

EcoDex

University of Miami

CSC431-S

Advisor: Vanessa Aguiar

Team 03



Michael Castellucci
Scrum Master

Carter Falkenberg
Developer

Phuong Uyen Dang
Developer



INTRODUCING Ecolex

PROJECT

APPLICATION DESIGN
& DEVELOPMENT FOR
THE CATEGORIZATION
OF FLORA AND FAUNA



what you can do with EcoDex

Functional
Requirements



Sort the species by uploading a photo you have taken of the specimen.



Learn about different plants and animals and how to differentiate between them.



Connect with others and share your love for nature with like-minded individuals.



Flora/Fauna Recognition

Simple steps to identify a species
of flora or fauna.

Step 1



Take
picture

Step 2



Upload
to app

Step 3



Identify
species

Step 4



Earn
badges



Connect with nature
lovers from around
the world!

Make an account and chat with like-
minded individuals to share the love for
nature!



View Logs Map

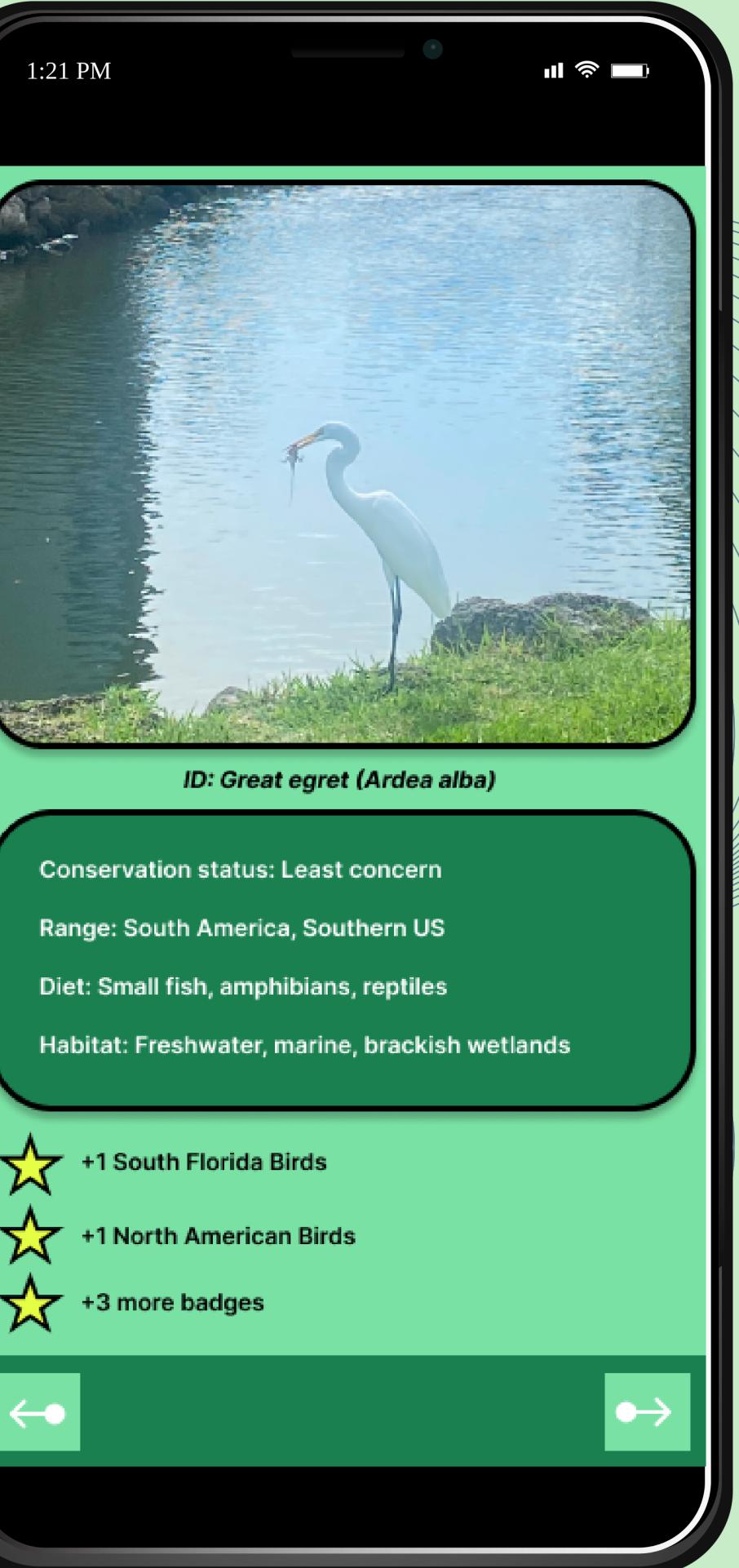
Users could choose to view a map of other users' observations, flora and fauna native to some regions, and relevant information.





UI Example

- Uploaded picture of organism
- Information about it
- Badges earned



List of Functional Requirements

1

| | |
|-------------------|--|
| Title | Log flora/fauna |
| Description | The user should be able to take a picture of a plant/animal/fungi and upload it to the app for identification |
| Priority | 0 |
| Precondition(s) | App interfaces to allow access and ML classification |
| Basic Flow | User hits the button to create a log, and fills in location, a picture, optional comments, and a recommendation is given based on the picture. |
| Postconditions(s) | Log is added to the observations map and shown on friends feed. Log is included in database |
| Use Case Diagram | 3.1.1 |

2

| | |
|-------------------|--|
| Title | Identify flora/fauna |
| Description | Image classification of plant/animal/fungi |
| Priority | 0 |
| Precondition(s) | User must take photo of plant/animal/fungi |
| Basic Flow | After the user takes a photo, the model runs with an input of that image and outputs a classification. |
| Postconditions(s) | Display to the user the type of plant/animal/fungi |
| Use Case Diagram | 3.1.1 |

3

| | |
|-------------------|---|
| Title | Save user entry |
| Description | After classification, log the photo, classification, and location in the user's database. |
| Priority | 0 |
| Precondition(s) | Model must classify the plant/animal/fungi that the user took a photo of. Location must be enabled. |
| Basic Flow | After the user takes a photo that is classified, the metadata is automatically saved to the user's account. |
| Postconditions(s) | |
| Use Case Diagram | 3.1.1 |

4

| | |
|-------------------|---|
| Title | Make account |
| Description | User should be able to create an EcoDex account to keep track of their progress |
| Priority | 0 |
| Precondition(s) | Login page needs to be created |
| Basic Flow | First page encountered is a login page with an option to create an account. Account creation page takes email, name, and username and creates an account with it. |
| Postconditions(s) | Allows access to friend management and adding logs |
| Use Case Diagram | 3.1.1 |

5

| | |
|-------------------|---|
| Title | View observations map |
| Description | User should be able to view other users' observations around the world |
| Priority | 2 |
| Precondition(s) | Need a map system to display logs at GPS coordinates |
| Basic Flow | A button on the app takes you to the map where you can spin, zoom in/out, and click on other logs |
| Postconditions(s) | Logs can be viewed in detail |
| Use Case Diagram | 3.1.1 |

6

| | |
|-------------------|---|
| Title | Add/remove friends |
| Description | Users will be able to connect with each other by username |
| Priority | 2 |
| Precondition(s) | Account creation/login needs to be implemented |
| Basic Flow | User search page |
| Postconditions(s) | Users can see each others logs as they are created |
| Use Case Diagram | 3.1.1 |

List of Functional Requirements

7

| | |
|-------------------|---|
| Title | Login to account |
| Description | User should be able to access their created account |
| Priority | 1 |
| Precondition(s) | Making account must be implemented |
| Basic Flow | When opening app, user is prompted with a login screen where they can enter their account information and login |
| Postconditions(s) | Access to all other features of EcoDex |
| Use Case Diagram | 3.1.1 |

8

| | |
|-------------------|--|
| Title | Friend Feed |
| Description | Page where the user can see their friends' logs |
| Priority | 2 |
| Precondition(s) | User must have account with friends |
| Basic Flow | User logs in and is directed to EcoDex home page. From there an option is to view friend feed. |
| Postconditions(s) | User is able to interact with posts or go back to home page |
| Use Case Diagram | 3.1.1 |

9

| | |
|-------------------|---|
| Title | Chat |
| Description | Tab where user can chat with another user |
| Priority | 2 |
| Precondition(s) | User must be friends with the other user |
| Basic Flow | In friend feed, there is a chat icon next to each friend where the user can view all conversations with other users |
| Postconditions(s) | User is able to send messages to friends |
| Use Case Diagram | 3.1.1 |

10

| | |
|-------------------|--|
| Title | Report User |
| Description | An option in each user's profile to report |
| Priority | 2 |
| Precondition(s) | User must have account and is able to view other users' profiles |
| Basic Flow | User can view another user's profile and report them for review |
| Postconditions(s) | User can specify the reason(s) for reporting |
| Use Case Diagram | 3.1.1 |

11

| | |
|-------------------|--|
| Title | Get Badges |
| Description | Badges are given based on specifics of a log. i.e. if a rare animal is spotted |
| Priority | 3 |
| Precondition(s) | User must be able to add logs to profile |
| Basic Flow | User enters app and uploads a new log |
| Postconditions(s) | User can view what badges they've just unlocked and return to home screen |
| Use Case Diagram | 3.1.1 |

12

| | |
|-------------------|---|
| Title | Settings |
| Description | User can adjust account settings |
| Priority | 3 |
| Precondition(s) | User must be logged in |
| Basic Flow | User is able to view profile, and click on the Gear icon to view and adjust a variety of settings |
| Postconditions(s) | User can view what badges they've just unlocked and return to home screen |
| Use Case Diagram | 3.1.1 |

Non-Functional Requirements



Timely Classification

Specimen should be identified in less than 1 second.



Cross-Platform Compatibility

Portable on iOS and Android.



Scalability Feature

Should be scalable to a large number of users (up to 100,000 users)



User's Data Protection

Encryption of username, password, email, password, friends, location data, and photos.



Community Guidelines

Protect users from negative community experiences by making sure everybody adhere to the Community Guidelines.



1

| | |
|------------------|--|
| Title | Classify flora/fauna in less than 1 second |
| Description | When the user takes a photo, the plant/animal/fungi should be classified within 1 second |
| Priority | 3 |
| Applicable FR(s) | 1.1.2 |

2

| | |
|------------------|---|
| Title | Encrypt user information |
| Description | The user's data should all be secured and encrypted. This includes name, email, password, friends, location data, and photos. |
| Priority | 1 |
| Applicable FR(s) | 1.1.1, 1.1.3 |

3

| | |
|------------------|---|
| Title | Portable on IOS and Android |
| Description | The app should work for both IOS and Android users equally. |
| Priority | 0 |
| Applicable FR(s) | All |

4

| | |
|------------------|--|
| Title | Scalable up to 100,000 users |
| Description | Until nearly 100,000 users are active, the app should be completely functional for up to 100,000 users. Speed, reliability, and robustness should not be affected. |
| Priority | 2 |
| Applicable FR(s) | All |

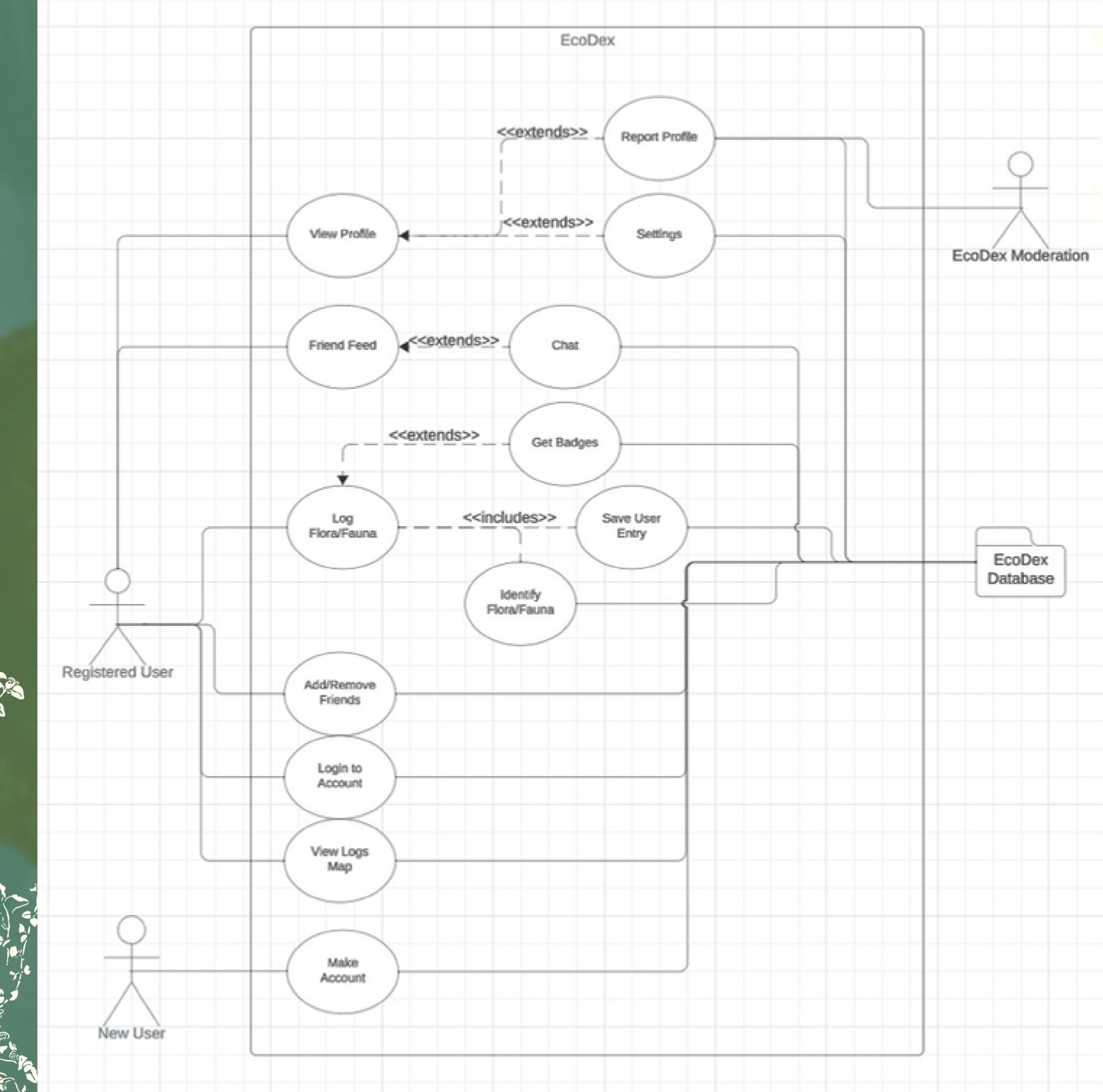
5

| | |
|------------------|--|
| Title | Community Guidelines |
| Description | Ensure user interactions stay civil and within terms defined in a page about Community Guidelines to protect users from possible negative community experiences. |
| Priority | 1 |
| Applicable FR(s) | All |

List of Non-Functional Requirements

Use Case Diagram

Requirements Modeling



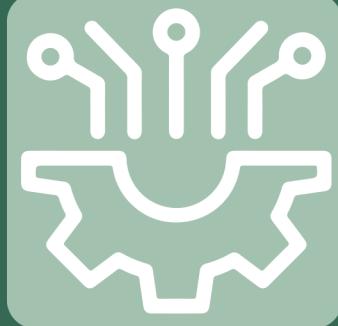
System Constraints

Several technical problems that should be addressed going forward with development.

By proactively addressing these technical issues, we can work more efficiently and effectively towards achieving the project goals.

Model Training

Ensuring accurate identification may require substantial training of the machine learning model.



Device Compatibility

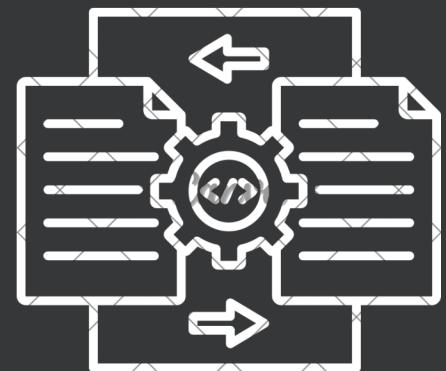
Smooth operation on a range of devices with diverse specifications and capabilities.



Collaboration Tools

Facilitating consistent and effective communication among team members.





Source Control

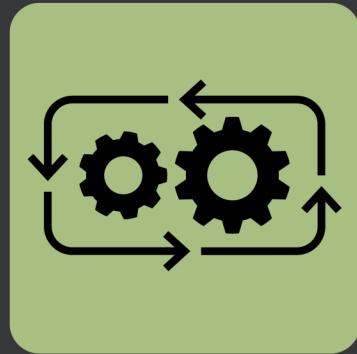


GitHub

Github will be utilized for source control due to its many benefits, most importantly collaboration and version control.

Collaboration

Various tools for branching, merging, and resolving conflicts, enabling simultaneous collaboration.



Version Control

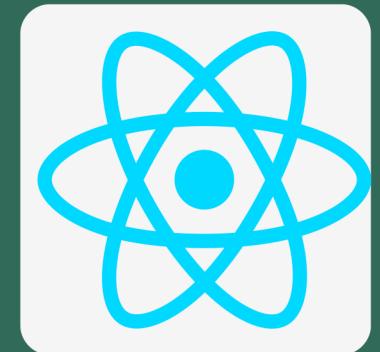
Robust version control, allowing for developers to track changes and revert to previous versions.



Mobile App Framework

React Native

React Native will be utilized for cross-platform compatibility between iOS and Android devices.



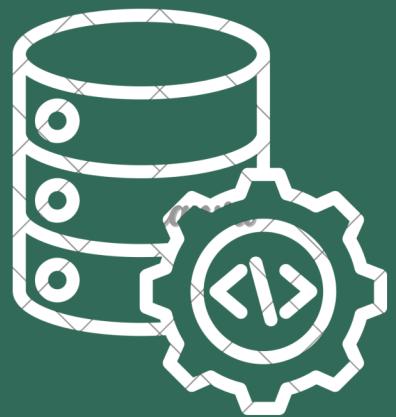
Android

Android-specific functionalities could be accessed when needed; adaptable to various configurations.



iOS

Adjusting to iOS design principles and simplifying development while maintaining a native aesthetic.



Backend Framework



🍀 Django 🍀

Django allows for a pythonic backend framework to optimize speed, security, and scalability.



🍀 PyInaturalist 🍀

A python client for Inaturalist, will be used as the backbone for the app's main functionality.



🍀 ML Model 🍀

A large CNN-centric architecture will be trained on public flora/fauna datasets.



Server



🍀 Microsoft Azure 🍀

The application's server will be primarily hosted on Microsoft Azure for its variety of beneficial services.



🍀 Scalable Infrastructure 🍀

Allows for flexible adjustment of app server capacity, improving cost efficiency.



🍀 Seamless Integration 🍀

Native compatibility means easy integration with a range of other Microsoft services and technologies.

List of System Constraints

| | |
|-------------|--|
| Title | Source Control |
| Description | GitHub will be used for source control. Source: https://github.com/ |
| Priority | 0 |

| | |
|-------------|--|
| Title | Mobile App Framework |
| Description | React native will be used so that the app can be deployed to both Android and IOS Source: https://reactnative.dev/ |
| Priority | 0 |

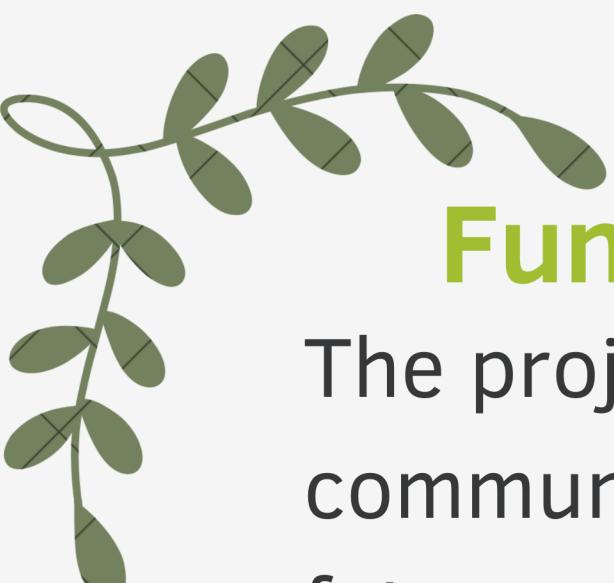
| | |
|-------------|--|
| Title | Model Framework |
| Description | We will use django and pyinaturalist, written in python, in order to make calls to the classification model. Source: https://github.com/pyinat/pyinaturalist Source: https://www.djangoproject.com/ |
| Priority | 0 |

| | |
|-------------|--|
| Title | Server |
| Description | We will host a server using Microsoft Azure https://azure.microsoft.com/en-us |
| Priority | 0 |



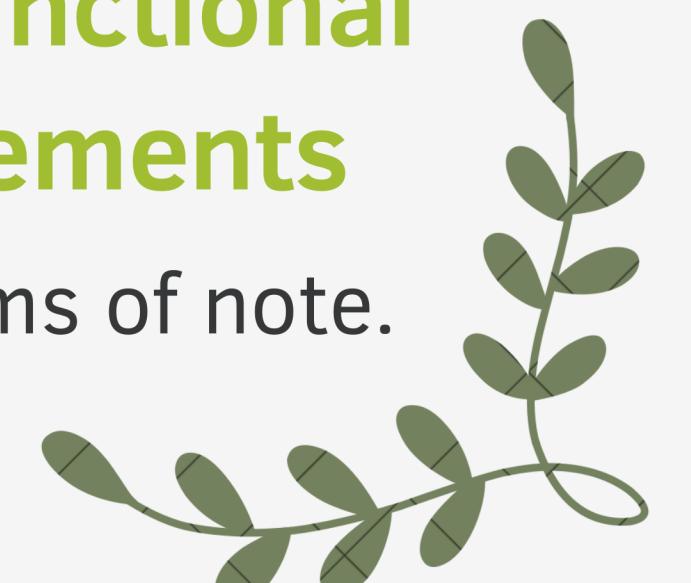
Evolutionary Requirements

Continuous refinement of existing functions as well as additional development of new features.



Functional Requirements

The project aims to enhance community options in the foreseeable future.



Non-Functional Requirements

Other items of note.



*Group
Options*



*Post
Options*



*Other
Options*

Functional Requirements

Features to be implemented in subsequent versions of the project





Group Options



Post Options

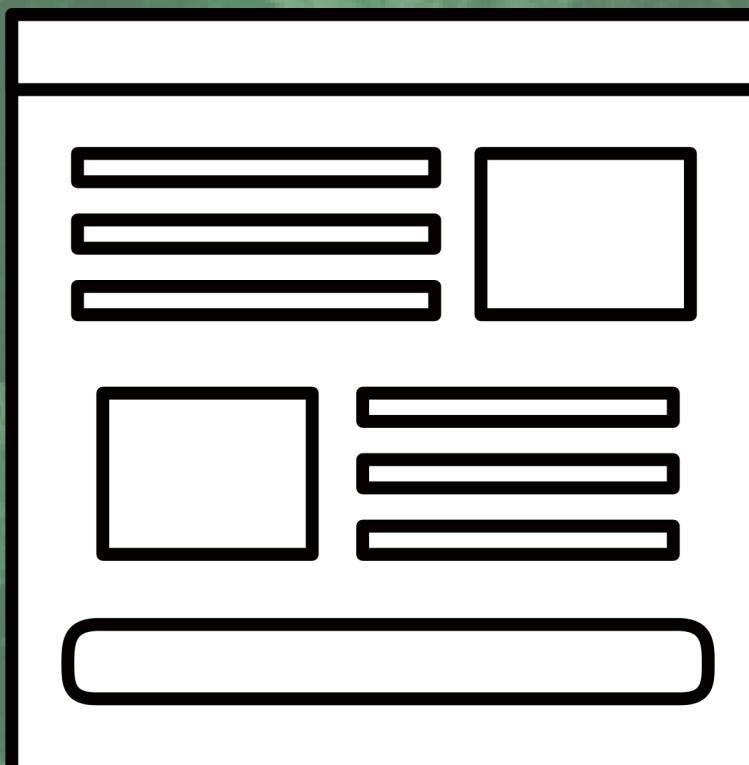


Other Options

As a registered EcoDex user, you can:

- ❖ Create Group
- ❖ Manage Group

Form a community with
nature lovers like yourself!





*Group
Options*



**Post
Options**



*Other
Options*

As an EcoDex group member, you can:

- Create Post
- Comment on Post
- React to Post





Group Options



Post Options



Other Options

View Account History
Displayed as a timeline of log entries

Block/Unblock User

Flora/Fauna of the Day



List of Evolutionary Functional Requirements

1

| | |
|-------------------|--|
| Title | Create Group |
| Description | Users can join together in groups where they can share their finds and earn badges together. |
| Priority | 5 |
| Precondition(s) | The user must have an account and friends. |
| Postconditions(s) | Allows access to group functions and activities. |
| Use Case Diagram | N/A |

2

| | |
|-------------------|---|
| Title | Manage Group |
| Description | A user designated as the group leader is able to invite, kick, and mute members in their group. |
| Priority | 5 |
| Precondition(s) | A group must be created with members. |
| Postconditions(s) | Group homepage is shown along with group posts (if any). |
| Use Case Diagram | N/A |

3

| | |
|-------------------|--|
| Title | Make Post |
| Description | Users have the option to make a post separate from a log, where they are allowed more freedom in what they post and talk about |
| Priority | 5 |
| Precondition(s) | User needs an account. |
| Postconditions(s) | Post will be displayed to friends of that user and the user will be brought back to home page. |
| Use Case Diagram | N/A |

4

| | |
|-------------------|---|
| Title | Comment on Post |
| Description | Users can write text comments on their friends' posts. |
| Priority | 5 |
| Precondition(s) | User must have friends and one of them must have posted. |
| Postconditions(s) | Friends can see their comments and then, the user is brought back to post view. |
| Use Case Diagram | N/A |

5

| | |
|-------------------|--|
| Title | React to Post |
| Description | Works in the same way as 4.1.4, but with a single emoji. |
| Priority | 5 |
| Precondition(s) | Same as 4.1.4 |
| Postconditions(s) | Same as 4.1.4 |
| Use Case Diagram | N/A |

6

| | |
|-------------------|---|
| Title | Flora/Fauna of the Day |
| Description | The system automatically finds a fascinating flora/fauna to display it to all users. This action is repeated daily. |
| Priority | 5 |
| Precondition(s) | User navigates to home page to view flora/fauna of the day |
| Postconditions(s) | An organism of the day is displayed to the user along with some facts about it. |
| Use Case Diagram | N/A |

7

| | |
|-------------------|--|
| Title | View Account History |
| Description | Account history could be viewed by the user as a timeline of log entries, unlocked species, badges, and other significant account events. |
| Priority | 4 |
| Precondition(s) | The user navigates to their profile and chooses the option to View Account History. |
| Postconditions(s) | A timeline of the account history is visible to the user. The user can interact with the events on the timeline by clicking on any event to view more information including location and time (if applicable). |
| Use Case Diagram | N/A |

8

| | |
|-------------------|--|
| Title | Block/Unblock User |
| Description | The user is able to view and block/unblock other users from their profiles. |
| Priority | 4 |
| Precondition(s) | The user must be able to view another user's profile. |
| Postconditions(s) | Another user is thus blocked and cannot interact with the user, including viewing the user's feed or chatting with the user, and vice versa for the user. If the blocked account is friends with the user, this action automatically unfriends them. The user can still view a blocked account's profile and choose to unblock it. |
| Use Case Diagram | N/A |

Non-Functional Requirements

On enhancing future versions
of the application.

Record Account Events



The system automatically logs account events with location and time details for a history summary accessible from the user's profile.



Report Suspicious Activity



The system identifies activities that might appear suspicious and takes cautionary steps to protect user information.



List of Evolutionary Non-Functional Requirements

1

| | |
|------------------|---|
| Title | Record Account Event |
| Description | System automatically records the location and time (if applicable) of certain account events for a summary of the account history which could then be viewed from the user's profile. |
| Priority | 4 |
| Applicable FR(s) | 4.1.7 |

2

| | |
|------------------|--|
| Title | Report Suspicious Account Activity |
| Description | System automatically recognizes activities (such as important account information changes) that could not be reasonably attributed to the rightful account owner and sends a warning email to the email address currently listed in account information. |
| Priority | 4 |
| Applicable FR(s) | All |

THE END
THANK
YOU
ECODEX
TEAM 03

PRESENTATION TEMPLATE BY CANVA