

# EcoDex

## Final Presentation

University of Miami

CSC 431 - S

Advisor: Vanessa Aguiar

# Team members

 Team 03

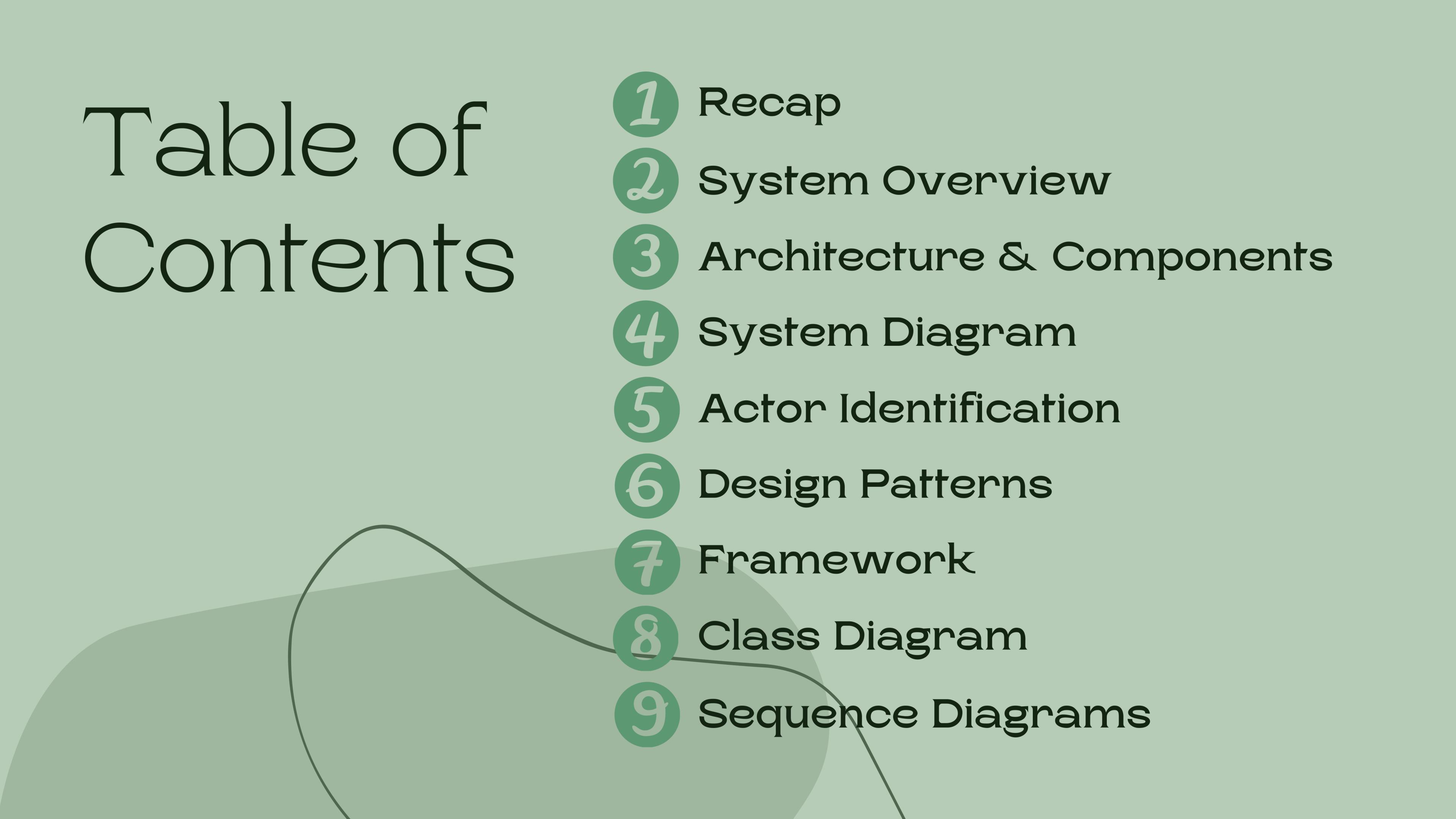


Michael Castellucci  
Scrum Master

Carter Falkenberg  
Developer

Uyen Dang  
Developer

# Table of Contents

- 
- 1 Recap**
  - 2 System Overview**
  - 3 Architecture & Components**
  - 4 System Diagram**
  - 5 Actor Identification**
  - 6 Design Patterns**
  - 7 Framework**
  - 8 Class Diagram**
  - 9 Sequence Diagrams**



# Recap

- Give people an incentive to explore nature
- Connect with other nature lovers
- Classify/Learn about plants, animals, and fungi
- Elements of social media + mobile game
- Target audience is all ages, anyone who is curious about nature

# System Overview

EcoDex's goal is to educate and increase awareness about various species of plants and animals to safeguard ecosystems.

- Identifying a species of flora or fauna by either a photo upload of it or via the in-app camera feature.
- Connecting users to other users who might share the same interests.
- Displaying a logs map of regional flora and fauna.
- Informing users about interesting and relevant facts about a flora or fauna (such as its uses in medical history, etc.)



# Architecture

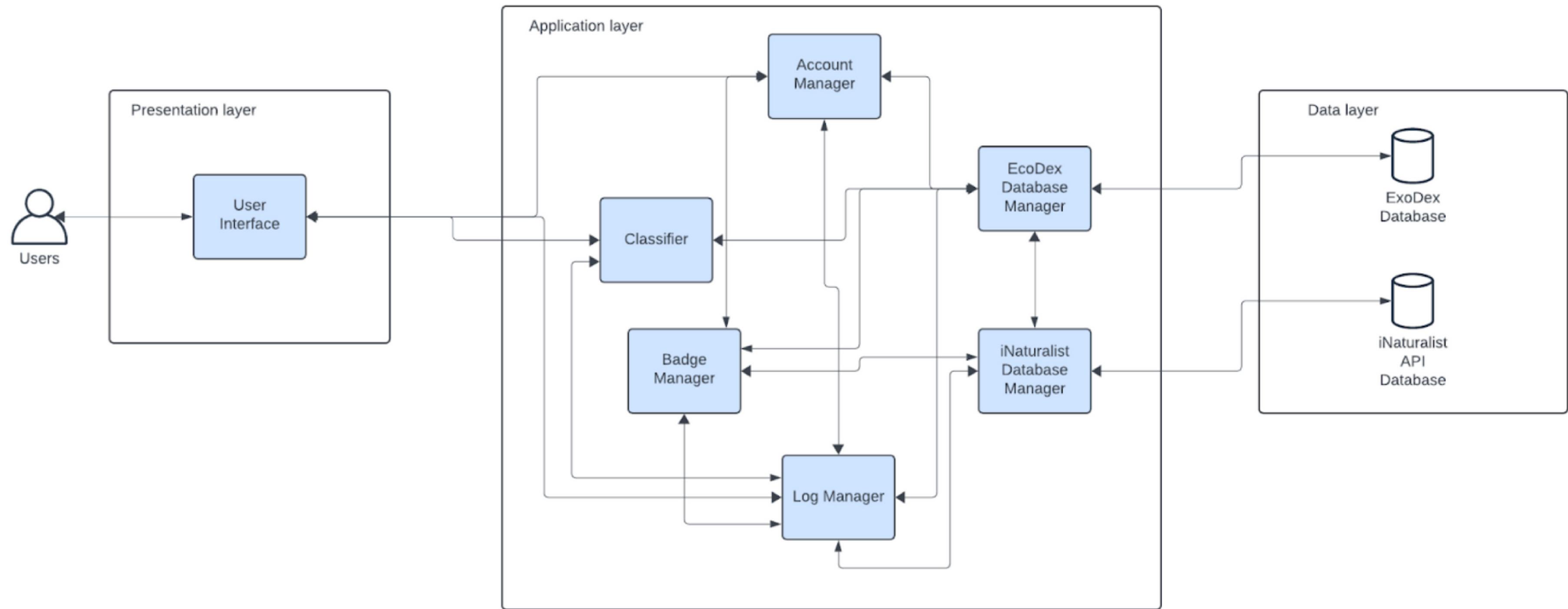
The application uses a **three-tier architecture**:

- **Presentation:** Blends a social media feed with a Pokémon-style collection of unlocked species, alongside a global map showcasing identified species worldwide.
- **Application:** Using machine learning to classify and retrieve photos for the feed, and linking the map to the correct location and species data.
- **Data tier:** All necessary data for the app, including user information, stored photos, machine learning model weights, friends lists, and other relevant information.

# Components

#	COMPONENT	MAIN FUNCTIONALITY	SECONDARY FUNCTIONALITY	INTERACTIONS
1	User Interface	Display information to the user and accept inputs	Pass user data to other components	2,3,4,5
2	Classifier	Relay classification info to EcoDex database and back to Log Manager		3,6
3	Log Manager	Manage functions related to creating Logs		1,2,4,5,6,7
4	Badge Manager	Manage functions related to badge earning		1,3,6
5	Account Manager	Manage functions related to user accounts	Help verify logins	1,3,4,6
6 & 7	Database Manager(s)	Facilitate access to EcoDex and iNaturalist databases		2,3,4,5

# System Diagram



# Actor Identification

- **NEW USER:** HAS NO PRIOR ECODEX ACCOUNT.
- **REGISTERED USER:** HAS AN ECODEX ACCOUNT.
- **MODERATION TEAM:** HANDLES REPORTED CONTENT.
- **PLATFORM SERVER:** STORES USER DATA AND APP CONTENT.

# Design Patterns

## Factory Method

In some cases, objects and components may need to be created based on user actions or system requirements. This design pattern offers a solution by delegating the object creation process to subclasses, enabling flexible management of different types of objects.

## Façade Method

This design pattern simplifies and unifies the interface to complex underlying subsystems within an application. This pattern encapsulates interactions with message styles, feed presentation, data manipulation, and storage components, presenting a unified interface to higher-level application logic.

# Framework

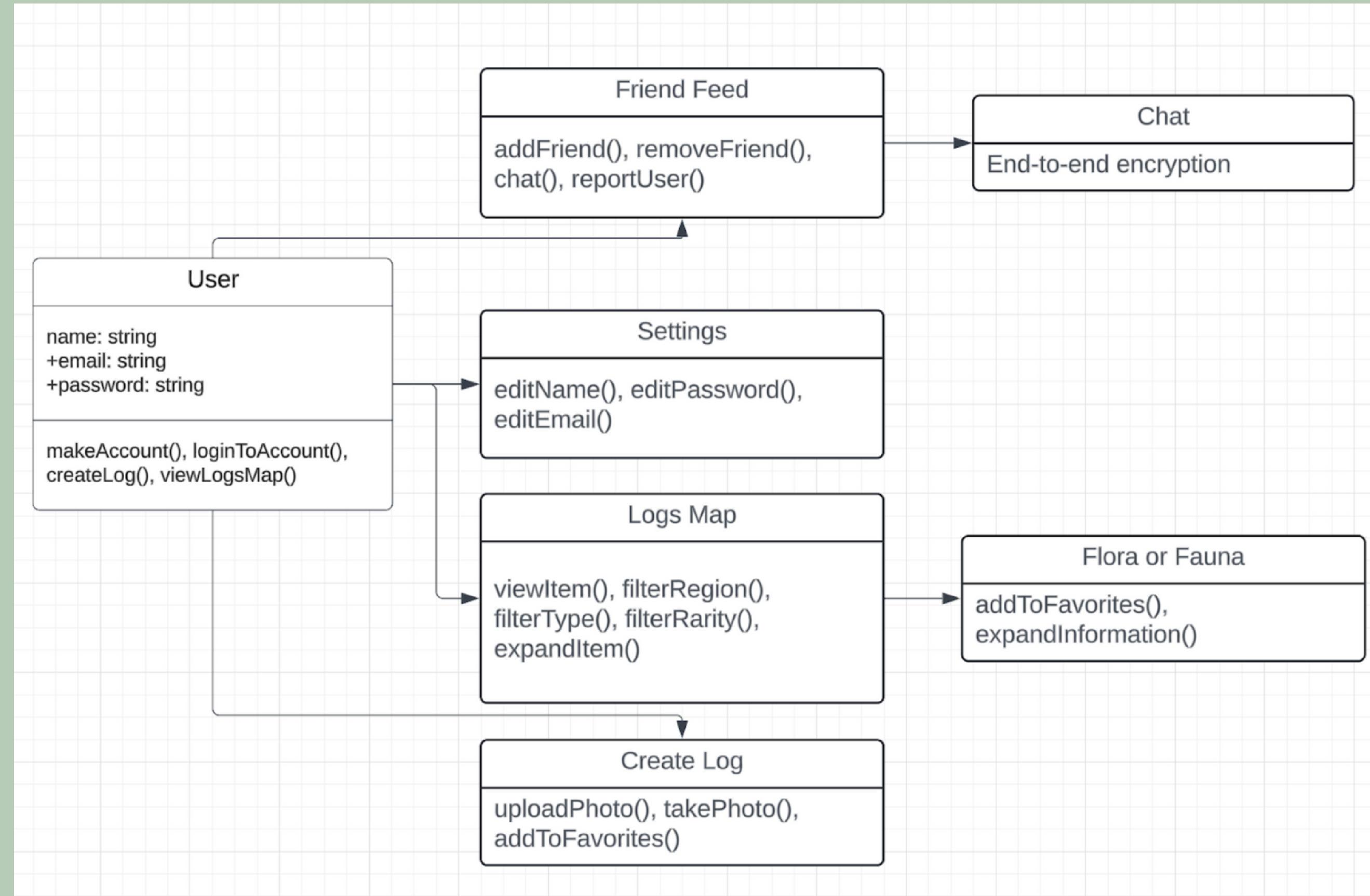
**Machine Learning Algorithm:** PyTorch is open-source, allowing for highly customizable machine learning model architectures. This will be suitable for optimizing a CNN-based species recognition model. Its extensive libraries and tools facilitate efficient model optimization.

**Front End:** React is selected for its versatility and robustness, offering seamless integration with Android and iOS platforms. The decision is influenced by the abundance of templates and comprehensive online documentation, making React an ideal choice despite limited front-end experience.

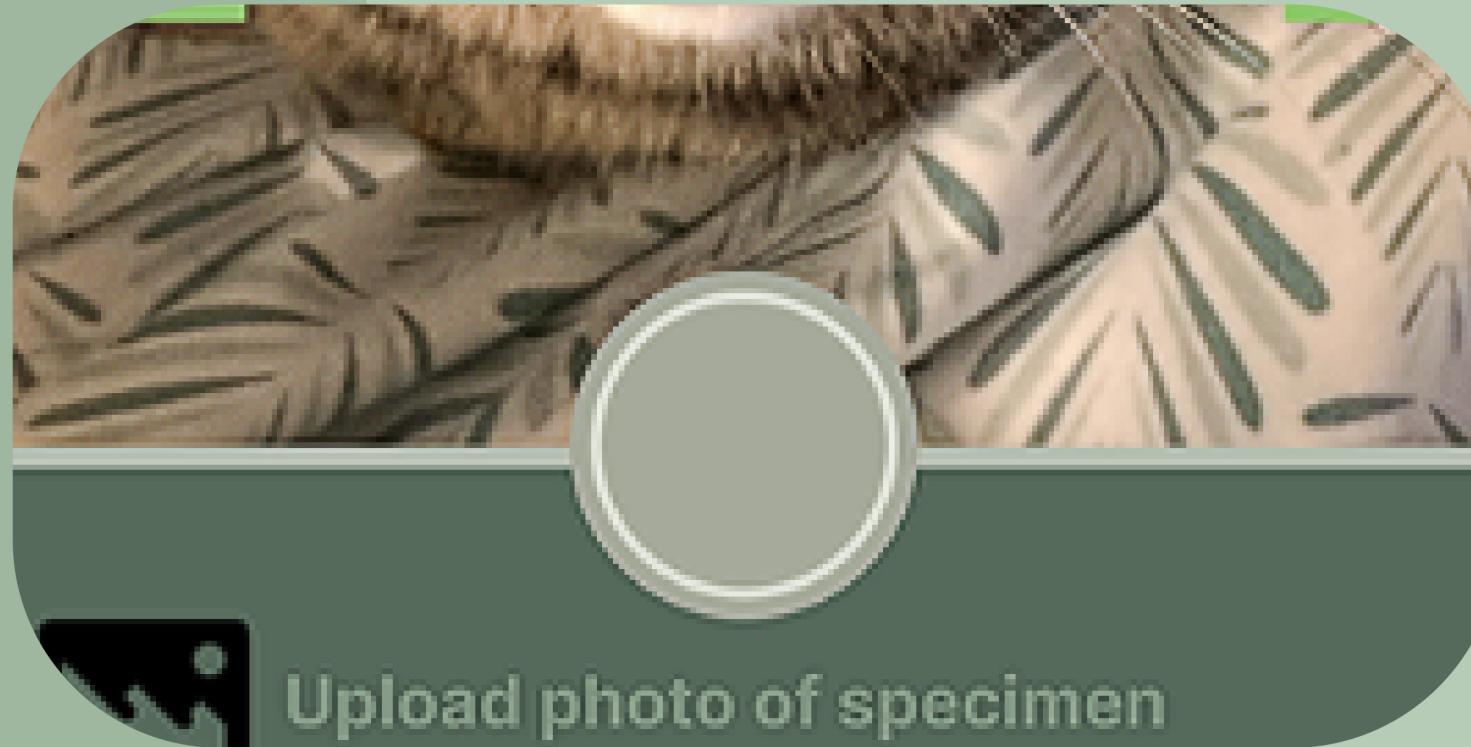
**Back End:** Django is preferred for the back end due to its straightforward Python integration with the machine learning model and scalability for the application. It aligns well with the Python-based PyTorch framework used for the machine learning algorithm.

**Species References and Information:** IPyNaturalist is used to access species references and information, leveraging from INaturalist's popular app. IPyNaturalist, written in Python, integrates well with PyTorch and the developers' skill sets, making it a suitable choice over directly using the INaturalist API.

# Class Diagram



# Sequence Diagrams



## Create Log Sequence Diagram

Creating Logs is one of the main functionalities of EcoDex, allowing users to create posts and earn badges from them.

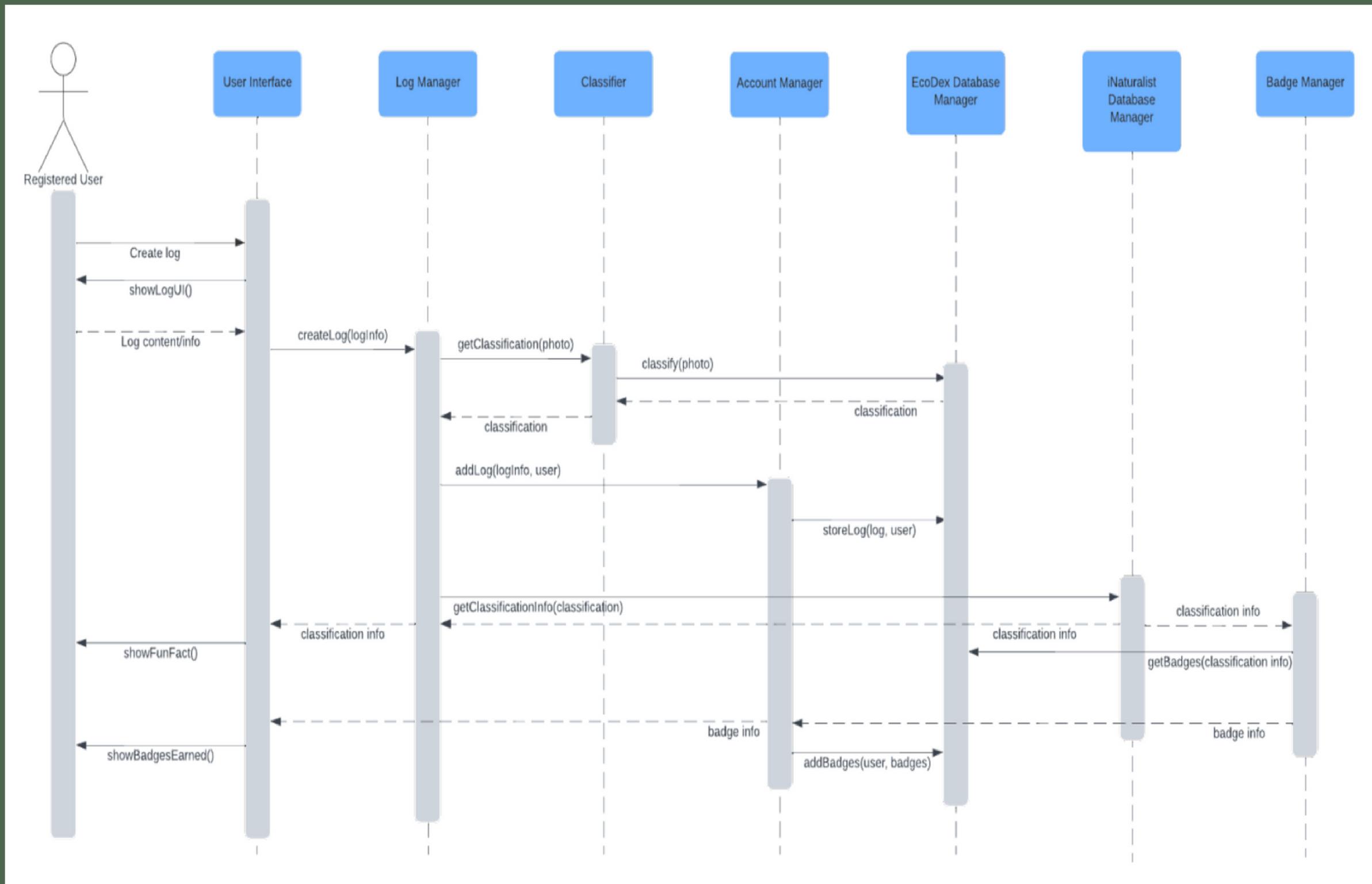
A simplified login form with fields for "Email" and "Password". Below the password field is a small eye icon. At the bottom is a large, rounded rectangular button with a gradient from blue to pink, labeled "LOGIN".

## Login Sequence Diagram

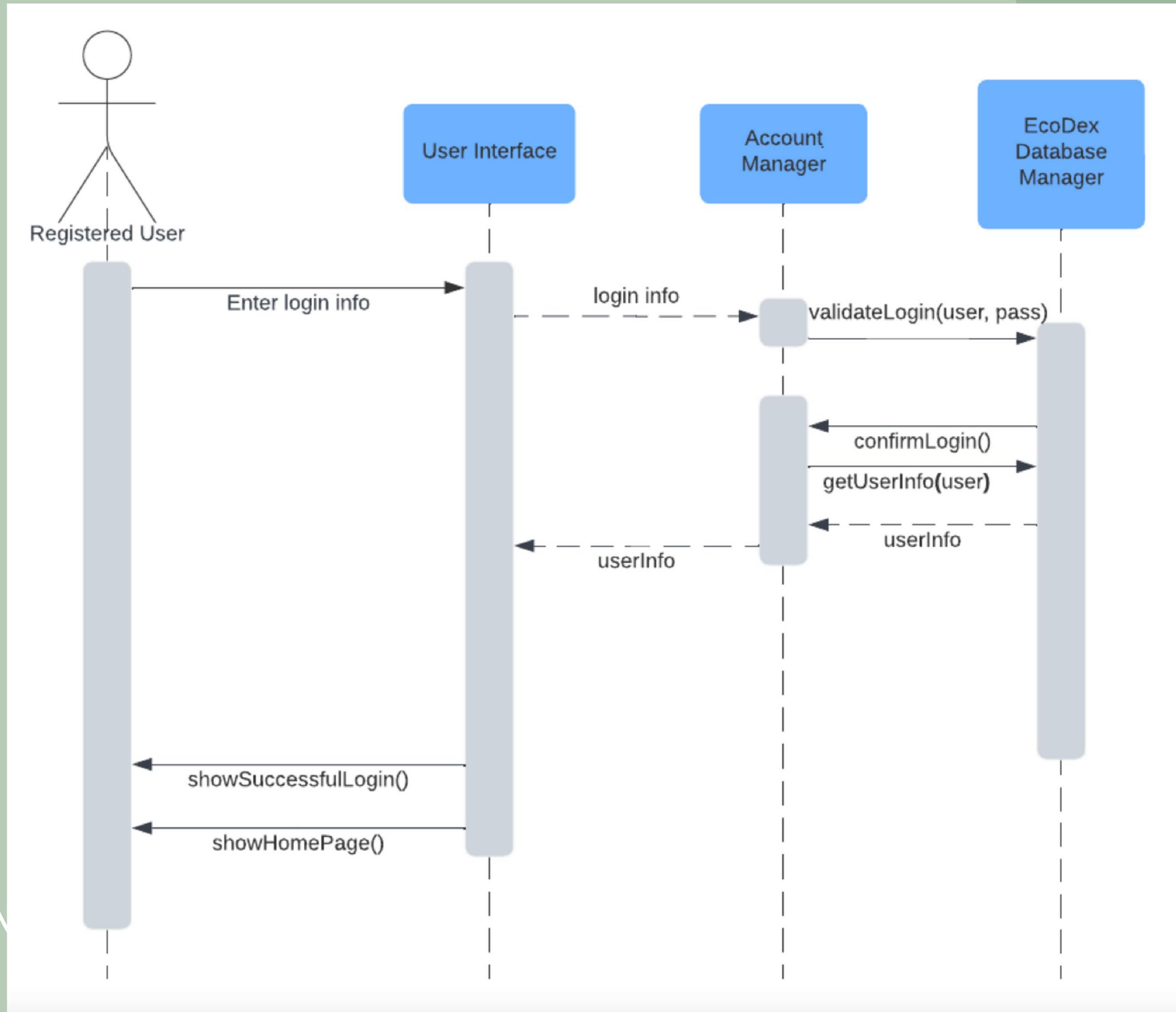
Logging into EcoDex is required for all other functionality, so it is essential to the user experience.

# Sequence Diagram I: Create Log

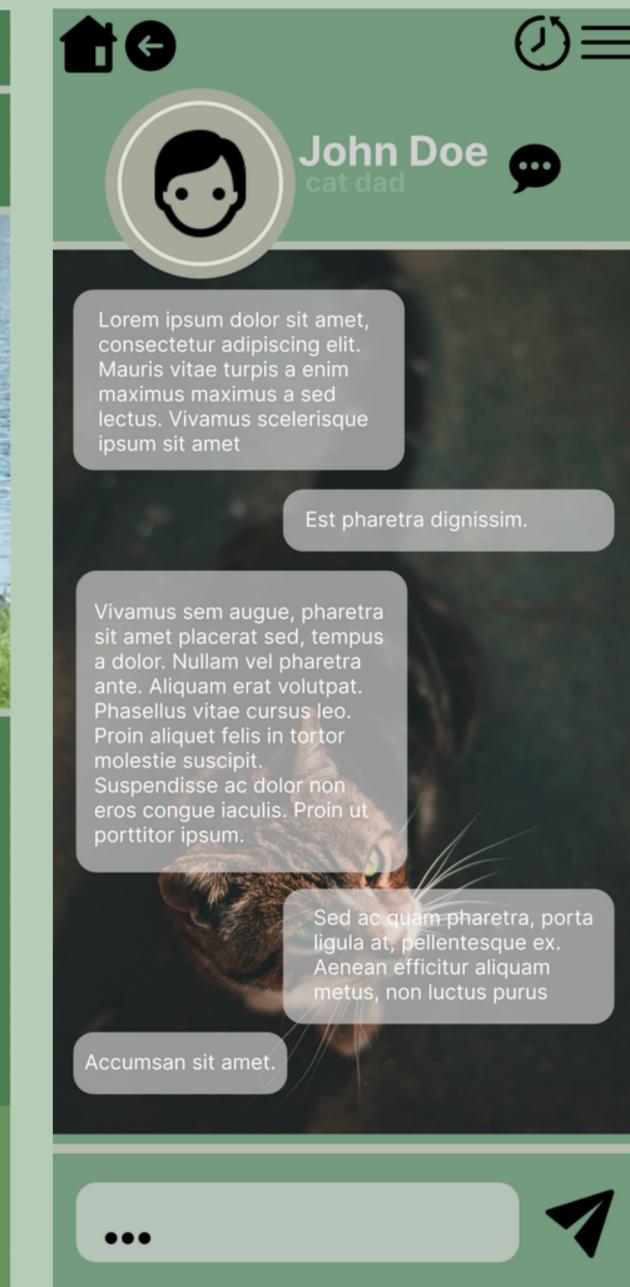
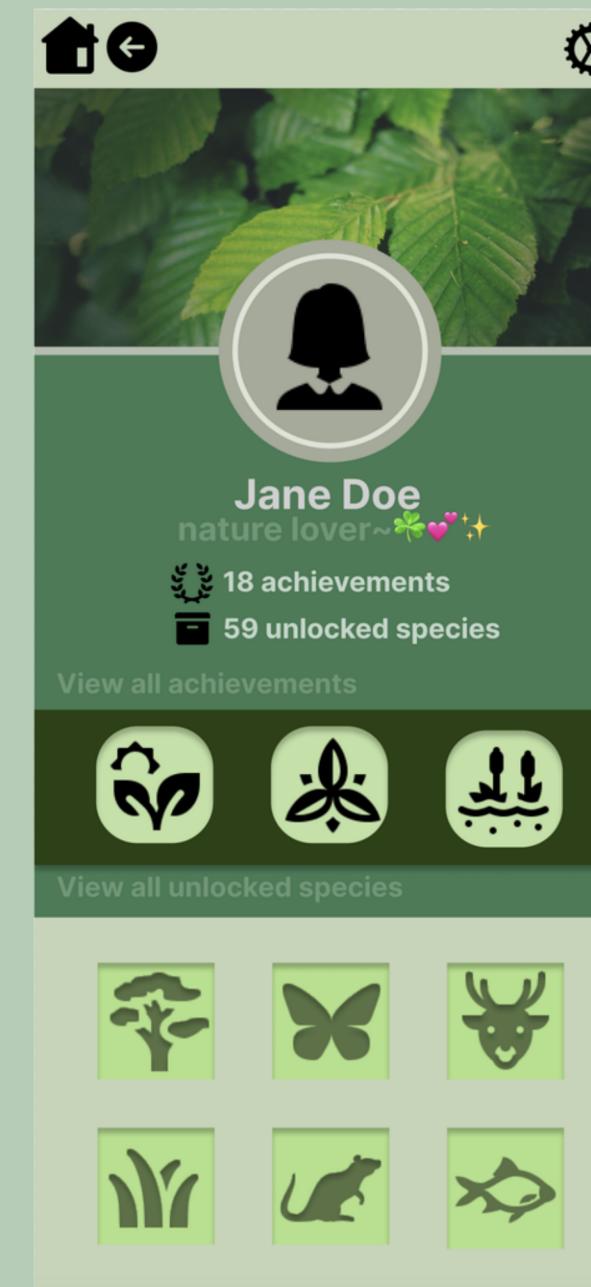
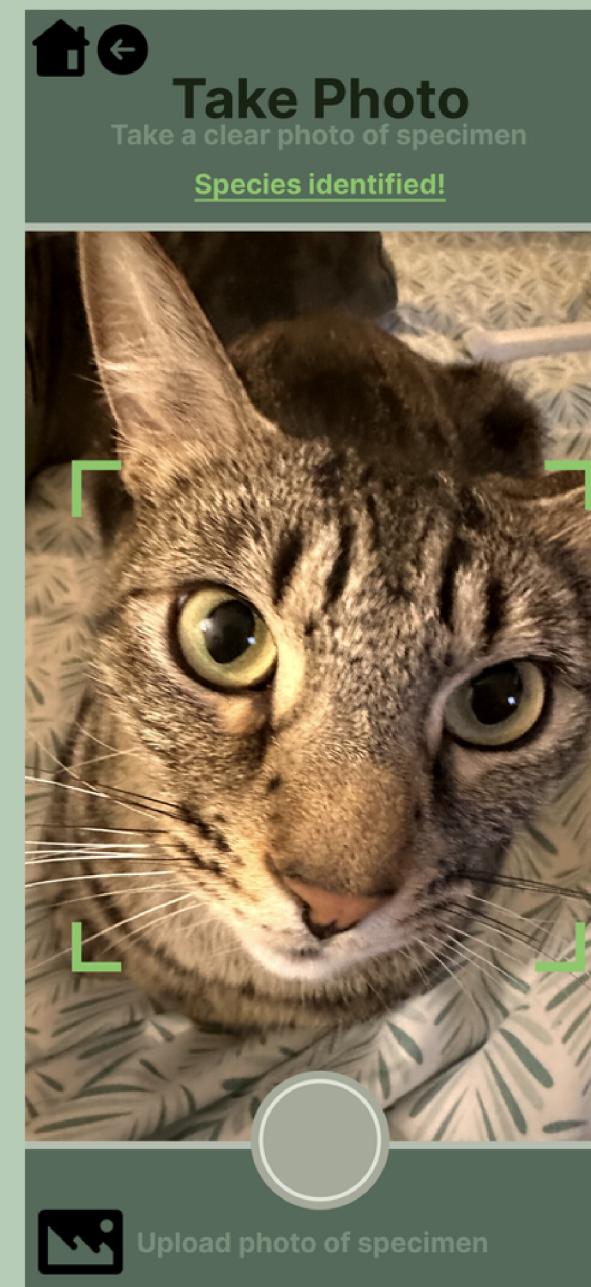
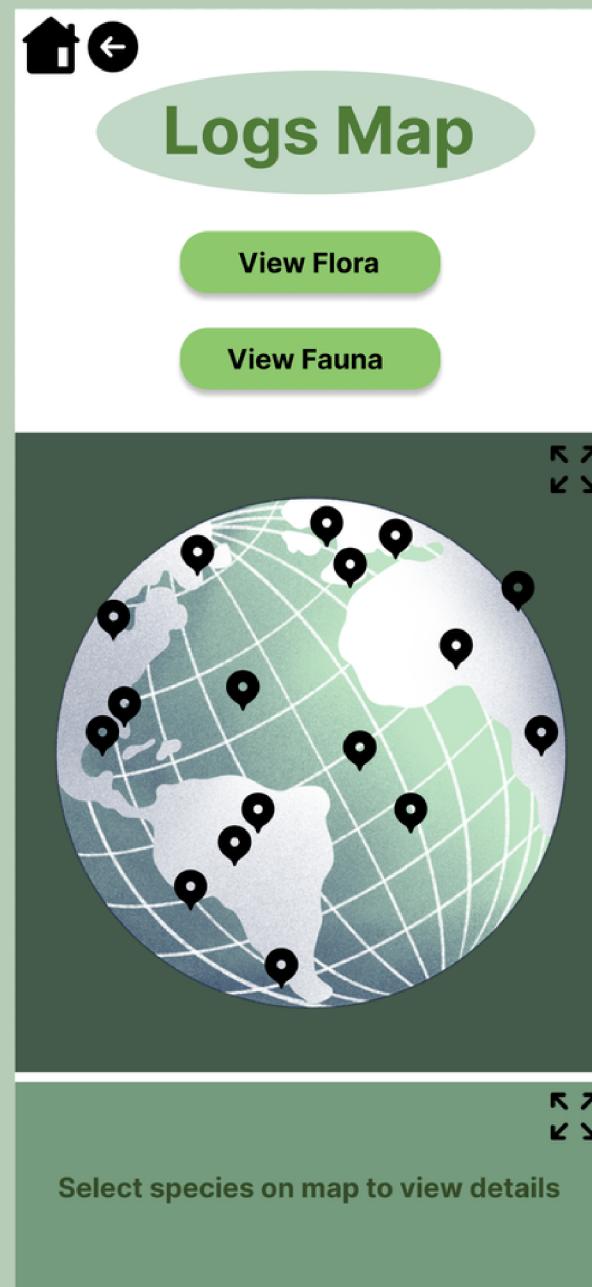
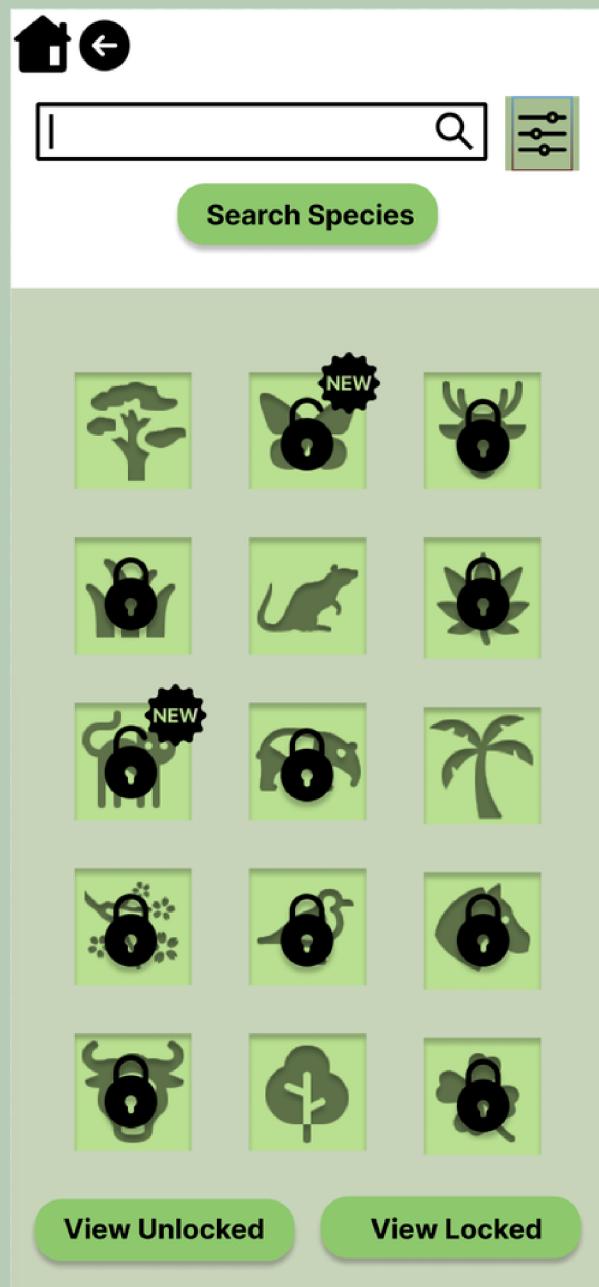
## Sequence Diagram



# Sequence Diagram 2: Login Sequence Diagram



# Application's Screens





Thank you!