is indicated with a checkmark (✓) and its absence with a cross (×). Exclusive functions for each map type are also listed in the last column.

Map type	*Projection*	*Basemaps*	*Marker clustering*	*Query XC API v2*	*Pop-up API*	*Pegman & full-scren*	*Zoom w/ mouse wheel*	*Exclusive functions*
Region	Web Mercator	street, terrain, satellite	✓	✓	✓	✓	✗	**Draws** bounding-box
Genus			✗	✗	✓	✓	✗	
Species & subspecies			✗	✓	✓	✓	✗	**Shows** the distribution ranges of species (KMZ)
Upload			✗	✗	✗	✓	✗	**Utilizes** forward & reverse geocoding, **Acquires** elevation data
Mysteries			✓	✓	✓	✓	✗	
Forum			✗	✗	✗	✓	✗	**Shows** the distribution ranges of species (KMZ)
User profile			✓	✓	✓	✓	✗	

map applications.

Overall, XC is a highly valued citizen science platform that enables recordists and other users to enhance their knowledge of bird vocalization, helping in bird identification and research and enjoying bird sounds worldwide. A summary of features is presented in Table 4.1.

4.3 RESULTS OF SKELETON STAGE: FORWARD ENGINEERING APPROACH

Our objective, as detailed in Section 1.3.1, was to develop an improved geodata visualization solution by using an open-source web map library for the Xeno-canto portal. This enhancement aims to not only improve the visualization of high-dimensional geospatial data but also to establish a more financially sustainable model by transitioning from Google Maps to Leaflet.

4.3.1 Browse-by-taxonomy: Genus, species, and subspecies map types

Core changes

As explained in Section 3.5.3, significant improvements were made in geodata visualization techniques for the browse-by-taxonomy map types: `genus`, `species`, and `subspecies` map types. Firstly, we introduced a dynamic marker color coding system compatible with the IOC World Bird List. This system visually represents, to the extent possible in a linear approach, the taxonomic relationships between subspecies. The color scheme was optimized for human color perception, using a 240° spectrum on the color wheel and selecting seven primary colors to provide variety and distinction.

Secondly, we transitioned from KMZ/KML files to GeoJSON/GeoJSON.zip files as part of our effort to move away from the Google ecosystem. The current species ranges are in the system

Table 4.2 File size comparisons before and after compression

Species	KML	KMZ	reduction (%)	GeoJSON	GeoJSON.zip	reduction (%)
Corvus macrorhynchos	357.3 kB	120.4 kB	72.94	436.4 kB	96.7 kB	77.84
Corvus corax	7,0 MB	1,9 MB	77.14	7,8 MB	1,6 MB	79.49
Anser albifrons	2,5 kB	734.7 kB	76.40	3,0 MB	604.1 kB	80.34
Ibidorhyncha struthersii	3.9 kB	1.9 kB	56.41	4.7 kB	1.7 kB	63.83

as KMZ files, which are KML files zipped. GeoJSON is not zipped, and its file size is bigger than KMZ or KML. We can achieve size efficiency with GeoJSON files by compressing them into a .zip format. Zipping the GeoJSON files creates file sizes that are quite comparable to KMZ files, ensuring that they are also manageable and easily shared or transferred. Comparative data presented in Table 4.2 shows file size reductions achieved with the compression process.

Genus map

As presented in Section 3.5.3, the `genus` map was enhanced with the dynamic coloring of species markers. Furthermore, new buttons are added, allowing users to toggle the visibility of markers and species ranges to enhance user experience.

The previous `genus` map type only displays markers and does not contain information about species ranges. To increase this functionality, we combined the spatial extents of individual species ranges to create an overlay of the species ranges map at the genus level. This file is then encoded into the GeoJSON format. For an effective visualization of species diversity, we use the Chroma plug-in. In Figure 3.8, the current genus map type is demonstrated. In Figure 4.1, our newly developed `genus` map is shown, displayed in various shades of purple to indicate the species count within each species range for the genus *Anser*.

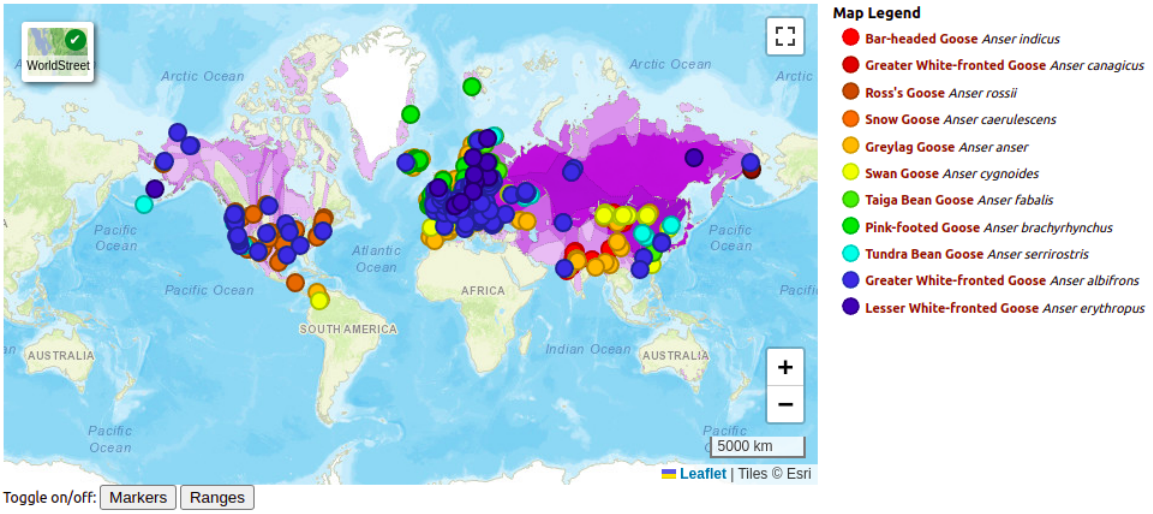


Figure 4.1: Newly developed genus map type for the genus *Anser*

Species map

As covered in Section 3.5.3, the dynamic color coding system is also applied to the species map. In this updated approach, each subspecies is assigned a color based on its position in the IOC bird list, and the corresponding marking color and name for these subspecies are presented in the legend in the same sequence. The figure 4.2 below shows the current `species` map, and the Figure 4.3 demonstrates our developed `species` map.

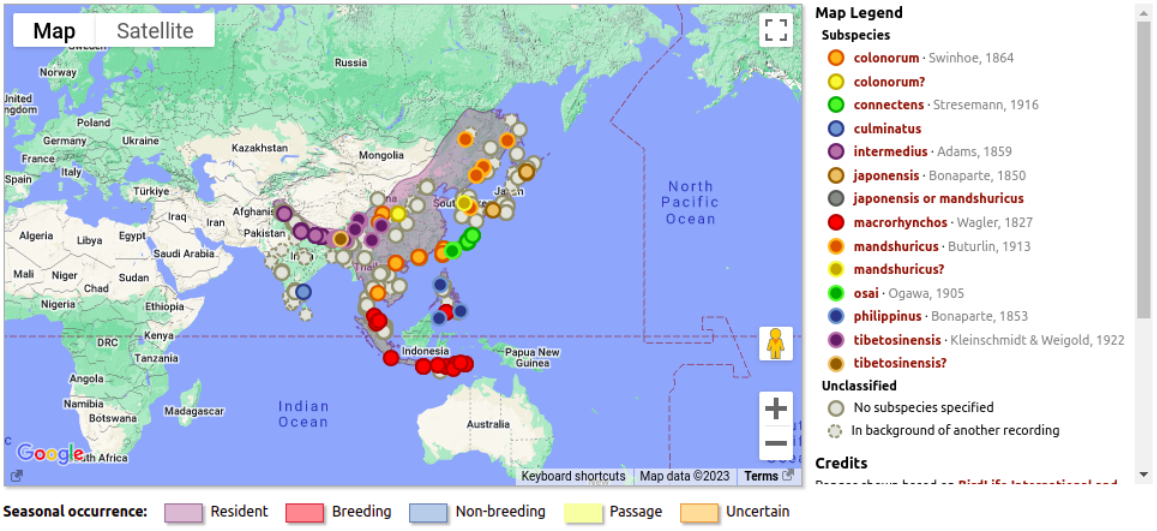


Figure 4.2: Illustration of current `species` map for *Corvus macrorhynchos*

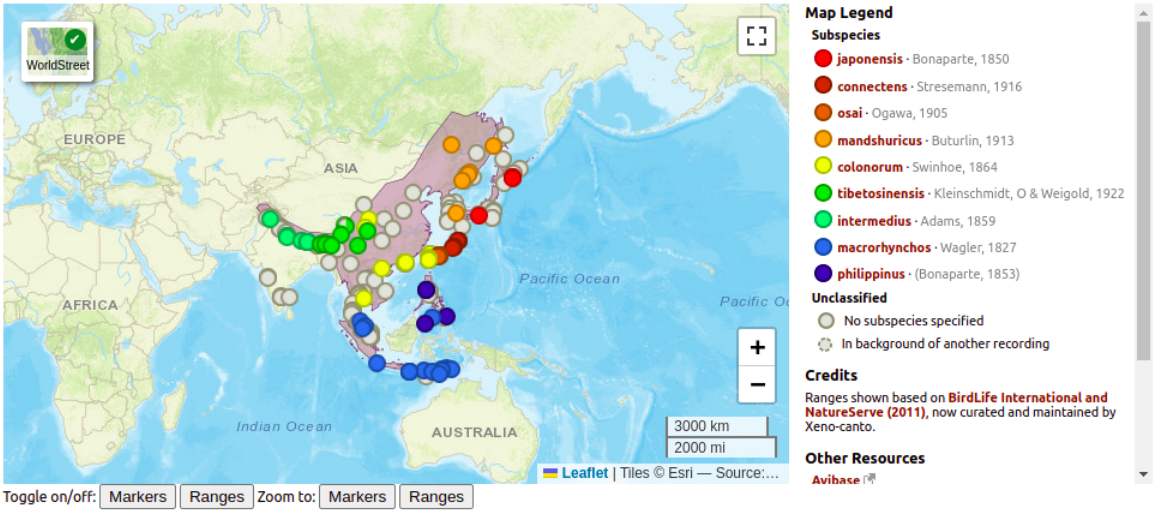


Figure 4.3: Illustration of developed `species` map for *Corvus macrorhynchos*

Besides that, we created a new way to show information using pie charts for locations where there are recordings of more than one subspecies. The figure below illustrates how these pie chart markers appear on the map.

Moreover, as detailed in Section 3.5.3, special attention is given to monotypic species, ensuring their inclusion and accurate representation in the `species` maps. Lastly, efforts have been made to

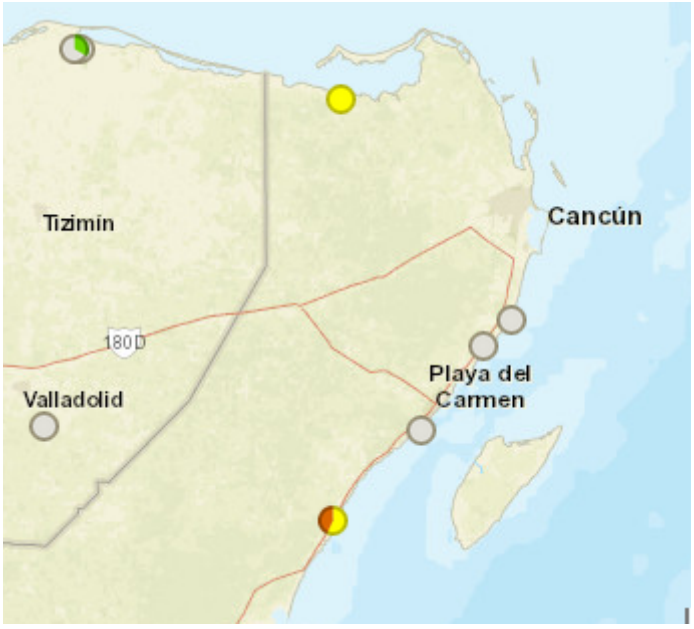


Figure 4.4: Pie chart markers on the species map illustrating the distribution of recordings across multiple subspecies for *Vireo pallens*

provide alternative projections for species in polar regions, but difficulties have been encountered in finding suitable basemaps.

Subspecies map

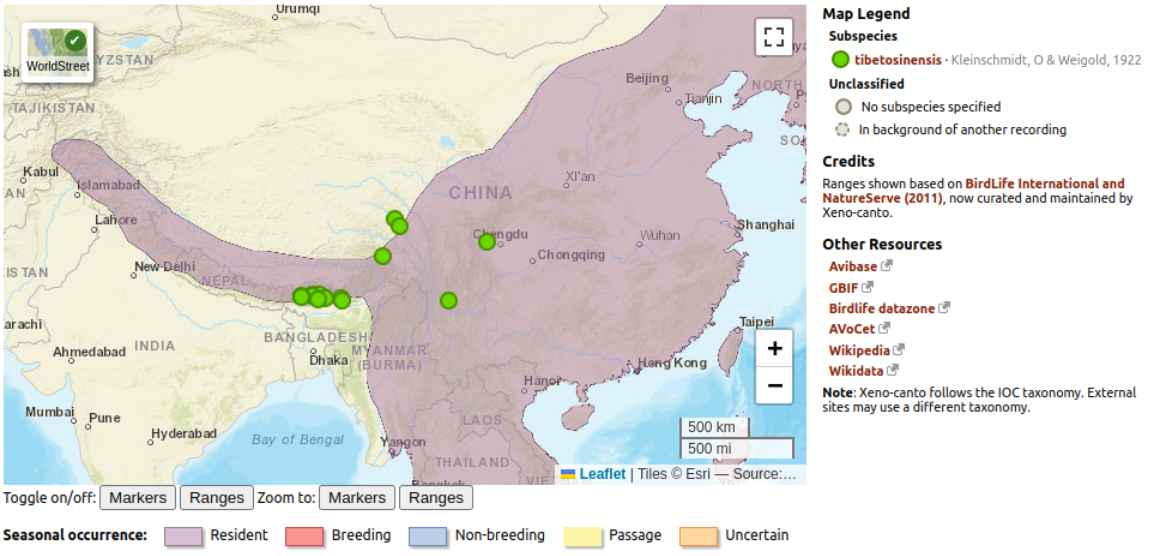


Figure 4.5: Illustration of newly developed subspecies map for Corvus macrorhynchos tibetosinensis

Adoption of the new color coding scheme offers further benefits. In the current map, when

a particular subspecies is selected, the marker for recordings for that subspecies is always red or orange. This is because the current system assigns the first suitable icon always. This creates inconsistencies between the species and subspecies maps. We now assign each subspecies its own color, and this is used in both map types always. For example, the subspecies *Corvus macrorhynchos tibetosinensis* is assigned the color green in our `species` map as shown in Figure 4.3. So when a user selects this particular species, it will again be displayed in green in its subspecies map, as shown in Figure 4.5.

4.3.2 Browse-by-region map

As detailed in Section 3.5.3, we enhanced the map interface of `browse-by-region` with a new button that allows users to draw a bounding box. This user-friendly addition provides an intuitive alternative to the previously used shift-key method to draw a bounding box. Additionally, we also developed a grid-based system that allows users to toggle between the number of recordings and the number of species. This novel feature makes it more easy to detect regions with high species diversity.

Figure 4.6 illustrates the current `browse-by-region` map over Madagascar. To highlight the enhancements, Figure 4.7 demonstrates the grid-based representation of recordings over Madagascar, and Figure 4.8 similarly shows the grid-based representation of species distribution. The effectiveness of our new implementation of the `browse-by-region` map is discussed in Section 4.4.3.

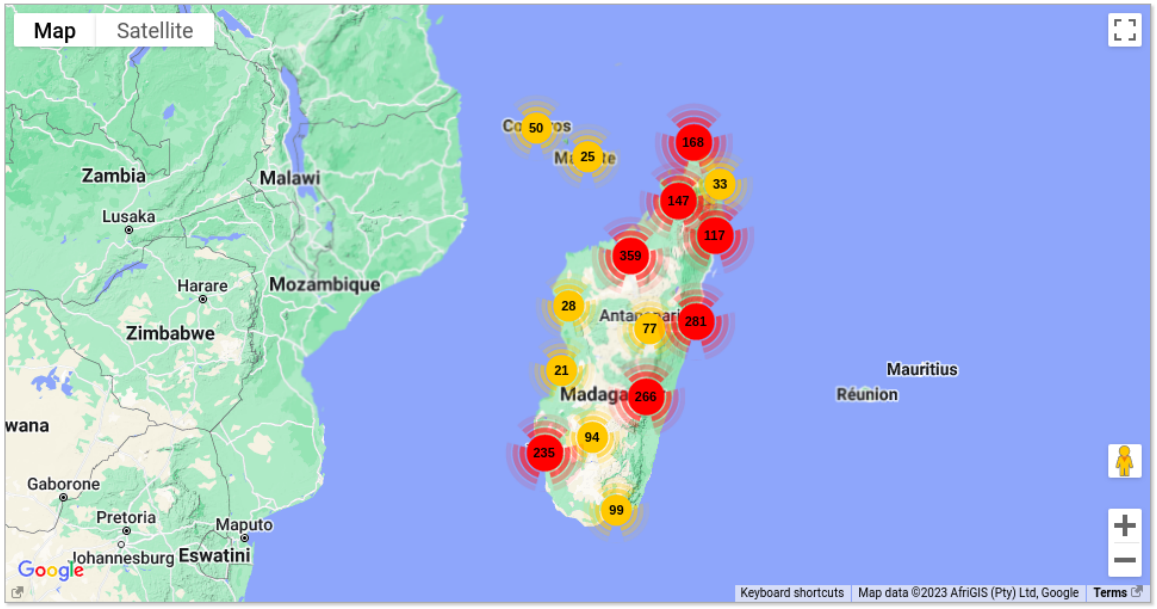


Figure 4.6: Existing XC map showing data by clustering method

4.3.3 Upload map type

As explained in Section 3.5.3, changes were made to the `upload` map type to help ending dependence on Google services. The move away from the Google ecosystem is not just technical

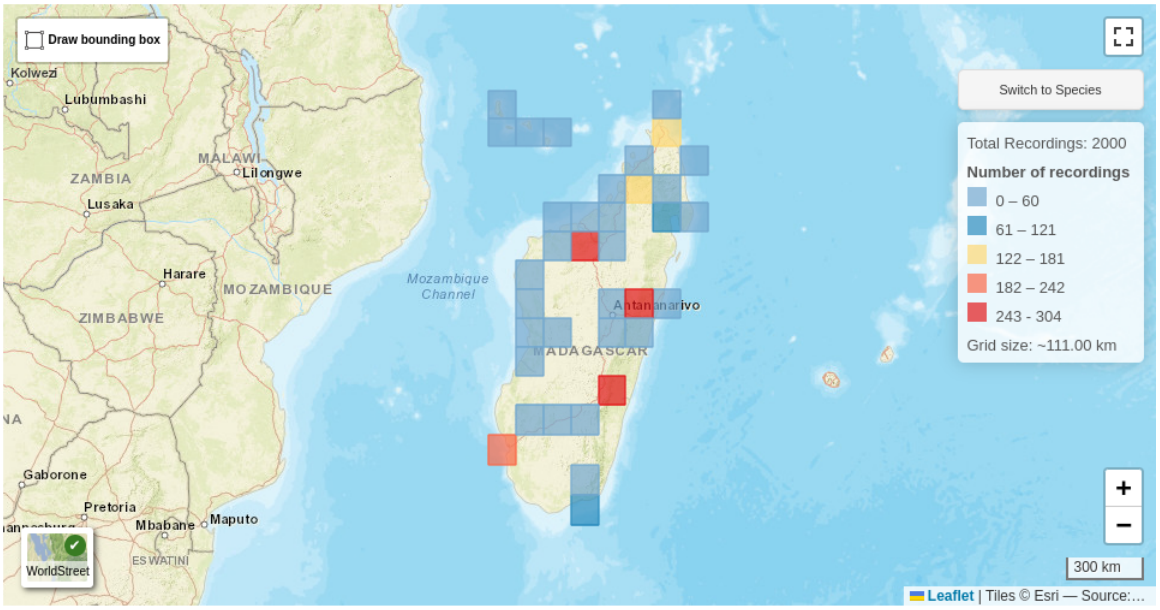


Figure 4.7: Map showing the number of recordings in a grid-based system

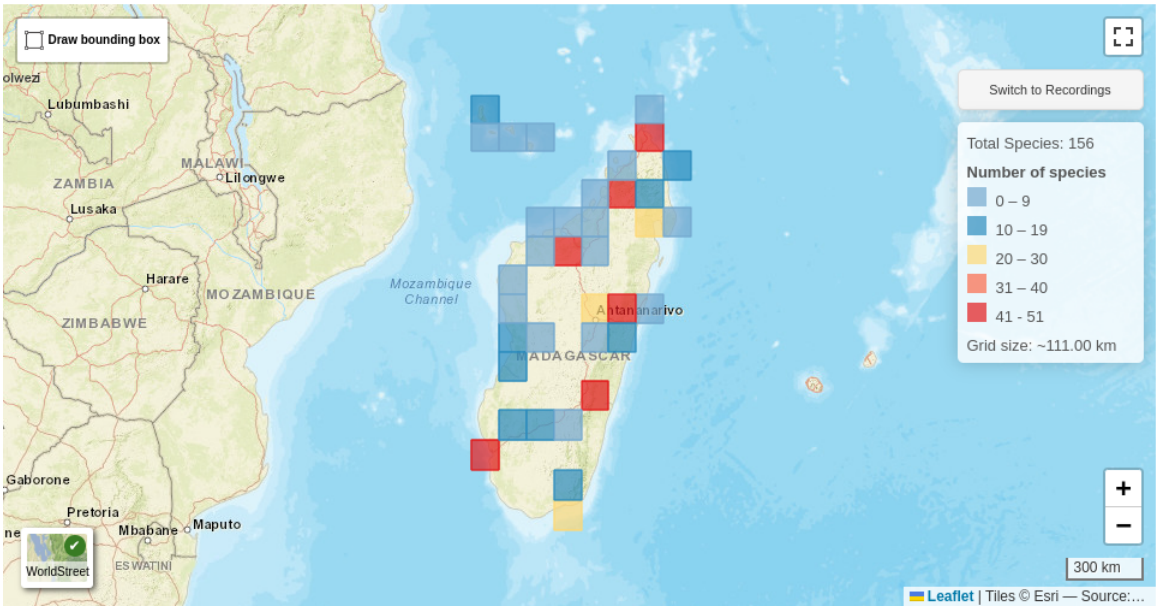


Figure 4.8: Map showing the number of species in a grid-based system

maintenance; it represents a strategic move toward cost efficiency. With these changes, users are provided with geocoding, reverse geocoding, and elevation services using open-source solutions. As outlined in Section 3.5.3, we conducted research on replacing Google's geolocation services and selected Nominatim and GeoNames as our primary service providers for geocoding and elevation data, respectively.

Our research and experiment results indicate that *Nominatim* and *GeoNames* can sustain the high standards required for geocoding and elevation services in mapping applications. This signif-

icant finding underscores the viability of these services as sustainable alternatives to proprietary options, thus aligning with our project's overarching goal of ending dependency on Google services.

We must acknowledge the technical difficulty encountered in user account access for the upload map type. Users encountered login issues that prevented access to the page. As a result, we were not able to include this map type in our usability testing.

4.4 RESULTS OF THE SURFACE STAGE: USABILITY TESTING

In this section, we present the results of our usability testing, which was conducted based on user-centered design principles. The usability testing process for the maps involved a combination of scale-based and open-ended questions, as outlined in Section 3.6.

We approached 41 XC users and received feedback from nine, all of whom used a desktop or laptop computer to access the questionnaire. Overall, the quantitative data from scale-based questions generally indicated positive user experiences. Table 4.3 displays the number of responses received for each question and their rating level. Additionally, the average scores were provided for each function. The user responses are categorized into five different levels, ranging from not at all useful to very useful.

Table 4.3 User ratings of map features. Summarized user feedback on map functionality features, with average scores.

	Map functionality	Not useful at all (1)	Slightly useful (2)	Moderate (3)	Highly useful (4)	Very useful (5)	Total average
Section 1 (Species)	Q2: Zoom with mouse wheel	2	-	1	1	5	*3.78/5*
	Q3: Show/hide species range and markers	2	1	2	2	2	*3.11/5*
	Q5: Color coding scheme for markers	1	-	-	6	2	*3.89/5*
	Q6: Marker color consistency in subspecies	1	-	3	4	1	*3.44/5*
Section 2 (Genus)	Q8: Effectiveness of species range overlay	-	2	2	4	1	*3.44/5*
	Q9: Purple shades representation	1	1	2	2	3	*3.56/5*
Section 3 (Browse-by-region)	Q11: Utility of drawing button for bounding box	1	2	1	2	3	*3.44/5*
	Q12: Number of species vs. recordings toggle	1	1	2	4	1	*3.33/5*

4.4.1 Section 1: Species and subspecies maps

As mentioned in Section 4.1, the ability to zoom in and out using the mouse wheel and toggling the map elements were seen as desired features. We received an average rating of 3.78 out of 5 for the usefulness of zooming with the mouse wheel feature, which aligns with user needs. Five out of nine respondents find the mouse wheel zoom functionality highly beneficial, giving it a rating of 5 in terms of usefulness. However, 2 out of 9 responses did not find it useful. While some users do not prefer to use a mouse or mouse wheel, most users want to have it. As having this functionality would not affect users who do not prefer it, thus, it is good to keep this function. Respondents have different opinions about the usefulness of toggle buttons in customizing the map view. Some

4 out of 9 responses find toggle buttons highly useful or useful; however, an equal number of respondents rated them as less useful or not useful. Consequently, it gets an average rating of 3.11.

For the basemap switcher, an open-ended question was provided. Respondents gave the basemap switcher positive comments. Elevation information on topographic basemap and having more basemap options are found to be very useful. Respondents stated that they mostly prefer topographic and satellite basemaps.

The color coding schema for markers is found to be intuitive for helping respondents identify and differentiate between subspecies. Consequently, it received the highest average score across our questionnaire, with a score of 3.89 out of 5. We have shown special effort on this color coding schema, and results indicate a positive user experience, suggesting that the feature effectively aids in distinguishing between subspecies. Six users scored it as four, and two users scored it as five. Only one user gave it a rating of 1. This could possibly be because of the user's distinct understanding of color familiarity and perception. The positive average score indicates that our effort to implement a color coding schema was successful. This implies good learnability because users can understand and use it without much instruction or prior knowledge. Continuous user feedback will further help optimize this feature for all users.

An additional benefit of implementing the color coding scheme is that it ensures consistency between the marker color for subspecies on both the species and subspecies map types. In terms of user feedback, five users rated it as 4, and one user gave it a rating of 5. However, three users rated it as 3, suggesting that while the feature is appreciated, it may not be considered essential always. Feedback from open-ended questions provided information about user preferences and areas for improvement. The idea of using different colors for subspecies was positively noted, indicating approval for visual differentiation.

Moreover, one user recommended including a distinctive icon or symbol to indicate nominate subspecies. Additionally, there was also a recommendation to adjust the zoom buttons to enable a closer view of species ranges.

4.4.2 Section 2: Genus map

The overlay feature for species ranges is perceived as effective in aiding users' understanding of bird diversity at the genus level, with an average rating of 3.56. Similarly, utilizing the different shades of purple to represent overlaying species ranges is found to have a relatively positive reception, with a score of 3.56 out of 5.

Open-ended feedback on the `genus` map revealed that while some users expressed dissatisfaction with the color scheme, others noted the map's potential usability and aesthetic improvements over previous versions.

4.4.3 Section 3: Browse-by-region map

The feedback on the usefulness of the bounding box drawing feature received an average rating of 3.44 out of 5. The overall response for toggling between the number of recordings and the number of species was moderately positive, with an average score of 3.33 out of 5.

The next question was designed to determine whether users always prefer the same user interface. The basemap switcher is placed differently for map types. We placed it at the bottom left on the `browse-by-region` map and at the top left on the `species` map. While four users state they prefer to see the basemaps switcher on the bottom left, the other four state that both positions are suitable for the respective map types. The user feedback suggests there is no overwhelming inclination for a single, uniform interface across different map types. This suggests that it may be reasonable to customize the user interface according to the distinct functions and contexts of each individual map type.

Finally, we posed an open-ended question to get further opinions for `browse-by-region` map. Respondents show a desire for more flexible and precise search tools, such as using queries before box selection and creating non-rectangular selection shapes.

Chapter 5

Conclusion and Future Work

This research represents a focused effort to enhance geodata visualization capabilities and the user experience on the Xeno-canto platform. As detailed in Section 4.4, positive feedback from usability testing not only underscores the success of our applications but also highlights the capability of Leaflet to provide geodata solutions that meet user needs.

The innovative introduction of a dynamic color coding schema, the transition from KML to the GeoJSON format, and a strategic move toward open-source development stand as significant achievements of this project.

XC consists of eight map types; however, we were not able to develop solutions for all of them due to limited time. For this reason, we had to narrow our focus to the map types that were most used and needed more advanced data visualizations. `browse-by-region`, `genus`, `species`, `subspecies`, and `upload` map types are developed within the scope of this project. However, further work is needed on the remaining map types: `forum`, `mysteries`, and `user profile`.

Refining and improving the user settings provides the potential to enhance the user experience for XC users. The current settings offer basic customization options, including language selection, preferences of table view, and control over the number of results displayed per page. However, having recognized the diversity of user preferences in Section 3.2, there is an opportunity to expand on these basic customization options. New customization options could potentially include options such as setting a default basemap, customizing scale units, and providing users with the option to customize the initial focus of the map upon loading. These improvements aim to develop a more personalized and intuitive user experience.

Appendix A

Integration of the Developed Code into the XC Codebase

This annex serves as a guide for integrating the developed features and functionality into the existing XC application infrastructure.

All modified PHP, JavaScript, and CSS scripts for each map type and necessary icons are published on my GitHub[1] repository.

XC is designed using a collection of Docker images. To get started, the docker container must be made operational by following the installation steps in the Xeno-canto GitLab[2] repository.

To successfully incorporate the new implementations, firstly, three key files need to be modified.

1. `xeno-canto/xc/Map.php`
2. `xeno-canto/xc/BaseTemplate.php`
3. `xeno-canto/static/css/xeno-canto.css`

Script `Map.php` plays a crucial role by linking to the JavaScript scripts and each plug-in listed in Table 3.7 and all map types, including `sspmap.js`, `gnsmap.js`, `browseregion.js`, and `map-upload.js`. need to be integrated with it. Similarly, `BaseTemplate.php` requires updates to accommodate the CSS for each plug-in and also include the `browseregion.css` to ensure proper styling of the generated legend on the `browse-by-region`.

It is important to distinguish between the existing Google Map script, `xc-map.js`, and the newly developed methods. The current script, `xc-map.js`, utilizes a single HTML `<div>` element, `map-canvas`, to render all map types. In contrast, the new approach employs separate JavaScript scripts for each map type, and they should be added to the xeno-canto.css file to ensure that each map is styled. These new IDs should be added to the `xeno-canto.css` file as shown below.

```
#species-profile #sspmap,
#species-profile #gnsmap,
```

[1] https://github.com/onrcrpk/Thesis-files
[2] https://gitlab.com/naturalis/bii/xenocanto/xeno-canto

```
#species-profile #browseregion,
#upload-location #map-upload {
    width: 100%;
    height: 400px;
}
```

Subsequent steps are then detailed for each map type.

A.1 SPECIES AND SUBSPECIES MAP

Script `sspmap.js` generaties species and subspecies map types. As explained above, this script must be included within `Map.php`.

In `SpeciesProfile.PHP`, identifies the HTML `<div>` element for the species and subspecies map as sspmap. Additionally, the HTML script that facilitates buttons for zooming and toggling between species and their recordings should be included in the `SpeciesProfile.PHP`.

The added HTML code for map `<div>` and buttons are shown below.

```
<!-- SpeciesProfile.PHP -->
<div id="sspmap"></div>
<span class='toggle-to-label'>Toggle on/off:</span>
<button id='toggleMarkerButton'>Markers</button>
<button id='toggleLayerButton'>Ranges</button>
<span class='zoom-to-label'>Zoom to:</span>
<button id='zoom-to-marker'>Markers</button>
<button id='zoom-to-range'>Ranges</button>
```

There is an additional `ioc.json` configuration file needed to operate the species and subspecies maps. As detailed in Section 3.5.3, it is crucial for assigning colors to subspecies. This JSON file specifies the order of bird species according to IOC standards. While both current XC maps and legend are generated by the back-end approach, we have followed the front-end approach for maps and stack with the current back-end approach for generating the legend. The database changes for the IOC sequence are also based on this JSON file.

For the legend, to set the subspecies on the IOC sequence and align with the color of markers for each subspecies, the `getLegendVfunc` function within `SpeciesProfileMap.PHP` must be modified. The `getLegendVfunc` function contains the query parameters used to retrieve data from the database. Before proceeding with the modifications, it is crucial to review the detailed database schema updates discussed in Section **??** Once the database schema is updated, the next step is to modify the SQL query as shown below.

```
$sql = "SELECT ssp, author, ssp_seq FROM taxonomy_ssp WHERE species_nr='" .
       $this->m_species->speciesNumber() . "' ORDER BY ssp_seq ASC";
```

Additionally, to facilitate the color ordering representation of species, two new functions, `createSvgMarker` and `calculateColor`, must be introduced to the `getLegendVfunc` function. These new functions are designed to generate HSL color codes and apply them to SVG images of 15x15 size instead of assigning a static order of PNG images. Should the name of the `getLegendVfunc` function be changed, it is critical to update all instances where the former name is referenced within `Map.PHP`. This step is necessary to preserve the application's functionality and ensure that all components work seamlessly together.

To plot species ranges, the file naming must follow the `<Genus>_<species>.geojson.zip` template. The JavaScript files assign the necessary colors for each seasonal occurrence. Each GeoJSON file includes a `season` attribute, which is assigned a value between 1 and 5, each representing a specific occurrence: (1) Resident, (2) Breeding, (3) Non-breeding, (4) Passage, and (5) Uncertain. As an example, the file `Corvus_macrorhynchos.geojson.zip` has been added to the repository.

A.2 GENUS MAP

The `gnsmap.js` script is designated for rendering the `genus` map and should be linked within the `Map.php` script.

Similar to the species map, the genus map requires a configuration file called `genusioc.json`. This file is important because it contains the IOC sequence for subspecies.

The genus map requires a unique `<div ID>` to function with the `gnsmap.js`. This is set within the `GenusProfile.PHP` as gnsmap. Accompanying this `<div>`, interactive elements such as buttons for zooming and toggling species or recordings must be embedded, as shown below.

```
<!-- GenusProfile.PHP -->
<div id='gnsmap'></div>
<span class='toggle-to-label'>Toggle on/off:</span>
<button id='toggleMarkerButton'>Markers</button>
<button id='toggleLayerButton'>Ranges</button>
```

To manage the map legend effectively, the script `GenusProfileMap.PHP` must be updated. Similar to the species map, it utilizes a function called `generateColors` to assign HSL color values to each genus represented on the map.

Regarding the genus name issues in the legend as outlined in Section 3.5.3, the `Query.PHP` script must be edited to accommodate the required changes for correct species name representation.

```
} elseif ($tag == 'gen') {
    // Force an exact match regardless of the operator provided
    $condition = $this->matchesCondition('birdsounds.genus', $term);
}
```

For a visual representation of the overlay of species ranges within a genus, the GeoJSON file

should follow a naming convention to `<Genus>.geojson.zip`. As an example, the file `Anser.geojson.zip` has been added to the repository.

A.3 BROWSE-BY-REGION MAP

The `browseregion.js` script is responsible for generating `browse-by-region`. To ensure its functionality within the existing codebase, this JavaScript file needs to be properly linked within the `Map.php`.

To integrate with `browseregion.js`, the existing HTML structure in `BrowseRegion.PHP` requires modification. the div-ID must be updated to `browseregion` in the `BrowseRegion.PHP`.

```
<!-- BrowseRegion.PHP -->
<div id="browseregion"></div>
```

Note: As mentioned in Section 3.4.2, `RegionResults.php` defines query parameters. The parameter used for querying the database has been set to 2000; thus, the number of recordings displayed on the map is limited to the first 2000 recordings. The code snippet below demonstrates the execution of the database query, highlighting the application of a limit on the number of records retrieved.

```
$res = $q->execute(Query::NO_PAGING, 2000);
```

A.4 UPLOAD MAP

The `mymap-upload.js` script is responsible for generating the web map for the `upload map` function. To operate correctly with the current code structure, it must be linked in the `Map.PHP`.

Similar to the other map types, the necessary changes include updating the map `<div>` from `map-canvas` to `map-upload` within the `UploadStepLocation.PHP`.

```
<!-- UploadStepLocation.PHP -->
<div id="map-upload"></div>
```

The JavaScript script can successfully locate and interact with the necessary HTML tags, enabling it to display coordinates, addresses, and elevation data.

List of References

Agarina, M., Karim, A. S., & Sutedi, S. (2019). User-Centered Design Method in the Analysis of User Interface Design of the Department of Informatics System's Website. *Proceeding International Conference on Information Technology and Business*, *0*(0), 218–230. https://jurnal.darmajaya.ac.id/index.php/icitb/article/view/2098

AlMarzouq, M., Zheng, L., Rong, G., & Grover, V. (2005). Open Source: Concepts, Benefits, and Challenges. *Communications of the Association for Information Systems*, *16*. https://doi.org/10.17705/1CAIS.01637

Ballhausen, M. (2019). Free and open source software licenses explained. *Computer*, *52*(6), 82–86. https://doi.org/10.1109/MC.2019.2907766

Battersby, S. E., Finn, M. P., Usery, E. L., & Yamamoto, K. H. (2014). Implications of web mercator and its use in online mapping. *Cartographica*, *49*(2), 85–101. https://doi.org/10.3138/CARTO.49.2.2313/ASSET/IMAGES/SMALL/01{_}49.2BATTERSBY{_}F11.GIF

Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S., & Schaub, T. (2016). The GeoJSON Format. https://doi.org/10.17487/RFC7946

Cambon, J., Hernangómez, D., Belanger, C., & Possenriede, D. (2021). tidygeocoder: An R package for geocoding. *Journal of Open Source Software*, *6*(65), 3544. https://doi.org/10.21105/JOSS.03544

Casteleyn, S., Garrigós, I., & Mazón, J. N. (2014). Ten years of Rich Internet applications: A systematic mapping study, and beyond. *ACM Transactions on the Web*, *8*(3). https://doi.org/10.1145/2626369

Chandra Mishra, K., & Dutta, S. (2023). A simple and secure user authentication scheme using Map Street View with usability analysis based on ISO/IEC 25022. *International Journal of Information Security*, *22*, 403–415. https://doi.org/10.1007/s10207-022-00636-0

Cybulski, P., & Horbinski, T. (2020). User Experience in Using Graphical User Interfaces of Web Maps. *ISPRS International Journal of Geo-Information 2020, Vol. 9, Page 412*, *9*(7), 412. https://doi.org/10.3390/IJGI9070412

Edler, D., & Vetter, M. (2019). The Simplicity of Modern Audiovisual Web Cartography: An Example with the Open-Source JavaScript Library leaflet.js. *KN - Journal of Cartography and Geographic Information*, *69*(1), 51–62. https://doi.org/10.1007/S42489-019-00006-2/FIGURES/13

European Commission. (2004). Why do we need to take care of our birds? https://ec.europa.eu/environment/nature/legislation/birdsdirective/docs/why_take_care_of_birds.pdf

Farkas, G. (2017). Applicability of open-source web mapping libraries for building massive Web GIS clients. *Journal of Geographical Systems*, *19*(3), 273–295. https://doi.org/10.1007/S10109-017-0248-Z/TABLES/10

García-Moreno, J., Clay, R. P., & Ríos-Muñoz, C. A. (2007). The importance of birds for conservation in the Neotropical region. *Journal of Ornithology*, *148*(SUPLL. 2). https://doi.org/10.1007/S10336-007-0194-5

Garrett, J. J. (2002). The Elements of User Experience: User-Centered Design for the Web. *Elements*, *10*, 208. http://www.amazon.com/dp/0735712026

Garrett, J. J. (2005). Ajax: A New Approach to Web Applications. http://www.adaptivepath.com/publications/essays/archives/000385print.php

Gede, M. (2023). Automatic Labels in Leaflet. *Advances in Cartography and GIScience of the ICA*, *4*, 1–5. https://doi.org/10.5194/ICA-ADV-4-8-2023

Gong, X., Erwee, F., & Rautenbach, V. (2019). GEOMETRY VIEWER FOR PGADMIN4: A PROCESS GUIDED BY THE GOOGLE SUMMER OF CODE. https://doi.org/10.5194/isprs-archives-XLII-4-W14-79-2019

Haklay, M., Singleton, A., & Parker, C. (2008). Web mapping 2.0: The neogeography of the GeoWeb. *Geography Compass*, *2*(6), 2011–2039. https://doi.org/10.1111/J.1749-8198.2008.00167.X

Hancock, S. C., Essl, F., Kraak, M. J., Dawson, W., Kreft, H., Pyšek, P., Pergl, J., van Kleunen, M., Weigelt, P., Winter, M., Gartner, G., & Lenzner, B. (2022). Introducing the combined atlas framework for large-scale web-based data visualization: The GloNAF atlas of plant invasion. *Methods in Ecology and Evolution*, *13*(5), 1073–1081. https://doi.org/10.1111/2041-210X.13820

Holzinger, A. (2005). Usability engineering methods for software developers. *Communications of the ACM*, *48*(1), 71–74. https://doi.org/10.1145/1039539.1039541

IOC World Bird List – Version 13.1. (2023). https://www.worldbirdnames.org/new/

Karmacharya, E. A. (2018). Positional Accuracy of Online Geocoding Services: Case Study of Bhaktapur District. *Journal on Geoinformatics, Nepal*, *17*(1), 16–21. https://doi.org/10.3126/NJG.V17I1.23004

Kelling, S., Gerbracht, J., Fink, D., Lagoze, C., Wong, W.-K., Yu, J., Damoulas, T., & Gomes, C. (2012). eBird: A human/computer learning network for biodiversity conservation and research. *IAAI*.

Khitrin, M. O. (2017). Comparison of JavaScript libraries for web-cartography. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control & Radioelectronics*, *17*(3), 147–152. https://doi.org/10.14529/CTCR170317

Kounadi, O., Lampoltshammer, T. J., Leitner, M., & Heistracher, T. (2013). Accuracy and privacy aspects in free online reverse geocoding services. *Cartography and Geographic Information Science*, *40*(2), 140–153. https://doi.org/10.1080/15230406.2013.777138

Kullenberg, C., & Kasperowski, D. (2016). What Is Citizen Science? – A Scientometric Meta-Analysis. *PLoS ONE*, *11*(1), 147152. https://doi.org/10.1371/JOURNAL.PONE.0147152

Laato, S., Mäntymäki, M., Islam, A. K., Hyrynsalmi, S., & Birkstedt, T. (2022). Trends and Trajectories in the Software Industry: implications for the future of work. *Information Systems Frontiers*, 1–16. https://doi.org/10.1007/S10796-022-10267-4/TABLES/4

Leaflet. (2023). An open-source javascript library for mobile-friendly interactive maps [Retrieved from https://leafletjs.com/].

Ledur, C., Griebler, D., Manssour, I., & Fernandes, L. G. (2016). Towards a Domain-Specific Language for geospatial data visualization maps with Big Data sets. *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, *2016-July*. https://doi.org/10.1109/AICCSA.2015.7507178

Lin, T., Liao, B. H., Hsu, S. L., & Wang, J. (2015). Experimental investigation of HSL color model in error diffusion. *2015 8th International Conference on Ubi-Media Computing, UMEDIA 2015 - Conference Proceeedings*, 268–272. https://doi.org/10.1109/UMEDIA.2015.7297467

Lin, W. (2017). Web-based Software Reengineering A case study on next generation product-selection system. *Degree Project Computer Science And Engineering*.

Lindsey Jean Schueman. (2021). Become a citizen scientist by recording the dawn chorus of birds | One Earth. https://www.oneearth.org/become-a-citizen-scientist-by-recording-the-dawn-chorus-of-birds/

Machwitz, M., Hass, E., Junk, J., Udelhoven, T., & Schlerf, M. (2019). CropGIS – A web application for the spatial and temporal visualization of past, present and future crop biomass development. *Computers and Electronics in Agriculture*, *161*, 185–193. https://doi.org/10.1016/J.COMPAG.2018.04.026

Majthoub, M., Qutqui, M. H., & Odeh, Y. (2018). Software Re-engineering: An Overview. *2018 8th International Conference on Computer Science and Information Technology, CSIT 2018*, 266–270. https://doi.org/10.1109/CSIT.2018.8486173

McClure, C. J., Lepage, D., Dunn, L., Anderson, D. L., Schulwitz, S. E., Camacho, L., Robinson, B. W., Christidis, L., Schulenberg, T. S., Iliff, M. J., Rasmussen, P. C., & Johnson, J. (2020). Towards reconciliation of the four world bird lists: hotspots of disagreement in taxonomy of raptors. *Proceedings of the Royal Society B: Biological Sciences*, *287*(1929). https://doi.org/10.1098/RSPB.2020.0683

Milić, P. (2022). Open data visualization by using javascript libraries. https://www.researchgate.net/profile/Petar-Milic/publication/369957252_Open_Data_Visualization_by_Using_Javascript_Libraries/links/6436b0c1609c170a1311051b/Open-Data-Visualization-by-Using-Javascript-Libraries.pdf

Mülazimoğlu, E., & Başaraner, M. (2019). User-centred Design And Evaluation Of Multimodal Tourist Maps. *International Journal of Engineering and Geosciences*. https://doi.org/10.26833/IJEG.535630

Netek, R., Brus, J., & Tomecka, O. (2019). Performance Testing on Marker Clustering and Heatmap Visualization Techniques: A Comparative Study on JavaScript Mapping Libraries. *ISPRS International Journal of Geo-Information 2019, Vol. 8, Page 348*, *8*(8), 348. https://doi.org/10.3390/IJGI8080348

Newman, G., Wiggins, A., Crall, A., Graham, E., Newman, S., & Crowston, K. (2012). The future of Citizen science: Emerging technologies and shifting paradigms. *Frontiers in Ecology and the Environment*, *10*(6), 298–304. https://doi.org/10.1890/110294

Nielsen, J. (1993). Usability engineering, 362.

Nivala, A. M., Brewster, S., & Sarjakoski, L. T. (2008). Usability Evaluation of Web Mapping Sites. *The Cartographic Journal*, *45*(2), 129–138. https://doi.org/10.1179/174327708X305120

OpenLayers. (2023). Openlayers background tutorial [Retrieved from https://openlayers.org/doc/tutorials/background.html].

Ortega, S. V. M., & Sarria-Paja, M. (2022). Bird Identification from the Thamnophilidae Family at the Andean Region of Colombia. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *13293 LNCS*, 243–257. https://doi.org/10.1007/978-3-031-10539-5{_}18/TABLES/2

Priyadarshani, N., Marsland, S., & Castro, I. (2017). Avian Biology Journal of Avian Biology Automated birdsong recognition in complex acoustic environments: a review. https://doi.org/10.1111/jav.01447

Roth, R. E., Çöltekin, A., Delazari, L., Filho, H. F., Griffin, A., Hall, A., Korpi, J., Lokka, I., Mendonça, A., Ooms, K., & van Elzakker, C. P. (2017). User studies in cartography: opportunities for empirical research on interactive maps and visualizations. *International Journal of Cartography*, *3*(sup1), 61–89. https://doi.org/10.1080/23729333.2017.1288534

Roth, R. E., Ross, K. S., & MacEachren, A. M. (2015). User-Centered Design for Interactive Maps: A Case Study in Crime Analysis. *ISPRS International Journal of Geo-Information 2015, Vol. 4, Pages 262-301*, *4*(1), 262–301. https://doi.org/10.3390/IJGI4010262

Skarlatidou, A., Ponti, M., Sprinks, J., Nold, C., Haklay, M., & Kanjo, E. (2019). J Com User Experience Of Digital Technologies In Citizen Science User Experience Of Digital Technologies In Citizen Science. *Journal of Science Communication*, *18*(01). https://doi.org/10.22323/2.18010501

Štěrba, Z., Šašinka, Č., Stachoň, Z., et al. (2014). Usability testing of cartographic visualizations: Principles and research methods. *Proceedings of the 5th International Conference on Cartography and GIS Proceedings*, *1*, 147–256.

Tsou, M. H. (2011). Revisiting web cartography in the United States: The rise of user-centered design. *Cartography and Geographic Information Science*, *38*(3), 250–257. https://doi.org/10.1559/15230406382250

Tsou, M. H., & Curran, J. M. (2008). User-centered design approaches for web mapping applications: A case study with USGS hydrological data in the United States. *Lecture Notes in Geoinformation and Cartography*, *0*(9783540720287), 301–321. https://doi.org/10.1007/978-3-540-72029-4{_}20

Tsou, M.-H. (2013). Cartography and Geographic Information Science Revisiting Web Cartography in the United States: the Rise of User-Centered Design. https://doi.org/10.1559/15230406382250

van den Brand, M. G. J., Klint, P., & Verhoef, C. (1997). Reverse engineering and system renovationan annotated bibliography. *ACM SIGSOFT Software Engineering Notes*, *22*(1), 57–68. https://doi.org/10.1145/251759.251849

Varnajot, A. (2019). "Walk the line": An ethnographic study of the ritual of crossing the Arctic Circle—Case Rovaniemi. *Tourist Studies*, *19*(4), 434–452. https://doi.org/10.1177/1468797619836546/ASSET/IMAGES/LARGE/10.1177{_}1468797619836546-FIG4.JPEG

Velkoski, G., Gusev, M., & Ristov, S. (2017). Analysis of interactive image technologies. *24th Telecommunications Forum, TELFOR 2016*. https://doi.org/10.1109/TELFOR.2016.7818929

Vohland, K., Land-Zandstra, A., Ceccaroni, L., Lemmens, R., Perelló, J., Ponti, M., Samson, R., & Wagenknecht, K. (2021). Editorial: The Science of Citizen Science Evolves. *The Science of Citizen Science*, 1–12. https://doi.org/10.1007/978-3-030-58278-4{_}1

Voldán, P. (2010). Usability Testing Of Web Mapping Portals. *Geoinformatics FCE CTU*, *5*, 57–65. https://doi.org/10.14311/GI.5.5

Whelan, C. J., Wenny, D. G., & Marquis, R. J. (2008). Ecosystem Services Provided by Birds. *Annals of the New York Academy of Sciences*, *1134*(1), 25–60. https://doi.org/10.1196/ANNALS.1439.003

Won, S., & Westland, S. (2017). Colour meaning and context. *Color Research and Application*, *42*(4), 450–459. https://doi.org/10.1002/COL.22095

Xeno-canto. (2023). https://xeno-canto.org/about/xeno-canto

Xie, J., & Zhu, M. (2019). Handcrafted features and late fusion with deep learning for bird sound classification. https://doi.org/10.1016/j.ecoinf.2019.05.007

Yadav, P., & Darlington, J. (2016). Human Computation (2016) x:x:x-y Design Guidelines for the User-Centred Collaborative Citizen Science Platforms.

Zhang, G. (2020). Spatial and Temporal Patterns in Volunteer Data Contribution Activities: A Case Study of eBird. *ISPRS International Journal of Geo-Information 2020, Vol. 9, Page 597*, *9*(10), 597. https://doi.org/10.3390/IJGI9100597

Zulian, V., Miller, D. A. W., & Ferraz, G. (2021). Integrating citizen-science and planned-survey data improves species distribution estimates. *27*(12), 2498–2509. https://doi.org/10.2307/48632843

Zunino, A., Velázquez, G., Celemín, J. P., Mateos, C., Hirsch, M., & Rodriguez, J. M. (2020). Evaluating the Performance of Three Popular Web Mapping Libraries: A Case Study Using Argentina's Life Quality Index. *ISPRS International Journal of Geo-Information 2020, Vol. 9, Page 563*, *9*(10), 563. https://doi.org/10.3390/IJGI9100563