NewBird Documentation

Table of Contents

Project Description:

eBird is a bird observation and data collection citizen science project that enables users to record and submit their observations to a real time global database of bird sightings and behaviors. The data gathered through these efforts plays a crucial role in understanding bird populations, migration patterns, and habitat use, as well as informing conservation efforts and research initiatives.

Our program endeavors to integrate with eBird's database to give users a streamlined and user-friendly way to catalog their bird sightings without having to deal with the complicated minutia of eBird's science-focused site. With our program, not only will users be provided with a customized list of commonly seen birds in their region, they will have all the tools and information necessary to identify, log, and submit their sightings directly to eBird.

Process:

- Beginning:
    - Agile Methodology
        - Group Size: 4
        - Personnel: Skill levels varying from 1 - 3
        - Culture: Collaborative and Adaptable
        - Dynamism: Flexible
        - Criticality: Low Impact
- Approaching Interim:
    - Implementing our flexible hierarchy and adaptable framework on both Homework Assignments and project progress.
    - Due to the housing conditions of the group, with two members being off campus and two members being on, outside of our weekly full group meetings, it was much easier to work with Pair Programming.
- Interim/Iteration 2:
    - Due to unforeseen circumstances cropping up around Iteration 2, we had to rework our project slightly and lean into the strengths of the Agile Methodology.
        - The Iterative Process helped us rearrange our priorities and cut or postone certain unnecessary features, it helped us get back on our feet and carry on.

- Iteration 3:
    - Leading up to the completion of Iteration 3, the worst was behind us and we could look forward to sunnier horizons.
        - New challenges reared their heads, like the transition from Python to Javascript with Node.js or the subsequent integration of all that code into React. But as daunting as those challenges may have seemed, our Dynamism supported us well in tackling them.
- After Iteration 3:
    - Post Iteration 3, our focus shifted from creating new features to testing and streamlining our existing ones. Integrating each members' code with everyone else's was the main hurdle here.

Requirements & Specifications:

Use Cases:

- UC1: Login
- UC2: Log Bird
- UC3: Quiz

Actor:

- User
- eBird
- ipapi
- flickr
- xeno-canto
- google maps javascript API

***Quiz* Use Case**

Name:

- Quiz

Primary Actor:

- Bird App User

Goal in Context:

- Take a bird quiz of matching the bird's appearance with the correct name of bird species to test knowledge level and ability to identify birds and learn more about different species of birds

Scope:

- System

Level:

- User goal

Stakeholders and interests:

- User: wants to test themselves on bird knowledge level and ability to identify birds
- User: wants to learn more about different bird species by their appearance and calls

Preconditions:

- User has logged in [UC1]
- User has pressed Start Quiz

Main Flow:

- User logs in [UC1] and navigates to the Quiz option. User presses Start Quiz and chooses one answer among choices of names of bird species based on the given photo of the bird. User gets their score after the quiz ends.

***Log Bird* Use Case**

Primary Actor:

- Bird App User

Goal in context:

- Submit log card at the end of a birdwatching outing to update personal findings

Scope:

- System

Level:

- User Goal

Stakeholders and interests:

- User: wants to save (finalized) findings for bird watching outing
- User: wants to submit bird watching findings

Preconditions:

- User has logged in [UC1]
- User has pressed Start Bird Log
- User has logged at least one sighting

Main Flow:

- User logs in [UC1] and navigates to the Bird Log option. User

Guarantees:

- User has saved record of bird sighting information for respective outing
- General bird sighting information (quantity, type, location) has been updated for individual user
- General bird sighting information (quantity, type, location) has been updated for eBird database

Trigger: user logged last bird sighting

1. User logs last bird sighting for their outing
2. System has successfully saved each logged sighting
3. User presses Submit Log

4. System verifies that log wasn't empty
5. System confirms request and saves log card information

User Stories:

User Story: Creating, updating, and submitting a Bird Log to eBird is one of the most important and integral features of our app.

- **Title:** Log Bird
- **Acceptance test:** Verification of accurate submission to eBird under user's account.
- **Priority:** 1
- **Story points:** 5 days
- **Description:** After logging into the app with their eBird credentials, users will be able to start a Bird Log session, viewing entries and logging sightings as they go. Once they have finished logging their sightings, they can submit them to eBird's database.

User Story: As a bird enthusiast, I want to start a quiz about birds in a specific location, so that I can test my knowledge about local bird species.

- **Title:** Quiz
- **Acceptance test:** QuizGeneration&ContentRelevance
- **Priority:** 2
- **Story points:** 1 day
- **Description**: Start quiz generates a study set of birds relevant to the user's location data. This feature is intended for when the user wants to learn about and memorize the different birds in their selected location (prior to a bird watching outing) to better recognize the different species they may encounter.

Architecture & Design:

Use case diagram:

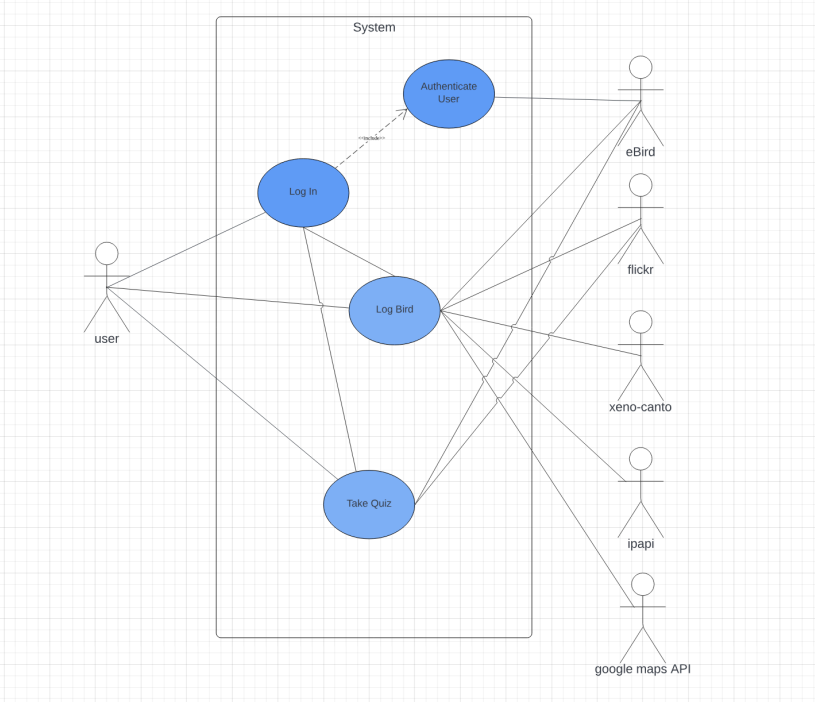

Figure 1. Use Case Diagram

Class diagram:

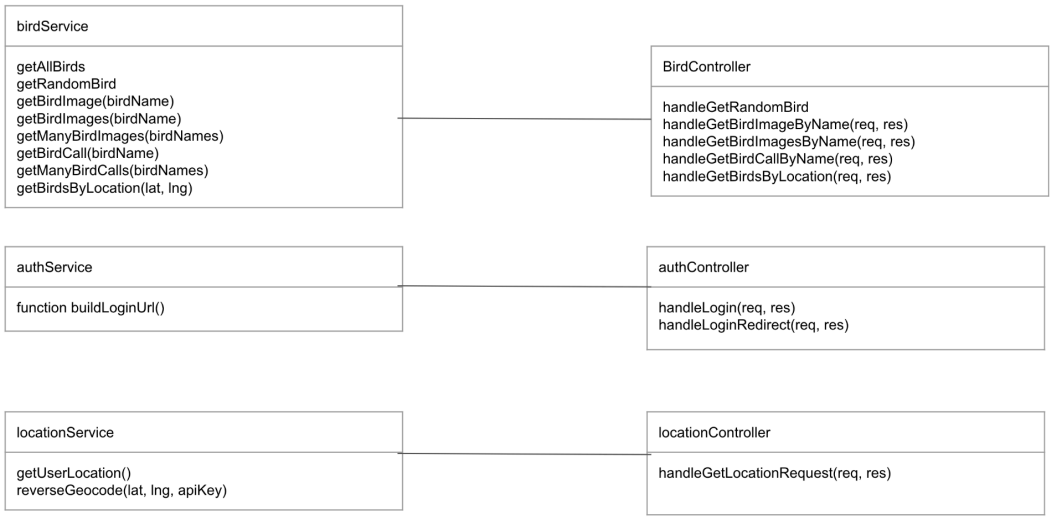

Figure 2. Class Diagram

The architecture of our system:

The overall structure of our system is divided into two layers: the frontend and the backend.
Frontend:

The frontend layer utilizes the React framework. React handles the user interface and interactions.

Backend:

In the backend, we follow a layered architecture structure based on MVC architecture. This way, we can change the frontend without affecting the backend, and vice versa. We have a service layer and a controller layer. The service layer (model layer) is responsible for the data and API logic of the application. The controller layer allows users to 'talk' to the service layer and request information. It handles user interface, processes the data and coordinates the responses.

How did the choice of framework influence the design of the system?

- Separation of concerns: the use of MVC provided us with a modular and organized code, where the logic behind the layers is easier to be managed.
- Scalability: by separating the layers we have a more scalable and maintainable system. This enables us to make changes and enhancements to one layer without affecting the other layers, improving the maintainability of our code
- Overall, the choice of React for the frontend and a Layered MVC architecture for the backend influenced the design of our system by promoting modularity, scalability, maintainability and flexibility, making our system more efficient.

Reflections and Lessons Learned:

Anna:

This class gives me the opportunity to work on a whole software engineering group project for the first time. I got to learn about the development process from practice. In the beginning, I thought that we would be following the architecture we designed, but it turned out that adjustments were needed based on the coding progress and problems we encountered later on. Learning from practice also applies to my coding skills. Recognizing the extensive coding requirements, I did a lot of research, self-learning and self-teaching for the coding of my responsible part of the project. It is a steep learning curve, but I learned what I wouldn't have learned in class from this process. Coding for a group project is different from an individual project. I need to have the big picture and understand the architecture and structure to consider the code integration while coding. Now, at the end of this project, I feel that my coding skills improved a lot. From the class and the project, I learned that software engineering is not just about coding. There are other important parts that need to be considered as well such as architecture, use cases, design patterns, etc. I appreciate that the project allows me to apply what

we learned in class to the real world, turning them from pure knowledge into practical techniques.

Bisera:

I came into this class as a total beginner in the field of software engineering, or more specifically, the processes and methods which make up SE. Coming into it blindly, the structure of this class helped me explore concepts one by one and connect them in order to try to deliver a final product. The first thing I learnt was the difference between the methods, and figured out that our group was most definitely gonna follow the agile methodology based on the criticality and the size of our team. One of the biggest things I learnt in this class is what happens in the backend of a system. I learned how to fetch information from APIs via HTTP requests, how to structure my code into layers in order to have my logic separate, how to test the code, and all of the different design patterns that I can encounter. I also learned a lot about MVC architecture, especially during the beginning of the project, as I had to set up the backend of the program based on this architecture. My biggest challenges in this course were integrating my code with my team member's code and putting everything together for a final project. Another one of our challenges was working together and communicating in an effective manner. At the end of this project, even though the end result is different than what we had planned for, the main thing I got out of it is very valuable experience and knowledge related to the software engineering processes.

Chloe:

I'm glad to have worked on this project, up until taking this class I hadn't learned much about back end logic when it came to web development. And although this course was challenging, I think the experience I got working with MVC architecture will be very useful to me in my career. With respect to some of the challenges we faced, there was a bit of a discrepancy in experience which led to a slightly disorganized production process - we initially worked on components separately, (and later had to integrate them) and when we tried to integrate them, realized that much of our individual logic didn't work together. For instance, we had intended to use REACT for the frontend and MVC architecture and later realized that they are not meant to function together (and integrating them is quite difficult). Due to this, we had to make quite a few changes when finally submitting our project.

Jay:

Going into this project, I was skeptical about the amount of real world value and experience that it would be garnering from it. Not many of the classes that I've taken so far have given me any practical knowledge or experience beyond learning theoretical concepts and programming practices, so my hopes were not high. So imagine my surprise when the course covered software development practices that were outside of a classroom setting. Learning about different methodologies and implementing them actively into our development process was a valuable opportunity to explore real world team dynamics.

Even if everything had gone completely right in our development process, this project would have still been a valuable learning experience, but everything certainly did not go right. Around the time of the Interim presentation and Iteration 2, our group ran into multiple large logistical challenges that set us back a lot. Recovering from this challenge was difficult, but taught me how to handle unprecedented stressful situations in big projects.

Overall, I have gained valuable insight into the dynamics of small group collaboration within a semi-professional setting, leading to the attainment of shared goals. More importantly, I have learned how I can be most effective when a part of said group.