

Facebook for Birds

Summary

To make a front-end user interface and back-end services for the JP Chickadee Project. JP Chickadee Project is a cross-disciplinary team of computer science and biology students. We are capturing the time and ID of RFID banded birds when they visit specially outfitted feeders. This is cool, but the data is not readily available to students and community members. This website's aim is to change that.

I have already created a basic website. Now I would like to add more statistics and have back-end services for them. There could be bird profiles with recent activity and links to their friends' profiles. (Birds who visit the same feeder within a certain timespan are friends or enemies if one quickly displaces the other.) Graphs showing when birds are most active throughout the day. A count of the total number of visits recorded since the start of the project.

Last semester Michael Whalen created a simple CRUD API for assessing visits, birds, and feeders. This is good for simple operations. But shows weakness when trying to do awesome analysis. So say for example, we wanted to display the total number of recorded visits ever on the website. To do this, each time the page is loaded, we would have to request a list of all visits and then count them. This is how I have it currently done. All statistics and maps would be the same; request all the data and compute the summary on the client side every time the page is loaded. Boo, too slow!

[Live website](#)

[Total visit count the bad way for fun \(it takes a minute\)](#)

[Graph of visits for a bird the bad way for fun](#)

Learning Objectives

I would like to crate statistical bird-analysis stuff as back-end services and have the client request the current summary. The statistics would be a layer added on top of the basic API. The website design would be modular and made of small reusable components.

There is a delay between when a visit is recorded and when it is added to the API. And visits can be added out of order. The current API doesn't have an endpoint for newly uploaded visits. So I need to figure out how to handle this.

microservices vs monolith and determine which is right for this project.

If service A uses data from service B:

1. Should A regularly ask B for data?
2. Should B regularly give data to A?

Features

I would like to get points for each service and user interface I implement.

Some possible services ideas:

- Total visits ticker
- List of feeders and most recent visit
- each bird's friends ★
- each bird's enemies
- each bird's favorite feeder ★
- Which feeder is the most popular right now ★
- Which bird has the most friends
- Which bird is the most dominate
- Graph of bird visits for the past 1, 7, and 30 days
- Path of bird from feeder to feeder
- Heat map of bird activity
- Graph showing the frequency that birds go from feeder A then to feeder B. ★
- Bird leaderboard, who visits the most
- Other ideas as they come...

Points per service

Back-end

Service design (**select one**)

A service that knows the internals of the MySQL database.	0.75
A service that gets <i>all</i> the bird-data from the API on a schedule and recalculates the stats.	1.5
A service that only gets <i>newly added</i> bird-data from the API or another service.	3

Other

The statistical portion of code has unit tests.	0.5
Errors are logged and handled nicely.	0.25

Front-end

Modular component	1
Updates without page refresh	1
Looks acceptable	0.5

Development Environment

• Setup a testing database	1
• Setup a testing API	1
• Write a script that seeds the database with fake data	1
• Write a script that continuously submits fake visits to the API	1

David Germain
CS 480
Winter 2018

Grading Scale

22 ≤	A
20	A-
18	B+
16	B
14	B-
12	C+
≤10	C

Technologies

- RESTful APIs
- Node.JS
- Vue.JS
- Sockets

Course Overlap Disclosure: I am enrolled in a 1-credit course under Alec Lindsay called “Lab Exp: Applied CS/Bio Res - BI 391.” I help with all parts of the JP Chickadee Project with a focus on making the basic website. Before this proposal, I would have left implementing the services and many of the statistical computing to someone else at a later semester.

The image shows a screenshot of a Facebook profile for a user named 'Chickadee McGee'. The profile picture is a photograph of a chickadee bird perched on a branch. The cover photo is a dark, solid color. The profile name 'Chickadee McGee' is displayed in white on the dark background, with 'Add Friend' and 'Message' buttons to the right. Below the name are navigation tabs for 'Timeline', 'About', 'Friends', 'Photos', and 'More'. The main content area shows a post from May 17, 2011, with the text 'love the bird seed at HQ' and tags for 'Frank the Bird' and 'Charlie'. The post has a 'Like' button, a 'Comment' button, and a 'Share' button. Below the post is a comment from 'David Germain' with a 'Write a comment...' input field and icons for emojis, photos, GIFs, and stickers. At the bottom, another post from 'Chickadee McGee' is partially visible with the text 'Chick a dee dee dee'. On the left side of the profile, there are sections for 'Intro' (listing 'Lives in CLIFF' and 'From Punxsutawney, Pennsylvania'), 'Photos' (Nothing to Show), and 'Friends' (13 friends, with three small photos of chickadees).