

UNIVERSITY OF TARTU  
Institute of Computer Science  
Software Engineering Curriculum

**Richard Meitern**

**Bird Colony: an application for collecting  
spatial and fitness data from seabird colonies  
for analysing global patterns**

**Master's Thesis (30 ECTS)**

Supervisors:  
Prof. Tuul Sepp,  
Madis Vasser, PhD

Tartu 2025

# **Bird Colony: an application for collecting spatial and fitness data from seabird colonies for analysing global patterns**

## **Abstract:**

Seabirds are vital indicators of environmental change due to their wide-ranging habitats and ecological sensitivity. Hence effective monitoring of seabirds is crucial for global understanding of environmental change. To better utilize the ongoing research efforts that collect data about seabird breeding there is a need for software tools that facilitate data collection in seabird research. This thesis introduces the Bird Colony application, a user-friendly tool developed with Flutter and Firebase to streamline data collection and management in seabird colony monitoring. Key features of the developed application include precise nest mapping, real-time data entry, and experiment management - all tailored to the specific needs of field researchers. The application's cross-platform compatibility and real-time database enable immediate data synchronization between researchers working in a colony. The open-source nature of the application promotes customization and collaboration within the developer community, potentially improving seabird research methodologies and conservation strategies in the long run. To test and validate the usefulness of the developed software the Bird Colony application has been effectively used on Kakrarahu islet, Estonia to improve the monitoring of its common gull (*Larus canus*) breeding colony. On Kakrarahu common gulls have been monitored for several decades — all in an effort to enhance the understanding of seabird populations and their responses to environmental change. By improving data collection methods and fostering collaboration among researchers, this application aims to contribute significantly to seabird conservation and ecological research worldwide.

## **Keywords:**

Seabirds, data collection software, Flutter, Firebase

**CERCS:** P170, Computer science, numerical analysis, systems, control; B280, Animal ecology

## **Bird Colony: merelinnukolooniate ruumiliste ja kohasusandmete kogumise rakendus globaalsete mustrite uurimiseks**

### **Lühikokkuvõte:**

Merelinnud on olulised keskkonnamuutuste indikaatorid nende laia geograafilise leviku ja tundlikkuse tõttu muutustele ökosüsteemides. Seega on merelindude tõhus seire globaalsete keskkonnamuutuste mõju mõistmise jaoks ülioluline. Selleks, et paremini ära kasutada ressursi, mis kulub merelindude uurimiseks, on vaja tarkvaravahendeid, mis hõlbustavad andmete kogumist merelindude pesitsuskolooniatest. Käesolev lõputöö tutvustab rakendust „Bird Colony“, kasutajasõbralikku tööriista, mis on välja töötatud Flutteri ja Firebase'i abil, et muuta sujuvamaks andmete kogumist ja haldamist merelindude kolooniate seires. „Bird Colony“ peamised funktsioonid hõlmavad täpset pesade kaardistamist, reaallajas andmete sisestamist ja katsete haldamist, mis kõik on kohandatud välitööd tegevate inimeste spetsiifilisi vajadusi silmas pidades. Rakendus on platvormiülene ja reaallajas uueneva andmebaasiga, mis võimaldab meeskondliku koostööd pesitsuskoloonias. Tänu avatud lähtekoodile on rakendus kohandamiseks kättesaadav ka teistele arendajatele, mis võiks pikas perspektiivis veelgi suurendada rakenduse kasulikust uurimismetoodikate täiustamisel.

Selleks, et testida ja arendada rakenduse kasulikust, on „Bird Colony“ rakendust arendatud mitme aasta vältel Kakrarahu kalakajakate (*Larus canus*) pesitsuskoloonias toimuvate protsesside jälgimise parandamiseks. Kakrarahu kajakakolooniat on jälgitud aastakümneid, eesmärgiga parandada merelindude populatsioonides toimuvate protsesside mõistmist muuhulgas ka keskkonnamuutuste kontekstis. Parandades andmete kogumise meetodeid ja edendades koostööd uurijate vahel, panustab käesolevas töös arendatud rakendus merelindude kaitse ja ökoloogilise uurimistöö arengusse kogu maailmas.

### **Võtmesõnad:**

Merelinnud, andmekogumistarkvara, Flutter, Firebase

**CERCS:** P170, Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria); B280, Loomaökoloogia

# Table of Contents

Introduction .....	5
1 Current Methods for Data Collection for Bird Colony Monitoring.....	7
1.1 Traditional methods.....	7
1.2 Modern Methods .....	8
1.3 Available Software .....	9
2 Technological Choices for Bird Colony Application.....	15
2.1 Development Framework .....	15
2.2 Data Storage .....	17
3 Overview of the Developed Software .....	20
3.1 Backend Architecture .....	20
3.2 Frontend Architecture.....	22
3.3 New Instance Setup .....	25
3.4 Formal Testing.....	26
3.5 Real World Usage.....	29
3.6 Future Directions .....	31
4 Conclusions.....	33
References .....	35
Appendix .....	42
I. Example Microsoft Copilot Prompt for Text Ameliorating.....	42
II. License.....	43

## Introduction

Seabirds are good model systems to study the effects of environmental change on free-living animals on wider spatial scales and time-scales (Grémillet and Boulinier 2009). In addition to being affected by anthropogenic environmental change, including climate change, depletion of fish stocks, bycatch and marine pollution, their long lifespan and flexible pace of life allows for long-term studies (Einoder 2009). Seabirds are exposed to pollution on land, in water and in the air, and as higher links in the food chain, they are vulnerable to the accumulation of pollutants (Burger and Gochfeld 2004; Kreitsberg et al. 2023). Most seabirds are colonial breeders, which is why they are particularly sensitive to factors related to population density, which in turn are related to habitat loss and fragmentation (Miller et al. 2019). Seabirds also often migrate over large distances, exposed to environmental changes on a global scale (Trevail et al. 2023).

Despite the wealth of data collected from seabird breeding colonies worldwide, there is a need for standardized data collection methods to facilitate large-scale analysis of global patterns in seabird fitness and ecology (Lewison et al. 2012; Grémillet, Chevallier, and Guinet 2022; Fierro-Calderón et al. 2021). This thesis addresses this gap by introducing Bird Colony, a Flutter application designed to streamline and standardize data collection from seabird colonies. By providing a standardized, user-friendly platform for data collection, Bird Colony has the potential to contribute to more comprehensive and comparable datasets across different geographical locations.

This thesis is structured as follows: The introduction presents the background and importance of data collection for bird colony monitoring. Chapter 1 begins by discussing traditional methods such as ground surveys, aerial surveys, and acoustic monitoring. It then explores modern methods, including the use of unmanned aerial vehicles (UAVs) and other more recent technologies, ending with an overview of available software tools for data collection and analysis. Chapter 2 looks into the development framework selection used for creating the Bird Colony application and the data storage solutions implemented. Chapter 3 provides a detailed look at the functionalities of the Bird Colony application, including the software architecture, real world usage and future directions. Finally, in the conclusions I reflect on how the Bird Colony application addresses the challenges identified in seabird colony monitoring, discuss its practical implications for ecological research and conservation efforts, and provide recommendations for future improvements and wider adoption within the research community.

Artificial intelligence (AI) tools were used in the creation of this thesis. For code writing, I utilized Android Studio<sup>1</sup> with GitHub Copilot's<sup>2</sup> auto-completion features enabled. The exact versions of GitHub Copilot used can be determined by comparing the git commit history with the release schedule of GitHub Copilot. For the creation of thesis text passages, I followed a procedure where I first wrote the paragraph, including references, and then prompted Microsoft Copilot<sup>3</sup> in "Protected" mode with the option to use GPT-4 and the conversation style set to "Precise" to fix the language and seek advice on the next steps. Next, I either accepted or ignored the Copilots edits based on my personal preference and decided whether there is merit in pursuing its suggestions based on my understanding of the topic. This procedure was iterated several times during the thesis writing as my understanding of the topic and the output I wanted to get grew. An example conversation for this paragraph's first version can be found in Appendix I.

---

<sup>1</sup> <https://developer.android.com/studio>

<sup>2</sup> <https://github.com/features/copilot>

<sup>3</sup> <https://bing.com/chat> (as of October 2024 the UI used in the writing of the thesis is changed and the link redirects to <https://m365.cloud.microsoft/chat> without the same options)

# 1 Current Methods for Data Collection for Bird Colony Monitoring

The collection of data on bird nest locations and breeding success is a critical component of avian ecology and conservation (Fierro-Calderón et al. 2021). Traditional ground surveys have been the cornerstone of bird colony monitoring that have been complemented with numerous digital alternatives like satellite or aerial imaging or on the ground video surveillance (Edney and Wood 2021). While each method has its own advantages, they also come with significant challenges, such as accessibility issues, potential disturbance to birds, and limitations in data accuracy (Edney et al. 2023; Hodgson et al. 2016; Zink et al. 2023). In recent years, Geographic Information System (GIS) data has emerged as a valuable tool, but longitudinal studies and those using spatial data can generate large amounts of data, which is time-consuming to collect and manage (Nowak et al. 2020). Consequently, fieldwork often focuses on gathering data specific to the ongoing research project or monitoring program, potentially hindering the use of ongoing seabird monitoring projects for large-scale standardized data analysis. This chapter aims to explore the various methods and software tools used for data collection in the field, highlighting both traditional methods and modern innovations.

## 1.1 Traditional methods

Ground surveys are perhaps the most common traditional method for monitoring bird colonies (Fierro-Calderón et al. 2021; Sepp et al. 2017; Clairbaux et al. 2024). Conducting these surveys by physically visiting nesting sites to count nests and observe nesting behaviors allows for detailed observations of individual nests, including the number of eggs, the presence of chicks, and parental behaviors. However, ground surveys are labor-intensive and time-consuming, often requiring multiple visits to the same site throughout the breeding season to gather comprehensive data (Edney et al. 2023; Sardà-Palomera et al. 2017; Larson and Gawlik 2023). Yet, studies have shown that monitoring the number of chicks that reach different life stages can provide valuable insights into the reproductive success of a colony (Bhusal et al. 2023). Traditionally ground surveys have often lacked the ability to capture fine-scale spatial and temporal dynamics within colonies.

Another method for bird monitoring is aerial surveys using manned aircrafts. Low flying planes have been employed to monitor larger colonies or those situated in hard-to-reach areas. These surveys can cover extensive geographical areas quickly, allowing researchers

to obtain a broader perspective on bird populations and nesting distributions (Sardà-Palamera et al. 2012). However, aerial surveys also come with drawbacks, including high operational costs, the need for specialized equipment and trained personnel, and potential disturbances to the birds being monitored (Hodgson et al. 2016). Additionally, the accuracy of data collected from aerial surveys can be compromised by factors such as weather conditions, altitude, and the observer's ability to identify nests from a distance (Žydelis et al. 2019; Davis et al. 2022). Also traditional aerial methods often lack the ability to capture fine-scale spatial and temporal dynamics within colonies (Edney et al. 2023). This limitation underscores the need for more advanced methodologies that can provide comprehensive data.

## **1.2 Modern Methods**

Acoustic monitoring has emerged as an innovative approach to supplement traditional methods, particularly for colonial waterbirds (Arneill et al. 2020). This technique involves quantifying vocalizations from nesting birds to estimate colony size and nesting success. For instance, the call rates of heron chicks could be correlated with nest counts, providing a non-invasive means of monitoring (Larson and Gawlik 2023). However, this method is sensitive to various environmental factors, including vegetation density and wind speed, which can affect sound propagation and detection especially for seabirds (Priyadarshani, Castro, and Marsland 2018).

The seemingly most popular emerging method in the field of bird nest monitoring is the use of unmanned aerial vehicles (UAVs) (Nowak, Dziob, and Bogawski 2018). UAVs offer a non-invasive method for data collection, allowing researchers to gather high-resolution imagery and spatial data (Edney and Wood 2021; Hodgson et al. 2016). The advantage of UAVs lies in their ability to access remote or difficult-to-reach nesting sites without disturbing the birds (Sardà-Palamera et al. 2012). UAVs can operate at various altitudes and angles, providing essential high-resolution images for accurate nest identification and population assessments (Edney et al. 2023). They can capture sequential aerial images during a single sampling day, enabling researchers to analyze potential census errors and obtain confidence intervals for their estimates (Sardà-Palamera et al. 2012). Researchers have used different UAVs with cameras to study various bird species nesting patterns (Edney and Wood 2021). However, the high initial cost and ongoing maintenance expenses of quality UAVs can be prohibitive for many research projects, particularly those with limited funding. Especially considering that UAVs also stress the birds and are sometimes attacked in the field (Zink et

al. 2023; Junda et al. 2016). Nevertheless, the integration of UAV technology with GIS can streamline the process of data collection, enabling the analysis of large areas and the identification of nesting habitats with greater accuracy (Quamar et al. 2023).

The rise of GPS tracking technology has provided a way to obtain high-resolution data on foraging behaviour and habitat utilization. Frequently GPS trackers have been used in research to gain insights into the movements and behaviors of individual birds within colonies and beyond (Bernard et al. 2021). This method allows researchers to obtain precise location fixes, enabling the identification of critical foraging areas and the assessment of interactions with different factors like fishing activities (Torres et al. 2011; Carroll et al. 2019), light pollution (Rodríguez, Rodríguez, and Negro 2015; Rodríguez et al. 2022) or migration (Frederiksen et al. 2012; Iverson et al. 2023). This multi-faceted approach is essential for effective conservation management, particularly in identifying marine protected areas and mitigating threats like bycatch and habitat degradation (Soanes et al. 2014; Dean et al. 2015).

Handheld GPS devices can also be used for ground surveys in colonies. Table 1 represents the results from a questionnaire for understanding how data is currently collected among the network of researchers studying seabirds and mercury pollution (ARCTOX<sup>4</sup> network) (Boeijkens 2023). These results showed that while some researchers still manually mark nest locations on a map with a pencil spatial data was mostly collected with geolocator devices such as Garmin GPS, GLS, GPS-UHF and field computers Yuma, Husky and GeTac and transferred to spreadsheet tables manually.

### 1.3 Available Software

For ground surveys general purpose mobile GIS applications can be used as valuable tools for field data collection. These applications allow researchers to collect and input data directly in the field, enhancing the efficiency of data collection (Sarbazvatan and Karimi 2023). For example, the ArcGIS Collector<sup>5</sup> (that seems to be discontinued in 2024) allegedly had an interface for recording site locations and entering attribute data, which then was instantly uploaded to a central database (Lindsay and Kong 2020; Nowak et al. 2020). Currently a plethora of biodiversity recording applications have the ability to record location data of different species (Table 2). Such real-time data collection capability is particularly

---

<sup>4</sup> <https://arctox.wordpress.com/>

<sup>5</sup> <https://doc.arcgis.com/en/collector-classic/>

beneficial for monitoring breeding success and nest locations over time. However, none of the responders from the ARCTOX group collected nest coordinates using a smartphone with any of these applications (see Table 1). Hence, either many researchers prefer to use only specialized equipment to collect spatial data or they find the functionality of these applications (see Table 2) insufficient or there is no certainty if a closed source app will still be available in the next year (for an outdated list for Table 2 see Novak et al. 2020). Nevertheless, the wider use of mobile applications could make equipment for field work cheaper and data collection more streamlined. Yet as far as I know, there is no specifically designed open-source smartphone application currently available for data collection from seabird colonies. The development of such GIS tools would allow researchers and conservationists with limited resources to more easily engage in data collection and analysis (Mobasher et al. 2020). Open-source platforms can be customized for specific research needs, making them adaptable for various ecological studies, including those focused on avian nesting. Django Bird Colony<sup>6</sup> is an open source Django<sup>7</sup> application for managing bird colonies with SQL backend. Still, it lacks GIS functionality and is developed for aviaries making its use in field conditions hard without extensive rewriting.

Given the limitations of existing tools and the specific needs of seabird colony research, there is a clear demand for a specialized, open-source application tailored for field data collection in seabird colonies. While general-purpose GIS applications and biodiversity recording tools exist, they often lack the specific features required for comprehensive seabird colony monitoring. The absence of a dedicated smartphone application for this purpose, as evidenced by the ARCTOX group's reliance on specialized equipment, highlights a significant gap in the available toolset. A custom-built application that combines GIS functionality with specific features for tracking nest locations, breeding success, and individual bird data would streamline the data collection process, improve accuracy, and enable real-time data sharing among researchers. Such a tool would not only make fieldwork more efficient but also more accessible to researchers with limited resources, potentially expanding the scope and scale of seabird colony studies worldwide.

---

<sup>6</sup> <https://github.com/melizalab/django-bird-colony>

<sup>7</sup> <https://www.djangoproject.com/>

Table 1. A summary of questionnaire responses from researchers participating in the ARCTOX network adopted from Boeijkens (2023) (studying the effects of mercury pollution on seabirds). The projects are sorted based on the length of time period of monitoring program, starting with longer programs.

Studied species	Collected data	Location data collection method	Years studied	Data storage method	Data availability
Atlantic puffin ( <i>Fratercula arctica</i> ) and other seabird species	Nest GPS coordinates, weight of adults, eggs and chicks, blood samples	Before 2010: handheld GPS*, after: field work tablets (Husky, Yuma, GeTac)	42	On paper, but most data later reached Excel or Access databases	SEAPOP, SEATRACK and ICES databases, partly available upon request
Skuas ( <i>Catharacta sp.</i> ), gulls ( <i>Larus sp.</i> ), eiders ( <i>Somateria sp.</i> )	Nest GPS coordinates, weight of adults, eggs and chicks, blood samples	GPS	>30	Excel	Monitoring data available in Movebank database
Common eider ( <i>Somateria mollissima</i> )	Nest GPS coordinates, weight of adults, blood samples	GPS and GLS** loggers	28	Paper and Excel	GLS data available in SEAPOP database
Fulmars ( <i>Fulmar sp.</i> )	Nest GPS coordinates, weight of adults, eggs and chicks, blood samples	GPS, marking nests on maps by hand	22	MS Access	Dryad, if published
Common murre ( <i>Uria aalge</i> ), cormorants ( <i>Phalacrocorax sp.</i> )	Breeding success	GLS, photos	21	Paper, Excel, cloud storage	Yearly reports
Common eider ( <i>Somateria mollissima</i> ), glaucous gull ( <i>Larus hyperboreus</i> ), black-legged kittiwake ( <i>Rissa tridactyla</i> )	Nest GPS coordinates, weight of adults, blood samples	GPS coordinates for glaucous gull nests	<20	Excel	Some of the data is published, database building in progress

Studied species	Collected data	Location data collection method	Years studied	Data storage method	Data availability
Little auk ( <i>Alle alle</i> )	Nest GPS coordinates, weight of adults and chicks, blood samples	Handheld GPS	20	Excel, CSV	Data will be made available after publishing related papers (e.g. in Dryad)
Little auk ( <i>Alle alle</i> )	Nest GPS coordinates, weight of adults, eggs and chicks, blood samples	GPS (Garmin)	18	Paper, Excel	Data private, available upon request
black-legged kittiwake ( <i>Rissa tridactyla</i> ), Atlantic puffin ( <i>Fratercula arctica</i> )	Nest GPS coordinates, weight of eggs and chicks, blood samples	GPS in field work tablet (Yuma)	17	Excel and databases	Summary of data available on the SEAPOP page
Many seabird species (SEAPOP program)	Weight of adults and chicks, blood samples	Marking nests on maps by hand	12-14	Excel	Some data in Dryad, mostly private
Arctic tern ( <i>Sterna paradisaea</i> )	Nest GPS coordinates, weight of adults and chicks	GPS	13	Excel	Data is private
5 different seabird species	Nest GPS coordinates, weight of adults, blood samples	Handheld GPS (Garmin)	12	Excel, Access	Part of the data published every year on a public web page
Glaucous gull ( <i>Larus hyperboreus</i> )	Nest GPS coordinates, weight of adults and chicks, blood samples	GPS	12	Excel, database	Detailed data is private

Studied species	Collected data	Location data collection method	Years studied	Data storage method	Data availability
6 different seabird species	Nest GPS coordinates, weight of adults and chicks, blood samples	GPS, loggers	10	Excel	Regional database
Common eider ( <i>Somateria mollissima</i> )	Nest GPS coordinates, weight of adults	GPS (Garmin)	10	Excel	Data is not published, but has been sent to relevant authorities
Common edier ( <i>Somateria mollissima</i> ), common murre ( <i>Uria aalge</i> ), northern fulmar ( <i>Fulmarus glacialis</i> )	Weight of adults	GLS	8	Excel	Data will be published via publications
European herring gull ( <i>Larus argentatus</i> ), lesser black-backed gull ( <i>Larus fuscus</i> ), great black-backed gull ( <i>Larus marinus</i> )	Nest GPS coordinates, weight of adults, eggs and chicks, blood samples	GPS-UHF*** loggers	7	Files in a computer	Data is private
Bulwer's petrel ( <i>Bulweria bulwerii</i> )	Weight of adults and chicks, blood samples	Marking nests on maps by hand	4	Excel	Data publicly available in Barcelona University repository
Little auk ( <i>Alle alle</i> ), thick-billed murre ( <i>Uria lomvia</i> ), black-legged kittiwake ( <i>Rissa tridactyla</i> )	Nest GPS coordinates, weight of adults	Marking nests on photos, GPS (Garmin)	2	Excel, back-up hard drive, SQL database	Data published in SeaTrack webpage

Table 2. List of available applications that are designed for gathering biodiversity data, including observations, specimens, and material samples.

<b>Name</b>	<b>Description</b>	<b>Features</b>	<b>Platform</b>	<b>Link</b>	<b>Open source</b>
<b>PlutoF GO</b>	A tool for gathering biodiversity data, including observations, specimens, and material samples.	Record photos, videos, and sounds, Offline taxonomy search, Upload data to PlutoF workbench	iOS, Android	<a href="#">An-PlutoF GO</a>	No
<b>iNaturalist</b>	A community-driven app for recording and sharing biodiversity observations.	Species identification, Community validation, Data export for research	iOS, Android, Web	<a href="#">iNaturalist</a>	<a href="#">Yes</a>
<b>eBird</b>	A bird observation app by the Cornell Lab of Ornithology.	Bird species identification, Real-time data entry, Explore birding hotspots	iOS, Android, Web	<a href="#">eBird</a>	No
<b>ObsMapp</b>	An app for recording wildlife observations, linked to the Observation.org platform.	Offline data entry, GPS tracking, Integration with Observation.org	Android	<a href="#">ObsMapp</a>	No
<b>Nature's Notebook</b>	An app for tracking plant and animal life cycle events.	Phenology data collection, Data visualization, Community collaboration	iOS, Android, Web	<a href="#">Nature's Notebook</a>	No
<b>Map of Life</b>	An app for exploring species distributions and recording observations.	Species distribution maps, Offline data entry, Integration with global biodiversity databases	iOS, Android	<a href="#">Map of Life</a>	<a href="#">Yes</a>
<b>HerpMapper</b>	An app for recording reptile and amphibian observations.	Species identification, Data sharing with conservation organizations, Offline data entry	iOS, Android	<a href="#">HerpMapper</a>	No

## 2 Technological Choices for Bird Colony Application

The development framework and database for creating an application aimed at collecting spatial and fitness data from seabird colonies was carefully considered. I wrote a preliminary version of the software<sup>8</sup> for the Bird colony app (then called Kakrarahu app) back in 2017 in the field using the Ionic<sup>9</sup> framework frontend and MariaDB<sup>10</sup> (SQL) relational database with Laravel<sup>11</sup> php backend. Before rewriting a more general and feature-rich application, these choices were evaluated again together with Johan Boeijkens (then a Bachelor student supervised by me and Tuul Sepp) in the light of current technological advancements and project requirements. Among the options reviewed, Laravel with MariaDB was also considered due to its robustness and familiarity. However, Flutter<sup>12</sup> emerged as a more compelling choice due to its performance, development efficiency, and cross-platform capabilities. For data storage, Firebase<sup>13</sup> was selected for its Flutter integrations, real-time database features, seamless integration with mobile platforms, and scalability. This combination of Flutter and Firebase enabled to create a robust, efficient, and user-friendly application capable of meeting the complex demands of seabird colony research while ensuring adaptability for future enhancements.

### 2.1 Development Framework

Flutter's ability to significantly reduce development time and complexity is one of its primary advantages (Katti and Sowmya 2022). By enabling developers to create applications for both iOS and Android from a single codebase, Flutter eliminates the need to write and maintain separate versions of an app for different platforms, thereby saving substantial time and resources. Such efficiency was particularly beneficial in enhancing this application in the field. In the initial development phase, functionality updates and bug fixes were implemented swiftly, even during fieldwork days. Flutter simplifies the app development process,

---

<sup>8</sup> <https://bitbucket.org/zoobiologysoft/workspace/repositories/>

<sup>9</sup> <https://ionicframework.com/>

<sup>10</sup> <https://mariadb.org/>

<sup>11</sup> <https://laravel.com/>

<sup>12</sup> <https://flutter.dev/>

<sup>13</sup> <https://firebase.google.com/>

making it accessible to developers with backgrounds in languages like Java, C#, or C++ (Cheon and Chavez 2021). Unlike other cross-platform frameworks such as React Native<sup>14</sup>, which rely heavily on JavaScript and require a bridge to communicate with native components, Flutter compiles directly to native code using its own programming language, Dart (Katti and Sowmya 2022). This direct execution model enhances performance, ensuring that Flutter applications run smoothly—a critical factor for real-time data collection and analysis in field conditions.

Another advantage of Flutter is that its widget-based UI library empowers developers to craft highly responsive and smooth applications that deliver consistent experiences across various devices (Aung et al. 2024). This consistency is important in fieldwork scenarios where different fieldwork assistants may use different devices, as it ensures the application maintains a uniform look and feel across both iOS and Android platforms (Katti and Sowmya 2022). A well-designed user interface facilitates easier data entry and retrieval, which is essential when working in harsh field conditions with weather conditions varying on a daily basis.

Moreover, Flutter's extensive widget library allows developers to create highly customizable user interfaces that adapt seamlessly to various screen sizes and resolutions—a feature particularly valuable if the data is entered on the field from a smartphone but reviewed later in the station on a laptop (Aung et al. 2024). Another significant advantage of Flutter is its strong community support and comprehensive documentation, which accelerates the development process. As an open-source framework, Flutter provides developers with access to a wealth of resources, including libraries and plugins that enhance application functionality (Ameen and Mohammed 2022). This abundance of community-driven resources is particularly beneficial for applications requiring integration with various data sources or external APIs. It facilitates the implementation of features such as geolocation tracking or data visualization—capabilities often used in ecological research applications focused on tasks like nest or experiment management.

When compared to other cross-platform frameworks like React Native and Ionic, Flutter stands out for its superior performance and ease of use. React Native relies heavily on JavaScript and requires a bridge to communicate with native components, which can

---

<sup>14</sup> <https://reactnative.dev/>

complicate the development process and increase resource use(Oliveira et al. 2023). Ionic, on the other hand, builds applications using web technologies like HTML, CSS, and JavaScript, running them within a WebView (Novac et al. 2022). In the comparison between the 3 frameworks, Flutter imposes the least overhead in execution time and energy, whereas React Native and Ionic impose higher overheads due to their reliance on JavaScript and WebView rendering, respectively (Oliveira et al. 2023). Flutter's use of Dart allows for efficient native code compilation, eliminating the need for a bridge or WebView and streamlining the development workflow (Cheon and Chavez 2021). This single codebase approach and direct compilation to native code enhance performance and simplify updates.

Although the aforementioned technical merits of Flutter played a role in its selection, it is important to note that personal preference and prior experience were also key factors in this decision. My prior experience with Ionic had introduced me to the cross-platform development. This experience, coupled with Flutter's seemingly growing popularity and positive developer feedback, influenced my inclination towards Flutter. The familiarity with cross-platform development concepts from Ionic made the transition to Flutter feel natural, while Flutter's performance improvements, rich widget library and simple integration with Google Firebase backend offered exciting new possibilities for creating a new and better field research application.

## **2.2 Data Storage**

When considering the backend infrastructure for a Flutter application designed to collect spatial and fitness data from seabird colonies, the choice is mainly between SQL and noSQL databases (Sudiartha et al. 2020). Google Firebase Firestore, a noSQL solution, presents some advantages like schema flexibility and scalability over traditional relational databases like MySQL (Fotache and Cogean 2013; Khan et al. 2023). Also, Firestore provides a noSQL data storage that allows for real-time data synchronization across all connected clients (Sukmana and Rosmansyah 2021). This feature is particularly beneficial for applications that require immediate updates and feedback. In situations where data is collected in the field, real-time updates can facilitate immediate data sharing among data submitters, allowing for more dynamic and informed decision-making.

Firestore also excels in providing robust data security and privacy features. The platform includes built-in security rules that allow developers to define access controls at a granular level, ensuring that only authorized users can access or modify data (Chougale et al. 2021).

Firestore also simplifies user authentication, providing a secure and efficient way to manage user accounts (Beño, Kučera, and Bašista 2024).

The scalability of Firestore is another factor that makes it an attractive option for Flutter applications. Firestore's architecture is designed to support high volumes of concurrent users and data transactions without requiring extensive manual configuration or optimization (Sudiartha et al. 2020). This scalability is needed for applications that may experience fluctuating user loads, such as during peak data collection periods when multiple researchers are actively using the app (Khan et al. 2023). Firestore automatically scales to accommodate increased traffic (Kesavan et al. 2023), whereas scaling a SQL backend often involves managing server resources and may require complex configurations, such as load balancing or database replication (Uzzaman et al. 2024).

Furthermore, Firestore's serverless architecture allows developers to deploy and manage the application without worrying about the underlying infrastructure or cost of setting up and running an app with a small user base (Sukmana and Rosmansyah 2021; Kesavan et al. 2023). This is particularly advantageous for teams with limited resources or expertise in server management, enabling virtually maintenance-free use of the app for ecologists after the initial setup.

Firestore also supports cloud messaging, which can be used in keeping users engaged with timely notifications and updates (Kesavan et al. 2023). This feature allows developers to send push notifications to users, informing them of important events or updates related to their research activities (Patnaik et al. 2021). For instance, researchers can receive alerts when new data is available or when significant changes occur in the monitored seabird colonies. Another significant advantage of using Firestore is its comprehensive cloud storage capabilities. Firestore Storage allows developers to store and serve user-generated content, such as images, videos, and other files, securely and efficiently (Patnaik et al. 2021). This is particularly relevant for applications that may require the upload of large datasets or multimedia content, such as images of particular experimental nests or birds. The ability to handle large files without compromising performance is crucial in maintaining a smooth user experience, especially in mobile applications where bandwidth may be limited (Beño, Kučera, and Bašista 2024).

Furthermore, Firestore's easy integration with Android Studio allows developers to quickly implement functionalities using pre-built methods provided by Firestore service SDKs

(Bačanin Džakula 2024). Firebase's extensive documentation<sup>15</sup> and community<sup>16</sup> support can significantly reduce the learning curve for new developers.

The combination of Flutter and Firebase for the Bird Colony application represents a choice that addresses the unique challenges of seabird colony research. Flutter's cross-platform capabilities, performance, and rich UI features provide an ideal foundation for creating a user-friendly and efficient field data collection tool. Meanwhile, Firebase's real-time database, scalability, and robust security features ensure that the application can handle the complex data management needs. This technological pairing not only streamlines the development process but also enhances the application's ability to adapt to diverse field conditions and evolving research requirements. By leveraging the strengths of both Flutter and Firebase, the Bird Colony application is well-positioned to significantly improve the efficiency and accuracy of data collection in seabird colonies, potentially transforming how researchers conduct and collaborate on ecological studies in challenging field environments.

---

<sup>15</sup> <https://firebase.google.com/docs/>

<sup>16</sup> <https://firebase.google.com/community>

### 3 Overview of the Developed Software

To collect and analyse standardized data across seabird colonies, and to facilitate spatial data collection, nest- and individual-based monitoring data recording, and experimental work in seabird colonies, I completely rewrote the Bird Colony application in 2024<sup>17</sup>. This work was based on a proof of concept application<sup>18</sup> I created back in 2017 that was developed into a preliminary Flutter application (Boeijkens 2023). Bird Colony is a Flutter application using Firebase as its backend. The application allows simultaneous work and data collection for a group of researchers working on different tasks within the colony using their cell phones, automated storage of data to cloud, and adjustments for different settings. The application<sup>19</sup> allows users to create and manage nests, birds, eggs, and experiments, and view basic statistics related to ringing or nests. The app also includes a map feature for viewing nests on a map. Here, I describe this toolset and demonstrate its application in a seabird colony located on an islet near the Western coast of Estonia, where the application has been tested and developed simultaneously for several years for monitoring 1200+ common gull (*Larus canus*) nests and for conducting experimental work with gulls e.g. (Carbillet, Saks, and Sepp 2024), but also for recording location and banding data for other seabird species breeding on the islet. The full source code of the application is publicly available in Github<sup>20</sup>. The software is licensed under GPL-3.0<sup>21</sup> license. New features and app improvements can be added during the coming years according to the suggestions from the users. As the software has a permissive license tweaking for particular unseen use cases by other developers is also possible, given that the complete source code of licensed works and modifications is made public.

#### 3.1 Backend Architecture

The architecture of the Bird Colony Management backend in Firebase is structured around Firestore collections, organizing data into key entities such as users, nests, birds, eggs,

---

<sup>17</sup> [https://github.com/rix133/flutter\\_bird\\_colony/graphs/contributors](https://github.com/rix133/flutter_bird_colony/graphs/contributors)

<sup>18</sup> <https://bitbucket.org/zoobiologysoft/workspace/repositories/>

<sup>19</sup> <https://managebirdcolony.web.app>

<sup>20</sup> [https://github.com/rix133/flutter\\_bird\\_colony](https://github.com/rix133/flutter_bird_colony)

<sup>21</sup> [https://github.com/rix133/flutter\\_bird\\_colony/blob/main/LICENSE](https://github.com/rix133/flutter_bird_colony/blob/main/LICENSE)

experiments, and default settings. Each entity is represented as a collection, and documents within these collections store relevant attributes. The intended relationships between main collections are outlined in Figure 1. The "Birds" collection includes fields such as bird IDs, nesting locations, ring information, species, and associated experiments, while the "Nests" collection tracks nest IDs, species, accuracy of location, and discovery dates. Eggs are stored as part of the "Eggs" collection under each nest object, capturing details like the discovery date, ring information, and status. Experiments are a separate collection that can be linked with birds or nests. While the "Birds" collection is something that has new birds entered each year, for code simplicity and performance reasons as well as to maintain backward compatibility with data gathered in 2022, a new "Nests" collection is created for each breeding season that holds the nests for the given year. The data model also supports real-time updates and historical changes, with changelogs stored within each document (such as birds, nests, eggs and experiments) to provide a detailed history of modifications. Additionally, collections like "DefaultSettings" store key configuration data, including default location settings, species preferences, and measurement configurations. The Firestore database schema does not ensure that relationships between entities (e.g., birds and nests or experiments and nests) are clearly maintained. This is the main development bottleneck of the approach. Firebase Authentication manages user access, and Google Maps<sup>22</sup> is integrated to manage spatial data, while Firestore supports real-time synchronization of colony data across devices, ensuring updates are reflected immediately across the platform. Google maps was preferred above other map applications because it integrates well with Flutter and provides a satellite overlay, which is more informative on the field compared to plain maps.

---

<sup>22</sup> <https://developers.google.com/maps>

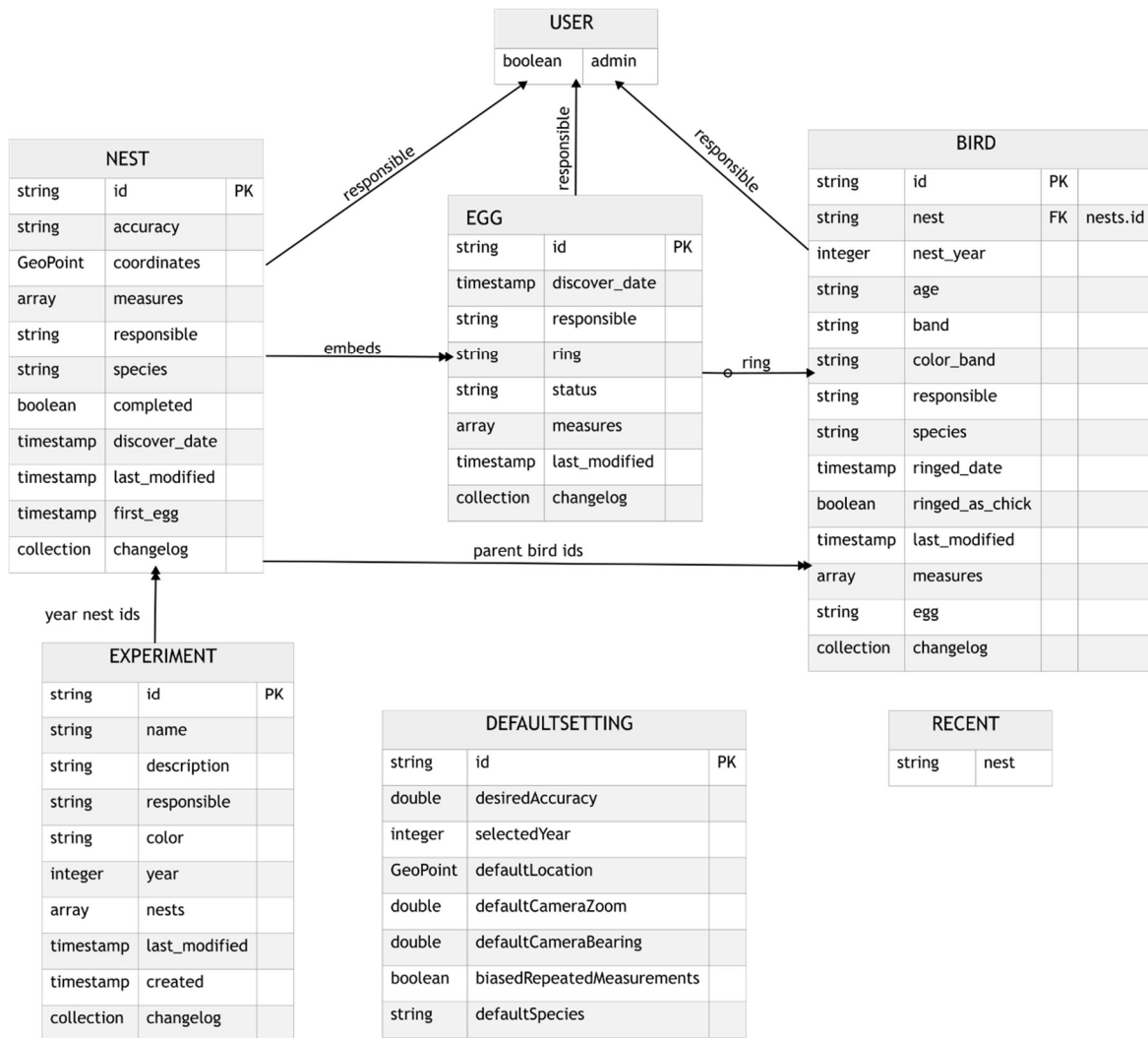


Figure 1. A schematic, non-comprehensive overview of the noSQL Firebase backend. This schema uses arrow notation (double arrow 0:n, single arrow 1:1) and intends to capture the general relationships and structure of how the data is used in the application.

### 3.2 Frontend Architecture

Next, I will lay out the frontend architecture with the features that are available in the current version (2.3.6) of the Bird Colony Flutter application. A 2 minutes usage overview video<sup>23</sup> allows for a glimpse of the application frontend. Most important components of the main user interface are depicted on figure 2. First, the application allows to view nests on a map and create nests from the map. This feature allows marking the precise location of a nest to the map, using GPS coordinates. The accuracy of the nest location assignment will be recorded, and the acceptable accuracy limit can be adjusted according to the characteristics of

<sup>23</sup> <https://youtu.be/SOzjIE1lf70>

the colony/landscape. Nest location accuracy can be improved after first recording during follow-up visits to the nests. Created nests will appear on the application “Map” section as location signs, the colors of which can be changed according to whether the nest has been checked over required intervals, whether the nest belongs to an experiment or not, and other fine-tuned characteristics of the nest. Information about the nest (e.g. species, nesting stage, band numbers of parents and chicks, etc.) can be entered and edited. The nests can thereafter be found on the map, which will simplify monitoring the colony. The colour of the nest mark can be adjusted to change after daily checks, allowing simultaneous colony monitoring with the app running in several field work phones at the same time.

The application allows to create, edit, and list nests. Nests can be added directly on the map during field work with the “Add nest” button. When a nest is added, the nest will appear automatically in all the devices using the app, and the app will recommend the next nest number in line to be entered to the data base to all users. Specific nests can be found with using the “Find nest” button, directing the researcher to a specific location on the map, but also showing information linked with the nest number. Nests can also be filtered, for example, to only show nests linked to an experiment.

As bird nests have parents and eggs, the application allows to monitor and record their status. Egg status is linked to the nests, and can be updated over the needed intervals (e.g. daily nest checks). Recording when the egg was laid, but also egg fate (when and whether the egg is hatching, hatched, predated or lost), can give valuable information about the breeding success of the colony. Nest parents are linked to a bird in the “Bird” collection allowing to record new information on subsequent sightings.

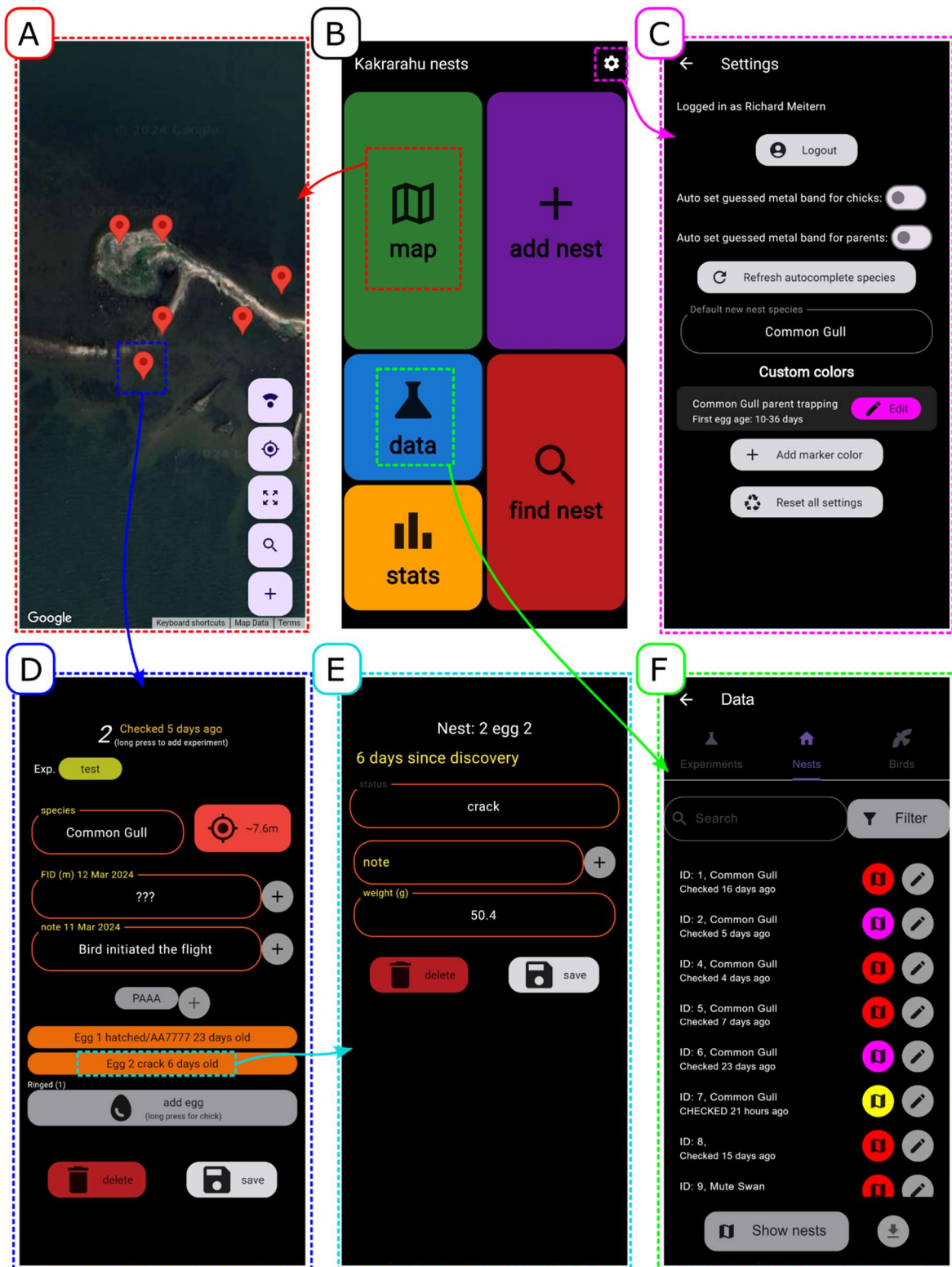


Figure 2. An overview of the main frontend screens. Arrows denote relationships between UI elements and screens. The depicted screens are: map (A), home (B), settings for non-administrative user (C), nest details (D), egg details (E) and nests overview (F)

Under the “Data” button of the app, nests, experiments and birds are listed. Bird details can include, among others, band number, banding date, species, banding person, age, breeding status updated based on nest status, etc. When banding the birds, band numbers will be synchronized between devices using the app, and the next band number will appear as a recommended entry to all devices. This allows synchronous banding data entry. Bird data can also be linked with historical data files about the same bird from the previous field work season. In addition, the Export Data feature allows to export data to an Excel file. This is available to administrator users under the “Data” button in the application.

The Experiment management feature of the application accessible also through the “Data” button allows to add, edit, and list experiments. Experiments can be limited to certain areas of the colony, or to specific listed nests. Several experiments can be running on the colony and application simultaneously. Experimental nests can be searched from the “Find nest” feature, and will appear on the map with designated color.

The Statistics feature allows to view basic ringing statistics related to the colony for the ongoing, but also previous/historical years, if responding databases are linked with the application. Statistics can be filtered based on year, day, and users. Information based on nests of different species, or based on banded birds of different species, will appear on the map, or as lists.

Finally, the User management feature allows to sign in and out, and manage users. Different status of users can be assigned, with administrator users having more rights than regular users. The person who creates a Firebase project for a specific colony will automatically become the administrator, and will have the rights to add and manage users to their project.

### **3.3 New Instance Setup**

To set up the Bird Colony application for use in a specific seabird colony, there are two main approaches outlined more clearly in the project *readme.md*<sup>24</sup>. In short, for those ecologists unfamiliar with programming the simplest option is to contact me, so I can configure the app for their particular colony. This option allows users to download and use the app directly from the Google Play Store<sup>25</sup> without requiring technical setup. Alternatively, a web version of the application is available for them at <https://managebirdcolony.web.app>

---

<sup>24</sup> [https://github.com/rix133/flutter\\_bird\\_colony/blob/main/README.md](https://github.com/rix133/flutter_bird_colony/blob/main/README.md)

<sup>25</sup> [https://play.google.com/store/apps/details?id=ee.ut.adapt.flutter\\_bird\\_colony](https://play.google.com/store/apps/details?id=ee.ut.adapt.flutter_bird_colony)

providing access to the app's functionalities from any device with an internet connection. This method is suitable for users who prefer a straightforward setup process without dealing with the technical intricacies of configuring Firebase and Flutter.

For those who prefer an independent setup or wish to customize the app, the process involves several steps and some familiarity with Flutter and Firebase. After installing Flutter and its command-line tools, users need to clone the project repository and install dependencies using *flutter pub get*. A Firebase project must then be created, and the *google-services.json* file should be added to the *android/app* directory to link the app with the Firebase backend. Additionally, the *firebase\_options.dart* file is generated using Firebase tools and placed in the `'lib'` directory. Users must also configure the *key.properties* file to include release key information and the Google Maps API key for map functionalities.

Additional setup steps include configuring Firestore indexing for the `'egg'` collection and implementing Firestore security rules to restrict access to authorized users. The app also relies on various dependencies such as *geolocator*, *firebase\_core*, *cloud\_firestore*, *firebase\_auth*, and *google\_maps\_flutter*, making proper configuration of these packages essential for a fully functional app. This independent setup offers flexibility and full control over the app's environment, making it ideal for advanced users or developers looking to integrate additional features or modify the existing toolset.

### 3.4 Formal Testing

To ensure the app's functionality and reliability, a set of test cases has been developed to obtain code coverage of about 90%<sup>26</sup>. These tests can be executed using the 'flutter test' command. The project also utilizes Github Actions to ensure that tests are automatically run and code coverage is monitored with each update to the codebase<sup>27</sup>. As can be verified from the git history, tests were added after the initial development to get to 90% of line coverage. Subsequent bugs were then addressed with a test first development practice. While path coverage is not formally addressed, care was taken to ensure that all of the most frequent usage scenarios in the field are covered by tests. A generalized set of testing scenarios implemented is outlined in Table 3. These scenarios encompass diverse functionalities,

---

<sup>26</sup> [https://app.codecov.io/gh/rix133/flutter\\_bird\\_colony](https://app.codecov.io/gh/rix133/flutter_bird_colony)

<sup>27</sup> [https://github.com/rix133/flutter\\_bird\\_colony/blob/main/.github/workflows/codecov.yml](https://github.com/rix133/flutter_bird_colony/blob/main/.github/workflows/codecov.yml)

including managing experiments (such as loading and editing experiments, customizing experiment details, and navigation to related entities), species management (covering creation, serialization, and filtering of species data), and handling of egg and bird management tasks (including banding, serialization, and deletion functionalities). Additionally, tests cover Excel output for various data types (e.g., experiments, nests, species) and Firestore operations, ensuring that changelog downloads are handled accurately. For the complete set of tests implemented, please refer to the source code available in GitHub.

Table 3: Categorization of test cases for the application, organized by functionality

Category	Test Cases
<b>Nest Management</b>	<ul style="list-style-type: none"> <li>• <u>UI Elements</u>: Display add egg and add parent buttons; Display nest details including species and eggs</li> <li>• <u>Navigation</u>: Navigate to edit egg; Navigate to <i>nest</i> after saving/deleting egg; Navigate after saving/deleting nest (with or without validation errors); Go to edit bird when add parent/egg is pressed/long pressed</li> <li>• <u>Notes Management</u>: Add new note on egg and nest when requested</li> </ul>
<b>Egg Management</b>	<ul style="list-style-type: none"> <li>• <u>Egg Model Tests</u>: Egg creation with required parameters; Egg ringed status based on ring value; Egg serialization to JSON; Egg knownOrder determination; Egg type determination; Egg getAgeRow method; Egg statusText for various statuses</li> <li>• <u>Nest Interaction</u>: Navigate to edit egg from nest; Navigate to nest after saving/deleting egg; Add new note on egg when requested</li> <li>• <u>Band Management</u>: Allow editing band on egg; Set next band on egg; Update nest chick egg when bird band is changed; Update nest egg ring when bird band is changed</li> <li>• <u>Deletion</u>: Delete bird artifacts from egg (chick/egg artifacts)</li> </ul>
<b>Bird Management</b>	<ul style="list-style-type: none"> <li>• <u>Loading Birds</u>: Load edit bird without arguments; Load with nest and egg/parent; Load with bird; Load default bird when not found; Load bird without band; Display metal band</li> <li>• <u>Editing Birds</u>: Allow edit band on egg; Change bird species; Show/hide measures after band input</li> <li>• <u>Saving Birds</u>: Set next band on egg; Save next band to local storage; Save color band on existing/new parent; Save nest on new parent after alert dialog;</li> </ul>

Category	Test Cases
	<p>Update parent nest and year after alert dialog; Handle not saving new parent if band exists; Update color band on nest parent; Raise alert dialog when overwriting existing bird</p> <ul style="list-style-type: none"> <li>• <u>Deleting Birds</u>: Delete bird and associated artifacts (chick bird artifacts from egg, egg bird artifacts from egg, parent bird artifacts from nest)</li> <li>• <u>Changing Bird Bands</u>: When bird band is changed, handle deletion of old bird and saving of new bird; Update nest parents and chick egg accordingly; Update nest egg ring</li> <li>• <u>Adding Notes</u>: Add note to bird saved with default settings or auto-assign next bands</li> </ul>
<b>Map Markers</b>	<ul style="list-style-type: none"> <li>• <u>Model Tests</u>: Create MarkerColorGroup with magenta color; Convert MarkerColorGroup to/from JSON; Get MarkerColorGroup from JSON when color is int; Update color when setColor is called; Return correct color when getColor is called</li> </ul>
<b>Experiment Management</b>	<ul style="list-style-type: none"> <li>• <u>Experiment Dropdown Widget</u>: ExperimentDropdown widget functionality</li> <li>• <u>Loading Experiments</u>: Load new experiment; Edit existing experiment; Change experiment type to bird</li> <li>• <u>Experiment Modifications</u>: Add new nest to experiment; Add new bird to experiment; Remove existing nest from experiment; Delete existing experiment; Save experiment measures to Firestore</li> <li>• <u>Experiment Customization</u>: Pick experiment color; Add new measure to existing/new experiment</li> <li>• <u>Navigation</u>: Go to nest editing from experiment; Go to bird editing from experiment</li> </ul>
<b>Species Management</b>	<ul style="list-style-type: none"> <li>• <u>Species Model Tests</u>: Species creation from English name; Species equality and inequality; Species band letters (including unknown species); Species serialization to/from JSON; Species created_date property handling; Species delete method behavior; Species changeLog returns sorted list; Get Species.empty when input is null</li> <li>• <u>Local Species List Tests</u>: Creation and copy methods; Name properties (local vs. English name); Get species by English name (case insensitive); Get species by species object</li> </ul>

Category	Test Cases
	<ul style="list-style-type: none"> <li>• <u>UI Tests</u>: ListSpecies displays species list; Navigates to edit species on tap; Filters species by name; Opens/closes filter dialog; Shows alert dialog when list tile is tapped</li> </ul>
Defaults	<ul style="list-style-type: none"> <li>• <u>Default Camera Position</u>: Update default camera position in Firestore when default settings are saved; Update default camera position in SharedPreferencesService when default settings are saved</li> </ul>
Updates	

### 3.5 Real World Usage

The current software was tested from April to June 2024 for monitoring a seabird breeding colony located on Kakrarahu islet in Matsalu National Park, on the west coast of Estonia (58°46' N, 23°26' E). Each year, several seabird species are nesting on the islet, including common gulls, herring gulls (*Larus argentatus*), mute swans (*Cygnus olor*), common (*Sterna hirundo*) and arctic (*Sterna paradisaea*) terns, and a few great cormorants (*Phalacrocorax carbo*). The main target species for the studies on the islet has been common gulls, who have been monitored on the islet for nearly 60 years, and due to their site fidelity and as a result of continuous banding of the hatching chicks, the exact ages of many of the birds in the colony are known, as is the pedigree data and breeding success for the previous years for each bird.

During the field season, researchers walk through each section of the islet using line transects to screen the colony. Each new nest is assigned a unique ID and marked with a wooden pole or rock, while the app records the precise GPS location and metadata, including the laying date and egg status. Eggs in common gull nests are monitored throughout the season, and their status is updated using predefined categories such as “crack,” “small hole,” “medium hole,” “big hole,” “hatched,” “lost,” or “predated.” Custom labeling is also allowed for more detailed notes if needed. This detailed tracking would not be possible without the app’s ability to handle data through classes like Nest, Egg, and Species, which are structured to store a variety of attributes and status information.

In cases where parents are already banded and can be identified, their IDs are linked to the nest within the app, providing an extensive database for pedigree analysis. When chicks hatch, they are banded on the first day, and the application immediately updates the nest record with the new band ID. The editing screens are central to handling these updates,

leveraging the services to ensure seamless synchronization between local changes and the central Firestore database.

In this colony, the numbers of breeding common gulls, but also other breeding bird species, have increased continuously over the decades. For example, when in 1970s the number of common gull nests on the islet were around a hundred, it increased to several hundreds in the 00s and is now more than 1200. At the same time, extreme weather events have become more common, resulting with large fluctuations of breeding success of common gulls on the islet. In parallel to the changes in climate competition within and between species has increased dramatically in the colony. All these phenomena, linked possibly with anthropogenic environmental change, can be monitored to record fitness effects, if the tools are available to cover this large colony. While field work diaries were used for monitoring the colony until 2018, it was very apparent that this method does not allow to cover the whole islet, and while the whole colony was monitored in 80s and 90s, starting from 2000s, more and more nests fell out of the continuous monitoring scheme. With the introduction of the software, it has been again possible to cover the whole colony, to understand the impact of environmental conditions on the breeding success and survival of seabirds. Moreover, in addition to common gulls, other species' nests have also been marked more efficiently.

In 2024 the application was tested for usage in the abovementioned islet. Two android smartphones (Doogee S86 and DOOGEE S89) were used for testing (some other iOS and Android devices were also used occasionally). Ten different individuals participated in testing. These participants included both experienced and novice fieldwork assistants. Issues encountered during testing were reported either through private communication or directly added to GitHub as issues by the user. I then classified and addressed the issues as needed within a few days (as can be seen from the git commit log). Occasionally I also participated in fieldwork, recording any problems encountered during real-world use and identifying potential enhancements to improve the application's performance and usability. This iterative testing and feedback process ensured the application was well-adapted for its intended field environment and robust against diverse usage scenarios. In 2024 a total of 2121 nest positions were recorded for different bird species (1253 common gulls, 25 herring gulls, 55 mute swans, 242 common and 384 arctic terns, and others species) with the application. In comparison, in 2022 and 2023 in total 1326 (1205 common gull) and 1540 (1121 common gull) nest positions were recorded respectively. Although field conditions are somewhat

different each year, it is plausible that the most recent rewrite of the application in 2024 also enabled researches to be more efficient on the islet in 2024 than in previous years.

### 3.6 Future Directions

While the developed Bird Colony Management application has proven useful in real world contexts, there are numerous issues for enhancements already raised in Github<sup>28</sup>. Future work should concentrate on expanding the application's features to enhance user experience, data accuracy, and overall functionality as well as making data available for a wider research community.

The to be added app features should include implementing advanced search options like a nearby nest search and a "Find Bird" feature to help users access parent data more efficiently. Also developing tools such as *data correction* interfaces and a *restore nest or egg previous state* features would enable users to maintain data integrity and streamline accidental data input errors fixes. Enhancements to database robustness and the introduction of custom map overlays would improve performance and usability, especially in areas with insufficient satellite imagery. Enabling offline functionality would allow the app to operate without internet access, syncing data when connectivity is restored. Furthermore, integrating the option to add images to nests, birds, and eggs would significantly enhance the application's utility and user engagement. By allowing users to upload and associate photos with specific data entries, the app would provide a way to design experiments where visual data input would be easier. This feature would be particularly beneficial for tracking individual animals over time, analysing egg texture variability, studying morphological changes, and sharing observations with the research community. However, the current design is optimized for cost-effectiveness, and adding features that require more bandwidth and storage capacity would possibly increase the price for the application owner.

In addition, developing an automated system to make data openly available on platforms like PlutoF<sup>29</sup> or GBIF<sup>30</sup> would promote data sharing and collaboration within the scientific community. By implementing data export functionalities or integrating APIs, the application could seamlessly transfer collected data to these global biodiversity databases. This

---

<sup>28</sup> [https://github.com/rix133/flutter\\_bird\\_colony/issues](https://github.com/rix133/flutter_bird_colony/issues)

<sup>29</sup> <https://plutof.ut.ee/>

<sup>30</sup> <https://www.gbif.org/>

integration would not only increase the visibility and accessibility of the data but also contribute to larger-scale ecological studies, biodiversity assessments, and conservation efforts worldwide. Automating this process ensures consistent data updates, reduces manual workload, and fosters a more collaborative and efficient research environment by adhering to open data principles.

## 4 Conclusions

This thesis introduces the Bird Colony application—a standardized, user-friendly tool designed to streamline data collection and management in seabird colony monitoring. Through a review of traditional and modern data collection methods, it was identified that while advancements like UAVs and GPS tracking have improved monitoring capabilities, they often have limitations such as high costs, technical complexities, and data fragmentation. Existing software tools may lack specific functionalities required for seabird research or are not widely adopted due to accessibility issues.

The development of the Bird Colony application using Flutter and Firebase provides a tool for seabird ecologist that can potentially enhance their fieldwork efficiency. Flutter's cross-platform capabilities and user-friendly interface ensure a consistent experience across devices, while Firebase's real-time database and secure data storage facilitate immediate data synchronization among researchers and fieldwork assistants. The application's functionalities—including precise nest mapping, real-time data entry, and experiment management—aim to address the specific needs of seabird colony monitoring.

Field testing on Kakrarahu islet has demonstrated the application's practical utility. The Bird Colony app enabled comprehensive monitoring of over 1,000 common gull nests in 2024, improving data coverage compared to previous years. This enhanced data collection capability is useful for assessing breeding success, egg survival rates, and responses to environmental stressors, particularly in the context of environmental changes impacting seabird populations.

By facilitating standardized data collection and promoting data sharing, the Bird Colony application has the potential to benefit seabird research. It addresses some limitations of fragmented datasets and specialized equipment, potentially making comprehensive monitoring more accessible to researchers with varying resources. The application's open-source nature allows for customization and collaborative improvements, which may foster a community-driven approach in the future.

The Bird Colony application represents a step forward in seabird monitoring and conservation efforts. Its adoption could lead to improved recording practices, standardized data analysis, and better accessibility to individual-specific spatial data. Future work should focus on addressing user provided feature requests, especially aiming to enhance data accuracy and

openness. Some effort is needed to introduce a way to make data collected with the application publicly available in an open database.

In conclusion, the Bird Colony application contributes to the field of ecological research and conservation by providing a tool to streamline data collection and encourage collaboration. By improving the efficiency of seabird monitoring, it may assist in the broader goals of biodiversity conservation. Continued development and adoption of this tool could have a positive impact on understanding and preserving seabird populations in the context of environmental changes.

## References

- Ameen, Siddeeq Y., and Dilkhaz Y. Mohammed. 2022. "Developing Cross-Platform Library Using Flutter." *European Journal of Engineering and Technology Research* 7 (2): 18–21. <https://doi.org/10.24018/ejeng.2022.7.2.2740>.
- Arneill, Gavin E., Emma Jane Critchley, Saskia Wischnewski, Mark J. Jessopp, and John L. Quinn. 2020. "Acoustic Activity across a Seabird Colony Reflects Patterns of Within-Colony Flight Rather than Nest Density." *Ibis* 162 (2): 416–28. <https://doi.org/10.1111/ibi.12740>.
- Aung, Soe Thandar, Nobuo Funabiki, Lynn Htet Aung, Safira Adine Kinari, Mustika Mentari, and Khaing Hsu Wai. 2024. "A Study of Learning Environment for Initiating Flutter App Development Using Docker." *Information* 15 (4): 191. <https://doi.org/10.3390/info15040191>.
- Baćanin Džakula, Nebojša. 2024. "Agile Multi-user Android Application Development With Firebase: Authentication, Authorization, and Profile Management." *Sinteza 2024 - International Scientific Conference on Information Technology, Computer Science, and Data Science*, 405–12. <https://doi.org/10.15308/Sinteza-2024-405-412>.
- Beňo, Lukáš, Erik Kučera, and Matej Bašista. 2024. "Smart Sleep Monitoring: An Integrated Application for Tracking and Analyzing Babies' Sleep—BabyCare." *Electronics* 13 (21): 4210. <https://doi.org/10.3390/electronics13214210>.
- Bernard, Alice, Ana S.L. Rodrigues, Victor Cazalis, and David Grémillet. 2021. "Toward a Global Strategy for Seabird Tracking." *Conservation Letters* 14 (3): e12804. <https://doi.org/10.1111/conl.12804>.
- Bhusal, Krishna Prasad, Ankit Bilash Joshi, Deu Bahadur Rana, Deelip Chand Thakuri, and Christopher J. W. McClure. 2023. "Population and Productivity of the Critically Endangered White-Rumped Vulture (*Gyps Bengalensis*) in the Argha Important Bird and Biodiversity Area, Nepal." *Journal of Raptor Research* 57 (3): 352–58. <https://doi.org/10.3356/JRR-22-61>.
- Boeijkens, Johan. 2023. "Ruumiandmete ja pikaajaliste andmeridade kogumismetodid ning kasutusvõimalused merelindude uuringutes." Tartu Ülikool.
- Burger, Joanna, and Michael Gochfeld. 2004. "Marine Birds as Sentinels of Environmental Pollution." *EcoHealth* 1 (3): 263–74. <https://doi.org/10.1007/s10393-004-0096-4>.

- Carbillet, Jeffrey, Lauri Saks, and Tuul Sepp. 2024. “Salivary Corticosterone Reflects Plasmatic Levels in a Wild Seabird.” *General and Comparative Endocrinology* 345:114390. <https://doi.org/10.1016/j.ygcen.2023.114390>.
- Carroll, Matthew J., Ewan D. Wakefield, Emily S. Scragg, Ellie Owen, Simon Pinder, Mark Bolton, James J. Waggitt, and Peter G. H. Evans. 2019. “Matches and Mismatches Between Seabird Distributions Estimated From At-Sea Surveys and Concurrent Individual-Level Tracking.” *Frontiers in Ecology and Evolution* 7. <https://doi.org/10.3389/fevo.2019.00333>.
- Cheon, Yoonsik, and Carlos Chavez. 2021. “Converting Android Native Apps to Flutter Cross-Platform Apps.” In *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, 1898–1904. <https://doi.org/10.1109/CSCI54926.2021.00355>.
- Chougale, Pankaj, Vaibhav Yadav, Anil Gaikwad, and B. Vidyapeeth. 2021. “Firebase-Overview and Usage.” *International Research Journal of Modernization in Engineering Technology and Science* 3 (12): 1178–83. No DOI available.
- Clairbaux, Manon, Mia Rönkä, Tycho Anker-Nilssen, Yuri Artukhin, Jóhannis Danielsen, Maria Gavriilo, Grant Gilchrist, et al. 2024. “An Ecologically Sound and Participatory Monitoring Network for Pan-Arctic Seabirds.” *Conservation Biology* 38 (6): e14287. <https://doi.org/10.1111/cobi.14287>.
- Davis, Kayla L., Emily D. Silverman, Allison L. Sussman, R. Randy Wilson, and Elise F. Zipkin. 2022. “Errors in Aerial Survey Count Data: Identifying Pitfalls and Solutions.” *Ecology and Evolution* 12 (3): e8733. <https://doi.org/10.1002/ece3.8733>.
- Dean, Ben, Holly Kirk, Annette Fayet, Akiko Shoji, Robin Freeman, Kerry Leonard, Christopher M. Perrins, and Tim Guilford. 2015. “Simultaneous Multi-Colony Tracking of a Pelagic Seabird Reveals Cross-Colony Utilization of a Shared Foraging Area.” *Marine Ecology Progress Series* 538:239–48. <https://doi.org/10.3354/meps11443>.
- Edney, T. Hart, M. Jessopp, A. Banks, L. Clarke, L. Cugniere, K. Elliot, et al. 2023. “Best Practices for Using Drones in Seabird Monitoring and Research.” *Marine Ornithology* 51 (2). <https://doi.org/10.5038/2074-1235.51.2.1544>.
- Edney, and Matt J. Wood. 2021. “Applications of Digital Imaging and Analysis in Seabird Monitoring and Research.” *Ibis* 163 (2): 317–37. <https://doi.org/10.1111/ibi.12871>.

- Einoder, L. D. 2009. “A Review of the Use of Seabirds as Indicators in Fisheries and Ecosystem Management.” *Fisheries Research* 95 (1): 6–13.  
<https://doi.org/10.1016/j.fishres.2008.09.024>.
- Fierro-Calderón, Karolina, Mario Loaiza-Muñoz, Manuel A. Sánchez-Martínez, David Ocampo, Santiago David, Harold F. Greeney, and Gustavo A. Londoño. 2021. “Methods for Collecting Data about the Breeding Biology of Neotropical Birds.” *Journal of Field Ornithology* 92 (4): 315–41. <https://doi.org/10.1111/jfo.12383>.
- Fotache, Marin, and Dragos Cogean. 2013. “NoSQL and SQL Databases for Mobile Applications. Case Study: MongoDB versus PostgreSQL.” *Informatica Economica* 17:41–58. <https://doi.org/10.12948/issn14531305/17.2.2013.04>.
- Frederiksen, Morten, Børge Moe, Francis Daunt, Richard A. Phillips, Robert T. Barrett, Maria I. Bogdanova, Thierry Boulinier, et al. 2012. “Multicolony Tracking Reveals the Winter Distribution of a Pelagic Seabird on an Ocean Basin Scale.” *Diversity and Distributions* 18 (6): 530–42. <https://doi.org/10.1111/j.1472-4642.2011.00864.x>.
- Grémillet, David, and Thierry Boulinier. 2009. “Spatial Ecology and Conservation of Seabirds Facing Global Climate Change: A Review.” *Marine Ecology Progress Series* 391:121–38. No DOI available.
- Grémillet, David, Damien Chevallier, and Christophe Guinet. 2022. “Big Data Approaches to the Spatial Ecology and Conservation of Marine Megafauna.” *ICES Journal of Marine Science* 79 (4): 975–86. <https://doi.org/10.1093/icesjms/fsac059>.
- Hodgson, Jarrod C., Shane M. Baylis, Rowan Mott, Ashley Herrod, and Rohan H. Clarke. 2016. “Precision Wildlife Monitoring Using Unmanned Aerial Vehicles.” *Scientific Reports* 6 (1): 22574. <https://doi.org/10.1038/srep22574>.
- Iverson, Autumn R, Jessica L B Schaefer, Shannon M Skalos, and Carly E Hawkins. 2023. “Global Positioning System (GPS) and Platform Transmitter Terminal (PTT) Tags Reveal Fine-Scale Migratory Movements of Small Birds: A Review Highlights Further Opportunities for Hypothesis-Driven Research.” *Ornithological Applications* 125 (3): duad014. <https://doi.org/10.1093/ornithapp/duad014>.

- Junda, James H., Erick Greene, Dan Zazelenchuk, and David M. Bird. 2016. "Nest Defense Behaviour of Four Raptor Species (Osprey, Bald Eagle, Ferruginous Hawk, and Red-Tailed Hawk) to a Novel Aerial Intruder – a Small Rotary-Winged Drone." *Journal of Unmanned Vehicle Systems* 4 (4): 217–27. <https://doi.org/10.1139/juvs-2016-0004>.
- Katti, Vijayalakshmi S., and K Sowmya. 2022. "Flutter-Cross Platform Ide for Mobile Applications." *Grenze International Journal of Engineering & Technology (GI-JET)* 8 (1): 468–72. No DOI available.
- Kesavan, Ram, David Gay, Daniel Thevessen, Jimit Shah, and C. Mohan. 2023. "Firestore: The NoSQL Serverless Database for the Application Developer." In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 3376–88. <https://doi.org/10.1109/ICDE55515.2023.00259>.
- Khan, Wisal, Teerath Kumar, Cheng Zhang, Kislay Raj, Arunabha M. Roy, and Bin Luo. 2023. "SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review." *Big Data and Cognitive Computing* 7 (2): 97. <https://doi.org/10.3390/bdcc7020097>.
- Kreitsberg, Randel, Lisanne Nääb, Richard Meitern, Jeffrey Carbillet, Jérôme Fort, Mathieu Giraudeau, and Tuul Sepp. 2023. "The Effect of Environmental Pollution on Gene Expression of Seabirds: A Review." *Marine Environmental Research* 189:106067. <https://doi.org/10.1016/j.marenvres.2023.106067>.
- Larson, Rachel C., and Dale E. Gawlik. 2023. "Call Rate as an Index of Nest Count in Wading Bird Colonies." *Ibis* 165 (2): 504–16. <https://doi.org/10.1111/ibi.13163>.
- Lewison, R., D. Oro, B. J. Godley, L. Underhill, S. Bearhop, R. P. Wilson, D. Ainley, et al. 2012. "Research Priorities for Seabirds: Improving Conservation and Management in the 21st Century." *Endangered Species Research* 17 (2): 93–121. <https://doi.org/10.3354/esr00419>.
- Lindsay, Ian, and Ningning Nicole Kong. 2020. "Using the ArcGIS Collector Mobile App for Settlement Survey Data Collection in Armenia." *Advances in Archaeological Practice* 8 (4): 322–36. <https://doi.org/10.1017/aap.2020.26>.
- Miller, Julie A. O., Robert W. Furness, Mark Trinder, and Jason Matthiopoulos. 2019. "The Sensitivity of Seabird Populations to Density-Dependence, Environmental Stochasticity and Anthropogenic Mortality." *Journal of Applied Ecology* 56 (9): 2118–30. <https://doi.org/10.1111/1365-2664.13448>.

- Mobasheri, Amin, Helena Mitasova, Markus Neteler, Alexander Singleton, Hugo Ledoux, and Maria Antonia Brovelli. 2020. "Highlighting Recent Trends in Open Source Geospatial Science and Software." *Transactions in GIS* 24 (5): 1141–46. <https://doi.org/10.1111/tgis.12703>.
- Novac, Ovidiu Constantin, Cornelia Mihaela Novac, Bogdan Ciora, Cornelia Emilia Gordan, Mircea Ioan Gordan, and Gyöngyi Bujdosó. 2022. "The Rise of Mobile Development: A Comparison between Ionic and Flutter." In *2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 1–10. <https://doi.org/10.1109/ECAI54874.2022.9847460>.
- Nowak, Maciej M., Katarzyna Dziob, and Pawel Bogawski. 2018. "Unmanned Aerial Vehicles (UAVs) in Environmental Biology: A Review." *European Journal of Ecology* 4 (2): 56–74. <https://doi.org/10.2478/eje-2018-0012>.
- Nowak, Maciej M., Katarzyna Dziób, Łukasz Ludwisiak, and Julian Chmiel. 2020. "Mobile GIS Applications for Environmental Field Surveys: A State of the Art." *Global Ecology and Conservation* 23:e01089. <https://doi.org/10.1016/j.gecco.2020.e01089>.
- Oliveira, Wellington, Bernardo Moraes, Fernando Castor, and João Paulo Fernandes. 2023. "Analyzing the Resource Usage Overhead of Mobile App Development Frameworks." In *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*, 152–61. EASE '23. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3593434.3593487>.
- Patnaik, Rahul, Rajesh Pradhan, Smita Rath, Chandan Mishra, and Lagnajeet Mohanty. 2021. "Study on Google Firebase for Real-Time Web Messaging." In *Intelligent and Cloud Computing*, edited by Debahuti Mishra, Rajkumar Buyya, Prasant Mohapatra, and Srikanta Patnaik, 194:461–69. Smart Innovation, Systems and Technologies. Singapore: Springer Singapore. [https://doi.org/10.1007/978-981-15-5971-6\\_50](https://doi.org/10.1007/978-981-15-5971-6_50).
- Priyadarshani, Nirosha, Isabel Castro, and Stephen Marsland. 2018. "The Impact of Environmental Factors in Birdsong Acquisition Using Automated Recorders." *Ecology and Evolution* 8 (10): 5016–33. <https://doi.org/10.1002/ece3.3889>.

- Quamar, Md Muzakkir, Baqer Al-Ramadan, Khalid Khan, Md Shafiullah, and Sami El Ferik. 2023. “Advancements and Applications of Drone-Integrated Geographic Information System Technology—A Review.” *Remote Sensing* 15 (20): 5039. <https://doi.org/10.3390/rs15205039>.
- Rodríguez, Airam, Beneharo Rodríguez, Yarci Acosta, and Juan J. Negro. 2022. “Tracking Flights to Investigate Seabird Mortality Induced by Artificial Lights.” *Frontiers in Ecology and Evolution* 9. <https://doi.org/10.3389/fevo.2021.786557>.
- Rodríguez, Airam, Beneharo Rodríguez, and Juan J. Negro. 2015. “GPS Tracking for Mapping Seabird Mortality Induced by Light Pollution.” *Scientific Reports* 5 (1): 10670. <https://doi.org/10.1038/srep10670>.
- Sarbavatan, Amirhossein, and Neamat Karimi. 2023. “Development of a Mobile GIS Application (LandInfo) for Land Use and Land Cover Field Data Collection.” Research Square. <https://doi.org/10.21203/rs.3.rs-3752659/v1>.
- Sardà-Palomera, Francesc, Gerard Bota, Núria Padilla, Lluís Brotons, and Francesc Sardà. 2017. “Unmanned Aircraft Systems to Unravel Spatial and Temporal Factors Affecting Dynamics of Colony Formation and Nesting Success in Birds.” *Journal of Avian Biology* 48 (9): 1273–80. <https://doi.org/10.1111/jav.01535>.
- Sardà-Palomera, Francesc, Gerard Bota, Carlos Viñolo, Oriol Pallarés, Víctor Sazatornil, Lluís Brotons, Spartacus Gomáriz, and Francesc Sardà. 2012. “Fine-Scale Bird Monitoring from Light Unmanned Aircraft Systems.” *Ibis* 154 (1): 177–83. <https://doi.org/10.1111/j.1474-919X.2011.01177.x>.
- Sepp, Tuul, Kalev Rattiste, Lauri Saks, Richard Meitern, Janek Urvik, Ants Kaasik, and Peeter Hõrak. 2017. “A Small Badge of Longevity: Opposing Survival Selection on the Size of White and Black Wing Markings.” *Journal of Avian Biology* 48 (4): 570–80. <https://doi.org/10.1111/jav.01136>.
- Soanes, L. M., J. P. Y. Arnould, S. G. Dodd, G. Milligan, and J. A. Green. 2014. “Factors Affecting the Foraging Behaviour of the European Shag: Implications for Seabird Tracking Studies.” *Marine Biology* 161 (6): 1335–48. <https://doi.org/10.1007/s00227-014-2422-x>.

- Sudiartha, I. K. G., I. N. E. Indrayana, I. W. Suasnawa, S. A. Asri, and Putu Wijaya Sunu. 2020. "Data Structure Comparison Between MySQL Relational Database and Firebase Database NoSql on Mobile Based Tourist Tracking Application." *Journal of Physics: Conference Series* 1569 (3): 032092. <https://doi.org/10.1088/1742-6596/1569/3/032092>.
- Sukmana, Yuda, and Yusep Rosmansyah. 2021. "The Use of Cloud Firestore For Handling Real-Time Data Updates: An Empirical Study of Gamified Online Quiz." In *2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT)*, 1239–44. <https://doi.org/10.1109/CECIT53797.2021.00220>.
- Torres, Leigh G., David R. Thompson, Stuart Bearhop, Stephen Votier, Graeme A. Taylor, Paul M. Sagar, and Bruce C. Robertson. 2011. "White-Capped Albatrosses Alter Fine-Scale Foraging Behavior Patterns When Associated with Fishing Vessels." *Marine Ecology Progress Series* 428:289–301. <https://doi.org/10.3354/meps09068>.
- Trevail, Alice M., Malcolm A. C. Nicoll, Robin Freeman, Matthieu Le Corre, Jill Schwarz, Audrey Jaeger, Vincent Bretagnolle, et al. 2023. "Tracking Seabird Migration in the Tropical Indian Ocean Reveals Basin-Scale Conservation Need." *Current Biology* 33 (23): 5247-5256.e4. <https://doi.org/10.1016/j.cub.2023.10.060>.
- Uzzaman, Arfan, Md Majadul Islam Jim, Nourin Nishat, and Janifer Nahar. 2024. "Optimizing SQL Databases for Big Data Workloads: Techniques and Best Practices." *Academic Journal on Business Administration, Innovation & Sustainability* 4 (3): 15–29. <https://doi.org/10.69593/ajbais.v4i3.78>.
- Zink, Richard, Elena Kmetova-Biro, Stefan Agnezy, Ivaylo Klisurov, and Antoni Margalida. 2023. "Assessing the Potential Disturbance Effects on the Use of Unmanned Aircraft Systems (UASs) for European Vultures Research: A Review and Conservation Recommendations." *Bird Conservation International* 33:e45. <https://doi.org/10.1017/S0959270923000102>.
- Žydelis, Ramūnas, Monika Dorsch, Stefan Heinänen, Georg Nehls, and Felix Weiss. 2019. "Comparison of Digital Video Surveys with Visual Aerial Surveys for Bird Monitoring at Sea." *Journal of Ornithology* 160 (2): 567–80. <https://doi.org/10.1007/s10336-018-1622-4>.

## Appendix

### I. Example Microsoft Copilot Prompt for Text Ameliorating

An example of a prompt provided to Microsoft Copilot Enterprise version on 7<sup>th</sup> of November 2024 based on OpenAI GPT-4 architecture, logged in as Richard Meitern Tartu University account.

#### Propt by me:

**Please fix the language and style of the following paragraph for a thesis inclusion, retain all references. Also highlight if something seems missing and what the next paragraph should talk about:** Artificial intelligence (AI) tools were used in the creation of this thesis. For code writing I utilized Android Studio with Github Copilot auto-completion features enabled. The exact versions of the Github Copilot used can be deduced by comparing the git commit history with the release schedule of Github Copilot. For the creation of thesis text passages, I followed a procedure where I first wrote the paragraph with adding references and then prompted Bing Copilot in "Protected" mode with option use GPT-4 and conversation style "Precise" enabled to fix language and asked for advice about next step.

#### Answer by Copilot:

Here's a refined version of your paragraph:

---

Artificial intelligence (AI) tools were instrumental in the creation of this thesis. For code writing, I utilized Android Studio with GitHub Copilot's auto-completion features enabled. The exact versions of GitHub Copilot used can be determined by comparing the git commit history with the release schedule of GitHub Copilot. For the creation of thesis text passages, I followed a procedure where I first wrote the paragraph, including references, and then prompted Bing Copilot in "Protected" mode with the option to use GPT-4 and the conversation style set to "Precise" to fix the language and seek advice on the next steps.

---

#### Suggestions:

1. **Missing Information:** It might be helpful to specify the exact versions of Android Studio and GitHub Copilot if possible. This adds clarity and precision to your methodology.
2. **Next Paragraph:** The next paragraph could discuss the specific benefits and challenges encountered while using these AI tools. This would provide a deeper insight into their practical application and effectiveness in your research process.

Would you like any further adjustments or additional suggestions?

## **II. License**

### **Non-exclusive licence to reproduce the thesis and make the thesis public**

I, Richard Meitern

1. grant the University of Tartu a free permit (non-exclusive licence) to

reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis

#### **Bird Colony: an application for collecting spatial and fitness data from seabird colonies for analysing global patterns**

supervised by Tuul Sepp and Madis Vasser

2. I grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.
3. I am aware of the fact that the author retains the rights specified in points 1 and 2.
4. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

*Richard Meitern*

*08/01/2025*