Mobile Programming and Multimedia - A.y. 2022-2023

HiBirdie

Project report

Trentin Daniele
Master's degree in Computer Science

September 14, 2023

Contents

1 Introduction

The HiBirdie project was developed for the Mobile Programming and Multimedia course. The project consists of a mobile application for birdwatching enthusiasts, available for both iOS and Android. HiBirdie is designed to be a simple social network where every registered user can create their own encyclopedia of birds and share their sightings with other users. By doing so, users accessing the app can know what has been spotted in their vicinity. When a user adds a sighting to their collection, it instantly appears on a dedicated page with all users' sightings. I developed this application using the React Native framework along with the Expo platform. Laravel with a MySQL database was used as the backend. The backend was deployed on an online server with the Nginx web server installed. Is it possible to access the backend via the following address: www.hibirdie.it. The two parts of the project (the frontend developed with React Native and the backend developed with Laravel) are available on GitHub at the following links:

- **Frontend**: https://github.com/dani3ntin/HiBirdieFrontend
- **Backend**: https://github.com/dani3ntin/HiBirdieBackend

2 React Native Framework

React Native is an open-source framework developed by Facebook that allows developers to create native mobile apps for both iOS and Android using JavaScript and React. React is a JavaScript library for building user interfaces, and React Native is a component-based framework with the following characteristics:

- Each part of the application is a component.
- Components are independent and reusable.
- A component is a JavaScript function that takes input (props) and returns a React element constructed with JSX.

Other features of React include the use of:

- **Virtual DOM**: Whenever something changes on the page, React creates a new virtual DOM and compares it to the old one, rendering only the changed elements.
- **States**: Each React component can have its own internal state. State is a JavaScript data structure that holds information specific to that component.
- **Redux**: An open-source library for managing the state of JavaScript applications, designed to address challenges related to state management in complex apps and providing a unidirectional data flow that simplifies data and interaction management in an app.

React Native inherits all the key features of React and contextualizes them for mobile development. Below are the main features of React Native:

- **Cross-Platform**: React Native allows developers to write a single source code in JavaScript and use it to create native apps for both iOS and Android.
- **Code and Component Reusability**: One of React Native's distinctive features is the ability to reuse code. Business logic and common functionalities can be shared across different platforms, while specific components for iOS and Android can be created when needed.
- **Hot Reloading**: This feature allows developers to see the effects of code changes immediately without having to reload the entire app, speeding up the development and debugging process.
- **Large Community**: Currently, React Native has a very active developer community and a vast library of ready-to-use components and add-on modules available through the npm dependency management system.

2.1 Expo

Throughout the project's development, I used Expo, a platform and set of tools for mobile app development with React Native. It is designed to simplify the process of developing iOS and Android apps, allowing developers to focus on writing the app's code and avoiding many of the complexities associated with native development. Expo simplifies the following aspects:

- Starting a new project.
- Accessing device features through appropriate libraries.
- Building and distribution processes.

An additional advantage of Expo is that it can be removed from the project at any time, leaving you with a 100

3 Project Architecture

The project was built by creating both a frontend and a backend. The frontend communicates with the backend using the REST API architecture.

3.1 Frontend

3.1.1 Introduction to the Technologies Used

The frontend was developed using React Native. During development, I paid attention to adhering to the best practices of this framework and tried to reasonably fragment screens into components. I was pleasantly surprised by the following aspects:

- In the application, many components are often repeated, so I was able to save time by reusing some of them.
- React Native's hot reloading allowed me to test the application in real-time.
- There is a lot of support and many libraries available online.
- Almost all the code I wrote works well for both iOS and Android. In the initial stages of development, I only tested on Android devices since I did not have access to iOS devices. When I tested for the first time on an iOS device, I was surprised to see how little code I had to modify to adapt it to iOS.

Here are some drawbacks I encountered during development:

- Fragmenting screens into components increases code reuse, but if not done judiciously, the code can become less readable.
- React Native has a steep learning curve and may be challenging for newcomers.

I believe that React Native is a good framework that offers many more pros than cons.

3.2 Backend

3.2.1 Introduction to the Technologies Used

The backend was developed using Laravel, an open-source MVC framework written in PHP for web application development. Laravel is a comprehensive framework that offers many features, but in the project, I used it solely for handling API calls from the frontend. I used MySQL as the database, a widely used and easy-to-use relational database management system (DBMS).

3.2.2 Backend Deployment

I deployed the Laravel application to a virtual machine in the cloud on Oracle Cloud Infrastructure. Currently, Oracle provides a completely free virtual machine in the cloud with 24GB of RAM, 4 CPUs, and 200GB of storage. Below is a list of all the steps I took to deploy the Laravel project:

1. I rented a virtual machine with text-based Ubuntu.
2. I installed all the necessary applications (MySQL, Nginx, Git, Composer, etc.).
3. I uploaded the Laravel project to the virtual machine using Git.
4. I obtained the domain hibirdie.it for free from the register.it website.
5. I set up the inbound rules necessary for the virtual machine to allow external access via IP address and port.
6. I configured the Nginx web server to route external requests to my Laravel project.
7. I used CertBot (a client that retrieves and deploys Let's Encrypt SSL/TLS certificates) to obtain a valid SSL certificate and configured the Laravel application to use the HTTPS protocol.

I must admit that the deployment was much more complex than I had anticipated. It took much more time than I thought and required more knowledge than I had initially. However, the deployment process was extremely instructive (sometimes frustrating) and allowed me to learn a lot about a topic I had never explored before.

4 HiBirdie Application

In this section, I will describe each page in HiBirdie and provide rationale for each design choice I've made. The application runs on both iOS and Android, and there are some differences in the UI that I will highlight. If the pages are the same on both platforms, I will only show the Android version.

4.1 General considerations for app design and implementation choices

Below, I list some design considerations that apply to the entire application and not to a specific page.

- All app buttons are at least 7mm (44 pixels) in size. Buttons placed at the bottom of the screen have been increased in size for better usability (70 pixels).
- I generally aimed to position all buttons and clickable components at the bottom of the screen to make the application easy to use.
- The 'back' button for Android devices works on all app pages.
- The swipe gesture to go back for iOS devices is functional on all application pages.
- The application adapts well to different font sizes, thereby enhancing accessibility.
- I centered the content of all pages on the screen.
- When a logged in user closes the application, he/she will automatically be logged in the next time he/she will reopen the app
- The application makes extensive use of images and 'preview' images (small round images used in lists). For this reason, I chose to manage two different versions of the same image: an original resolution version (the image uploaded by the user) and a compressed version (used for preview images). When the user uploads a new image in the application, whether it's a bird image or a profile picture, the backend saves both the original version and a compressed version. The data savings achieved through compression are significant; for example, a 1.64-megabyte jpg image in the original version weighs only 80 kilobytes in the compressed version. This procedure allows for significantly reduced internet consumption when there are lists of birds or user profiles, and it also greatly increases the loading speed of pages with lists.
- I developed the application in such a way that it adapted well to all screen sizes and font sizes. In the figure 1, you can see the application on a smartphone with the maximum text font size.

4.2 Header Bar

Before describing all the pages, I'd like to explain the choices made for the header bar. The header bar is 70 pixels tall and has a different color than the background to distinguish it. The content of the header bar changes depending on the page, but a back button is always positioned in the top left corner (except for the homepage). This button's placement is inherited from Apple's software design, where the back button is always in the top left corner. To cater to Apple users who are accustomed to this configuration, I chose to implement it in my project. However, recognizing that the button is particularly difficult to reach, Android users can use the system-provided back button, while Apple users can use the swipe gesture to go back, as used on iOS devices. The button in the header bar is 70 pixels large, so it meets the minimum size requirement.

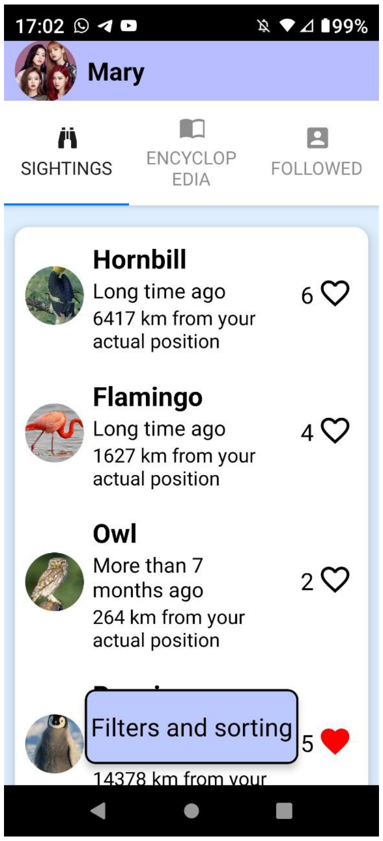


Figure 1: Application on a smartphone with the maximum text font size

4.3 Introduction Page

Every time the application is opened, users are taken to the introductory page (see Figure 2). If the user has previously logged in, he/she will be taken directly to the homepage without requiring authentication. If the user is not authenticated, he/she can choose to sign up for a new account or log in with an existing account.

I positioned an image in the center of the screen for the following reasons:

- To make the application more inviting,
- To ensure that the two access buttons are easily reachable for the user by placing them below without leaving empty space between the title and the buttons themselves.

4.4 Login and Sign up

When the user clicks on the Login or Sign up button, he/she is respectively taken to the Login and Sign up pages (see Figure 3).

For login, users can authenticate with the username or email used during registration. If the credentials are incorrect, a red message appears to communicate the problem to the user. For signup, the required data includes:

- Username (must be unique),
- Name,

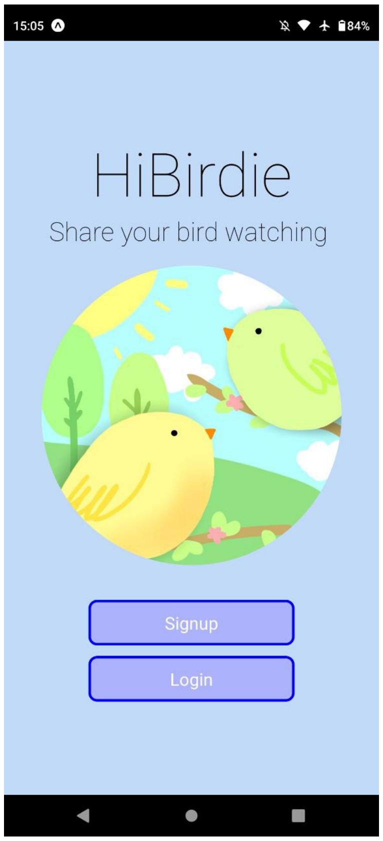


Figure 2: Introduction Page

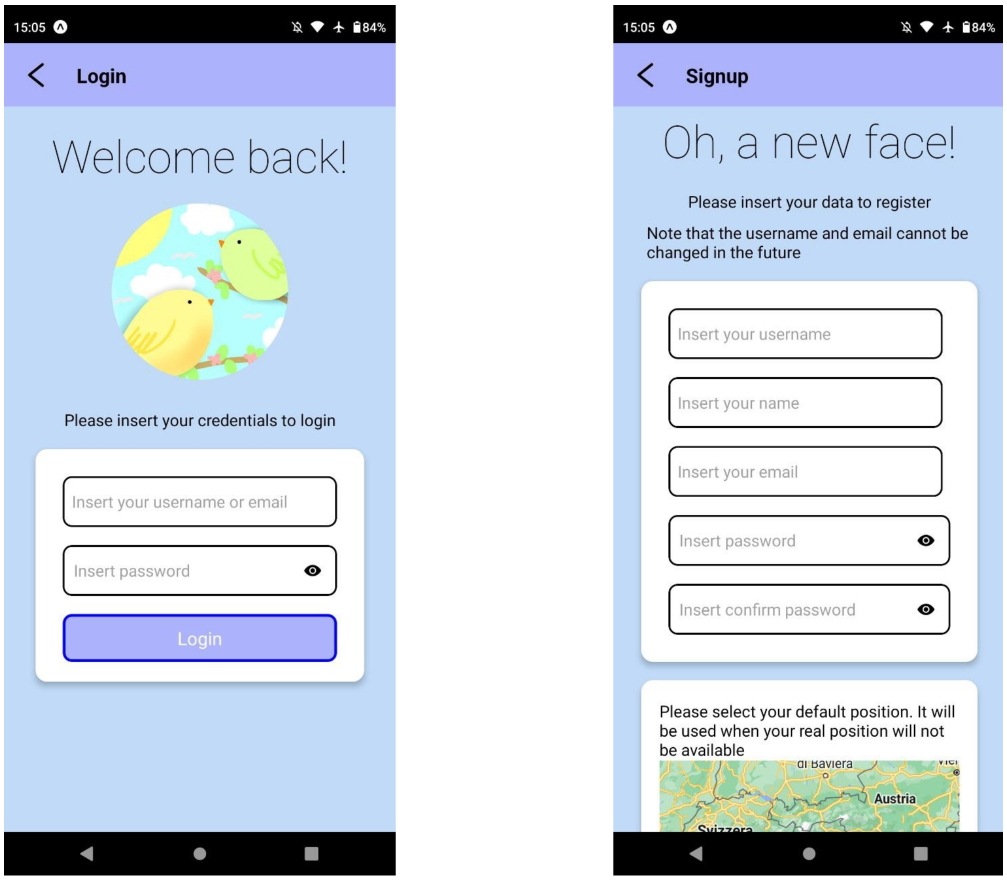
- Email (must be unique),
- Password,
- Confirm password,
- Default location.

The application checks that the username and email entered are not already in use by other users. The default location is mandatory and is used by the app to calculate the distance from sighted birds when real-time location is unavailable. At the bottom of the page, there is a sign up button. In case of missing data, the application displays a warning indicating the missing information. Each user profile also includes their profile image and a status. However, I chose to remove these inputs from the sign up process and keep only the essential ones for the app to function, to avoid overwhelming the user with too much data entry and to allow them to access the application as quickly as possible. Missing information can still be added later in the user settings page.

4.5 Homepage

The homepage (see Figure 4) is divided into three sub-pages:

- **Sightings**: The page displaying the latest sightings,
- **Encyclopedia**: The page for your own encyclopedia,


15:05

Login

Welcome back!

Please insert your credentials to login

Insert your username or email

Insert password

Login

15:05

Signup

Oh, a new face!

Please insert your data to register

Note that the username and email cannot be changed in the future

Insert your username

Insert your name

Insert your email

Insert password

Insert confirm password

Please select your default position. It will be used when your real position will not be available

Figure 3: Login and Sign up Pages

- **Followed**: The page for people you follow.

The homepage is structured differently depending on the device. On iOS, I placed the page navigation menu at the bottom for easy usability. On Android, I positioned the navigation menu at the top to avoid having a duplicate navigation menu at the bottom. Given the challenging usability of the top menu position on Android, users can use the ’swipe’ gesture to switch between sub-pages.

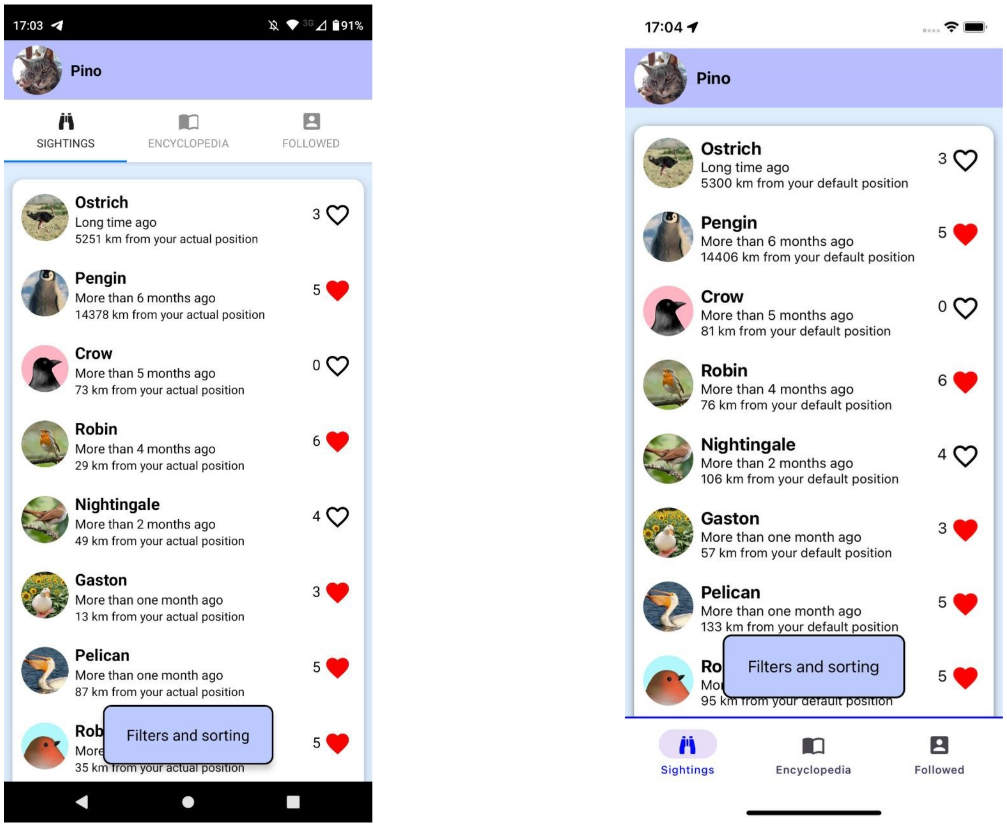
Below, I describe all three sub-pages.

4.5.1 Sightings Page

The sightings page (always visible in Figure 4) is the page displayed after signup or login. The page shows sightings made by other users of the app. When a user adds a sighting to their own encyclopedia, it appears in the ’Sightings Page’ for all other users. Users can also filter sightings based on their preferences and sort them according to certain criteria. For each bird, the page displays when it was sighted and the distance in kilometers from the user’s current location or default location (if real-time location is unavailable). I placed the button to open the filters at the bottom as a floating button for the following reasons:

- To avoid covering the main content of the page (the sightings),
- To make it easily accessible and clickable.

Clicking on any item in the list opens the detailed page of the selected bird. Clicking the ’Filters and Sorting’ button opens a modal where filters and sorting criteria can be selected. To close the modal, I placed


17:03 91%

Pino

SIGHTINGS ENCYCLOPEDIA FOLLOWED

Ostrich
Long time ago
5251 km from your actual position
3

Pengin
More than 6 months ago
14378 km from your actual position
5

Crow
More than 5 months ago
73 km from your actual position
0

Robin
More than 4 months ago
29 km from your actual position
6

Nightingale
More than 2 months ago
49 km from your actual position
4

Gaston
More than one month ago
13 km from your actual position
3

Pelican
More than one month ago
87 km from your actual position
5

Rob
More
35 km from your actual position
5

Filters and sorting

17:04

Pino

Ostrich
Long time ago
5300 km from your default position
3

Pengin
More than 6 months ago
14406 km from your default position
5

Crow
More than 5 months ago
81 km from your default position
0

Robin
More than 4 months ago
76 km from your default position
6

Nightingale
More than 2 months ago
106 km from your default position
4

Gaston
More than one month ago
57 km from your default position
3

Pelican
More than one month ago
133 km from your default position
5

Ro
Mo
95 km from your default position
5

Filters and sorting

Sightings Encyclopedia Followed

Figure 4: Homepage in Android and iOS Versions

the 'Close' button at the bottom, or users can click on the semi-transparent top part of the screen to close it (the latter method for closing the modal is less convenient, but I chose to implement it to avoid betraying user expectations). Finally, on this page, users can like sightings by clicking on the heart. The clickable area includes not only the heart itself but also the number next to it and a border around the heart. I made this choice to make the button easier to click and require less precision from the user. In the image (Figure 5), I highlighted the clickable area in yellow.

4.5.2 Encyclopedia Page

The Encyclopedia page (Figure 6) contains all the sightings made by the logged-in user. For each bird, the date of the sighting is displayed.

The structure is very similar to the Sightings page, with the button to add a new bird to the encyclopedia replacing the filter button. Clicking on an item in the list opens the detailed page of the selected bird. Clicking the 'Add New Bird' button takes users to the page for adding a new bird to their personal encyclopedia.

4.5.3 Followed Page

The Followed page (Figure 7) lists the users followed by the logged-in user. Each user's status is displayed. If the status is too long to fit in the designated space, the application truncates it with three dots.

The structure is very similar to the Sightings page, with the button to search for a user replacing the filter button. Clicking on a user in the list takes users to the profile page of the selected user. Clicking the 'Search Users' button opens a modal where users can search for an app user by username. As users begin typing, the application filters all app users, displaying only those that contain the searched string in the

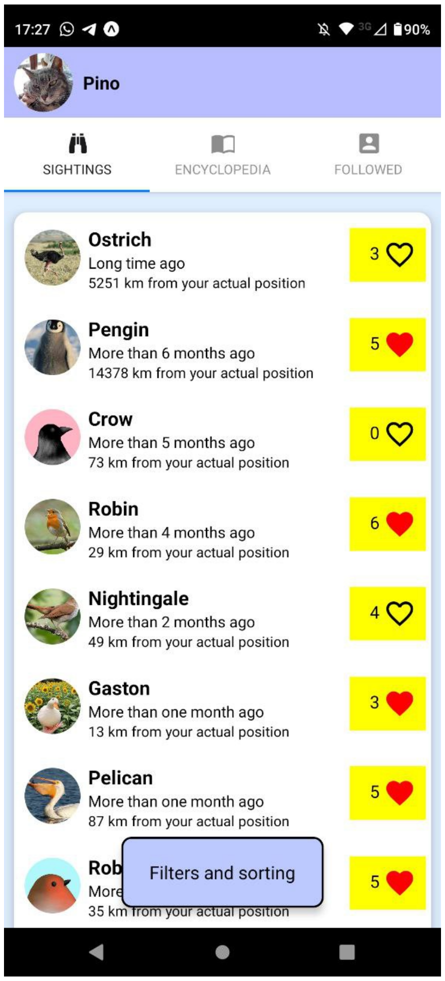


Figure 5: Clickable Area for Liking

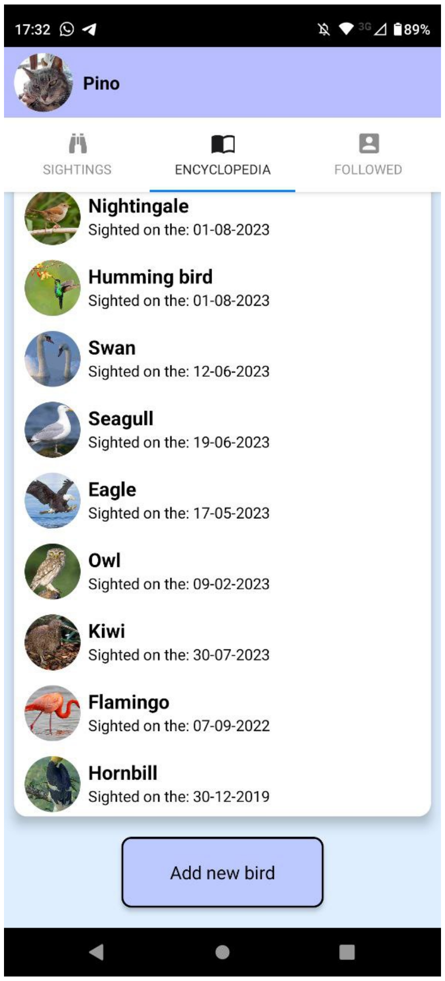
username. Clicking on a user takes users to the profile page of the selected user. To close the modal, I placed the 'Close' button at the bottom, or users can click on the semi-transparent top part of the screen.

4.6 Detail bird page

The Detail bird page (Figure 8) is the page that opens when the user clicks on a bird entry in the application. There are two versions of this page:

- **Detail bird page with author**: This version of the page appears when a user clicks on a bird in recent sightings or another user's profile. The only difference from the other version is that it displays who made the sighting.
- **Detail bird page without author**: This version of the page appears when a user clicks on a bird in the encyclopedia. One difference from the other version is that it does not display who made the sighting since the user, being on their encyclopedia page, knows that they are the author of the sighting. Additionally, this version includes buttons to remove the bird from the encyclopedia and to edit it.

The page, from top to bottom, contains:


17:32 89%

Pino

SIGHTINGS ENCYCLOPEDIA FOLLOWED

Nightingale
Sighted on the: 01-08-2023

Humming bird
Sighted on the: 01-08-2023

Swan
Sighted on the: 12-06-2023

Seagull
Sighted on the: 19-06-2023

Eagle
Sighted on the: 17-05-2023

Owl
Sighted on the: 09-02-2023

Kiwi
Sighted on the: 30-07-2023

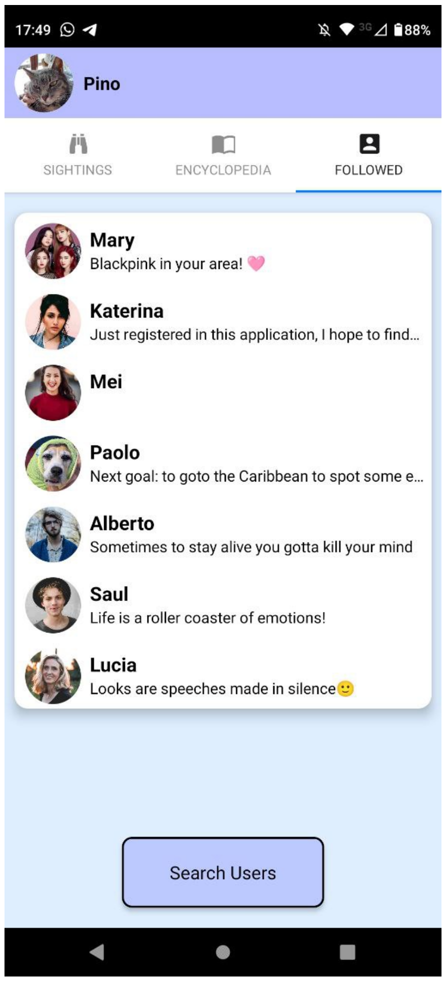
Flamingo
Sighted on the: 07-09-2022

Hornbill
Sighted on the: 30-12-2019

Add new bird

Figure 6: Encyclopedia Page

- **Bird image**: This is the image uploaded by the sighting's author. Clicking on the image allows users to view it in full screen and zoom in.
- **Sighting Author**: This information is only present in the version with the author. It includes the author's photo, their name, and the number of followers. Clicking on the author takes users to their Detail user page.
- **Sighting Date**
- **Personal notes**
- **Sighting Location**: To enable map navigation, users need to tap it first. This choice prevents accidental interaction with the map while scrolling through the page.
- **Edit and Delete buttons**: These are only present in the version without the author. They allow the encyclopedia's owner to edit the bird's properties or delete it.
- **Comments Section**: To add a comment, users need to write in the provided text box and press the 'Add a comment' button. Clicking on another user's image or name takes the users to their 'User detail page'.


17:49

Pino

SIGHTINGS ENCYCLOPEDIA FOLLOWED

Mary
Blackpink in your area! 💗

Katerina
Just registered in this application, I hope to find...

Mei

Paolo
Next goal: to goto the Caribbean to spot some e...

Alberto
Sometimes to stay alive you gotta kill your mind

Saul
Life is a roller coaster of emotions!

Lucia
Looks are speeches made in silence🙂

Search Users

Figure 7: Followed Page

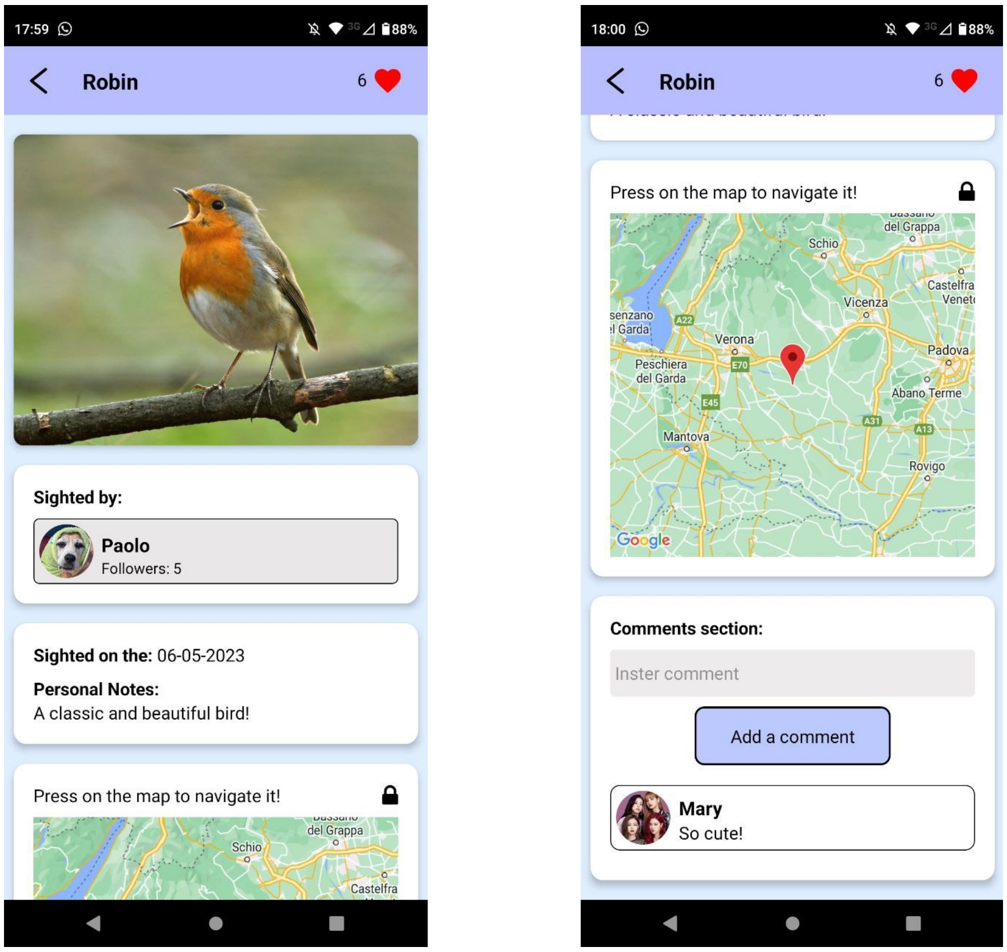
4.7 Add new bird page

The Add new bird page (figure 9) allows a new bird to be added to the logged-in user's encyclopedia. Automatically, the sighting will appear on the 'Sightings page' for all other users.

The page requires the following information to be entered for each bird:

- **Bird photo**: optional,
- **Bird name**: mandatory,
- **Personal notes**: optional,
- **Sighting date**: mandatory but already with a default value (today's date),
- **Bird location**: mandatory but already with a default value (the current location of the user. If it is not available, his default position).

I chose to place the component to insert the photo as first because it takes up height space making the button to add a photo easy to reach since it ends up in the center to the screen. This way the component is both useful, in that it shows the picture of the bird, and makes the page more usable. If I had put, for


17:59 88%

Robin 6

Sighted by:

Paolo
Followers: 5

Sighted on the: 06-05-2023

Personal Notes:
A classic and beautiful bird!

Press on the map to navigate it!

18:00 88%

Robin 6

Press on the map to navigate it!

Comments section:

Inster comment

Add a comment

Mary
So cute!

Figure 8: Detail bird page (with author version)

example, the name of the bird as the first component, it would not be easily reachable because it would be at the top of the screen. If the user for various reasons does not enter a photo of the bird (perhaps because he or she does not currently have the image available or perhaps because he or she wants to alert others to the presence of a bird in a location but does not have a photo available) the app inserts a default photo. For some birds, there are custom images available. If the app finds the name of a bird type for which there is a custom image, it will use it instead of the default one.

4.8 Edit a bird page

The Edit page is the same as the Add new bird page, the only difference is that all fields are filled in with the data of the bird you want to edit.

4.9 Detail user page

The Detail user page (figure 11) is the page that is opened when the user presses on a user's entry in the application.

The page shows all available information about a user:

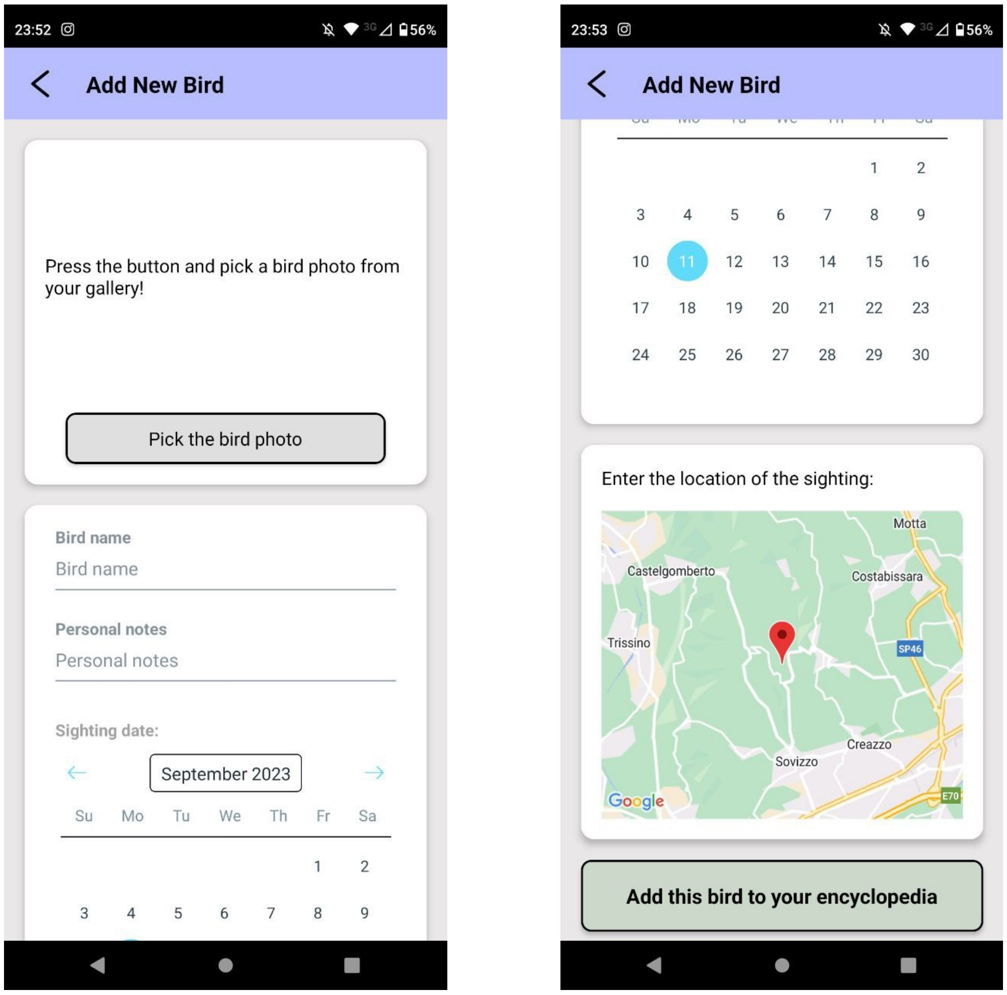


Figure 9: Add new bird page

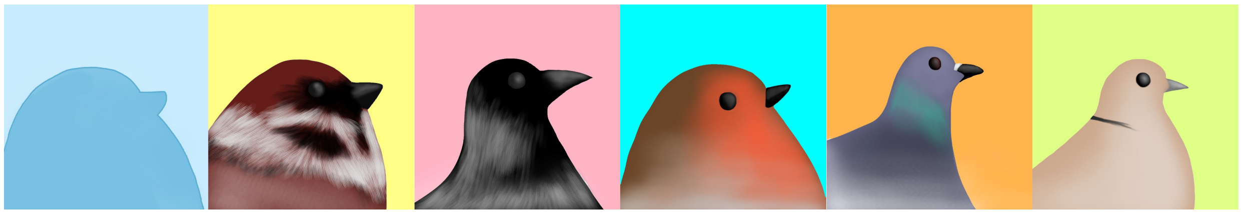
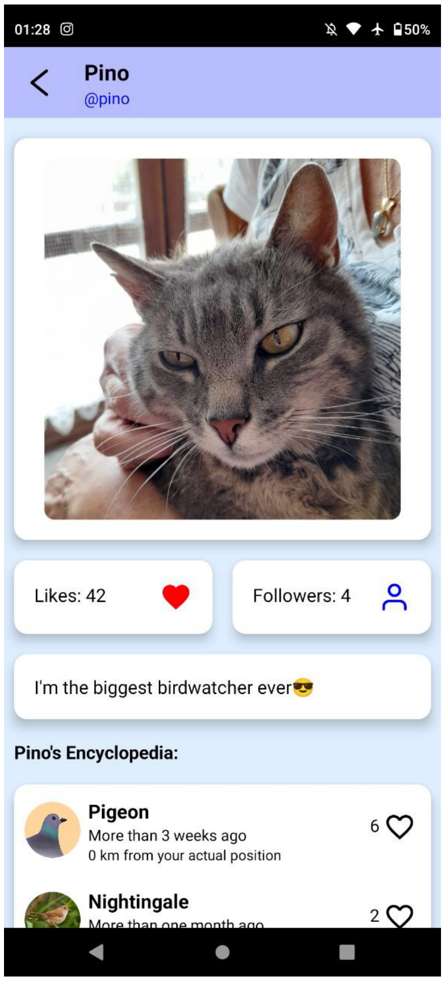


Figure 10: Default bird images. From left: default bird, sparrow, crow, robin, pigeon, turtle dove).

- **Profile Picture**: if pressed, it allows the profile picture to be displayed full screen and you can zoom in,
- **Likes**: if pressed, it shows the distribution of likes from other users,
- **Followers**: if pressed, shows the user's followers,
- **State**,



I'm the biggest birdwatcher ever😎

Pino's Encyclopedia:

Figure 11: Detail user page

- **User encyclopedia**: the user's complete encyclopedia is shown. All encyclopedia entries are pressable.
- **Follow button**: pressing it adds the user among the set of followed users. If the user was already followed then he/she is removed from the set of followed users.

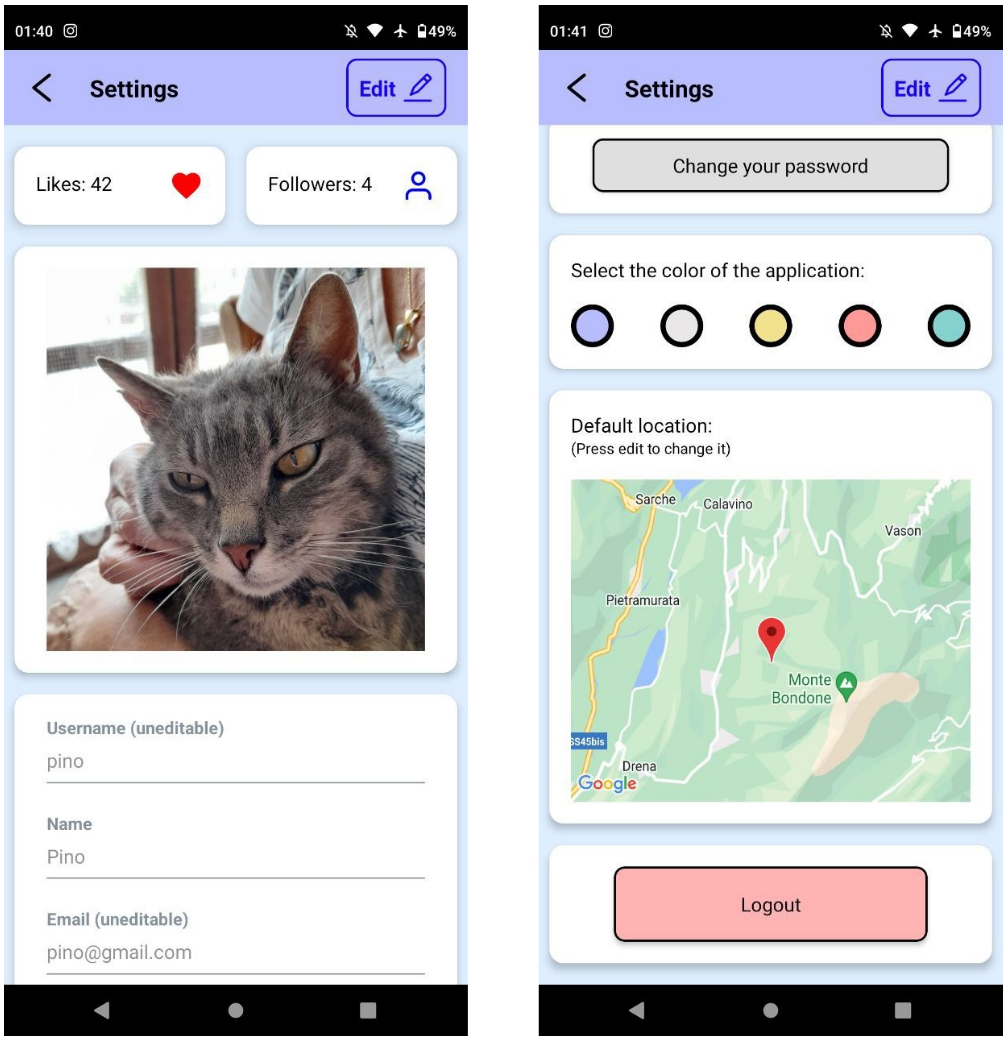
Again, I placed the image at the top of the screen to move the pressable parts to more comfortable areas while still showing useful information.

4.10 User settings page

On the User settings page (figure 12) you can edit information about your user and the application.

In order from top to bottom are the following components:

- **Likes**: if pressed, shows the distribution of likes by other users on one's account.
- **Followers**: if pressed, shows the followers on one's account.
- **Profile Photo**: if pressed, allows you to view the profile picture full screen and you can zoom in.
- **Personal info form**: this form contains the username, name, email and status of one's account.



Likes: 42

Followers: 4

Username (uneditable)
pino

Name
Pino

Email (uneditable)
pino@gmail.com

Select the color of the application:

Default location:
(Press edit to change it)

Figure 12: User settings page

- **Change your password**: button that if pressed allows you to change your account password.
- **Color of the application selection**: by pressing on the buttons you can change the basic color of the entire application.
- **Default location**: through this component you can change the default location of your account. This location is used to calculate the distance to spotted birds if real-time location is not available.
- **Logout button**: if pressed, it asks the user to confirm logout. If the user gives confirmation, logout is performed and the user is taken to the introductory page of the application. Confirmation is asked to prevent unintentional logout by the user.

In the upper right corner of the header bar is present a button that says 'Edit'. If pressed, all locked components become editable. The user after making as many changes as he/she wishes, presses the same button again (now marked 'Save changes') to confirm all changes.

5 Possible Improvements

Listed below are some improvements that can be made to the application that due to lack of time I have not implemented:

- Implement in the Encyclopedia page and Sighting page a partial loading of birds, e.g. load thirty elements at a time. To display thirty more elements, there will be a button at the bottom to press that says 'Load more elements'.
- Implement personal chat between users.
- Instead of being able to upload a single photo, add the ability to upload a gallery of photos and videos.
- Being able to group sightings within a specific radius. For example, the user can choose to view which sightings have been reported in the Trento area. By doing so, the user can determine if the selected city could be a potential destination.

6 Access and Login to the application

There is already some usable demonstration data in the application. The three users with the most test data entered are:

- Pino:
  - **Username**: pino
  - **Password**: password
- Mary:
  - **Username**: mary
  - **Password**: password
- Paolo:
  - **Username**: paolo
  - **Password**: password

In the current data, all users in the app have as their password: 'password'. So you can access every user in the app, just get their username, the password is always the same.