Butterfly Longevity Tracker

Bi-Weekly Report 6 (4/4/25 - 4/17/25)

Project Information

- Project Title: Global Butterfly Longevity Tracker
- Group Number: sdmay25-03
- Client: Nathan Brockman
- Advisor: Maruf Ahamed
- Team Members:
  - Alex Herting - Full-stack Engineer
  - Andrew Ahrenkiel - Full-stack Engineer
  - Charles Dougherty - Frontend Developer
  - Jaret Van Zee - Backend Engineer
  - Carter Awbrey - Visionary

Summary

On the frontend we wrapped up all of the core functionality besides the generate report functionality. The big thing we did was brainstorm ways to implement this feature as well as start the development of our updated solution. This took up a lot of our time over the past two weeks. We also have been making enhancements and minor changes to the applications functionality according to our clients feedback from his testing.

On our backend team, we focused heavily on improving system stability, performance and memory efficiency this week. We tackled back-end optimization by analyzing and enhancing API endpoints, implementing stream-based reporting to reduce server memory usage, and beginning the migration of database reports to streaming endpoints. On the infrastructure side, we configured AWS hosting with CloudFront, restructured deployment pipelines, and set up Docker for smoother app deployment. We also made progress on key features, including import functionality for user data and a new, optimized backend system for querying butterflies.

Accomplishments

- Alex
    - Worked on refactoring generate report functionality
    - Made a plan for the frontend in order to efficiently grab data from the backend
    - Made a list of functionalities that frontend will need from backend in order to generate thorough reports
    - Added all tag definitions to the generate report screen
    - Added logout functionality that clears local storage for each page
    - Assisted in debugging the importing feature of legacy data
    - Updated logout for guest to take them to the proper domain home page
    - Communicated with client on updates we can make to our application as well as provided solutions to new "problems" that have come up as new users test the product.
    - Participated in code reviews
- Andrew
    - Worked on updating the sighting enter page for admins based on client's request.
    - Looked into and tested aggregates for the generate report querying to accept list of domains
    - Updated styling bugs on the admin home page
    - Updated all screens to use the proper logo and tagging image per the domain.
    - Updated the "update facility screen" to allow an admin to change their logo and tagging images.
    - Updated the Access Data request to work for domain and super admins correctly
    - Looked into updating the sighting entry response for admin users, updated the response body of the request, awaiting implementation.
    - Participated in code reviews for peers
- Charles
    - Import Function implementation to correctly parse and send mass data to the backend
    - Styling and functionality for the import webpage
    - Started implementing tag auto-filling
    - Worked with the backend to figure out solutions to crashing issues
    - Brainstormed solutions to new issues
- Jaret
    - Discussed changes to the current design document
    - Examined the overall performance of several different back-end API endpoints
    - Build out importing features to allow users to import existing data into their facility

    - Built out caching methods in the backend for efficiency
    - Code reviews
- Carter
    - AWS to Local DB Storage
    - Configured AWS lightsail hosting, DNS and Cloudfront proxy
    - Updated Pipelines to deploy to new server properly
    - Updated Gitlab Repository structure
    - Docker Setup and Deployment of Spring App
    - Optimized MongoDB Aggregation pipelines
    - Created and Optimized a new system for querying butterflies on the backend
    - Implemented stream based report features to optimize memory usage on server
    - Began migrating database reports screen over to streaming endpoints instead
    - Worked on issues related to poor server performance and stability

Pending Issues

Front End

- Complete UI functionality
- Generate report statistics and data results
- Add cookies (some sort of increased security measure)
- Deleting existing domain data (Archiving possibly)

Back End

- Unit Tests
- Optimize features (as necessary)
- Improve Input Validation
- Create API documentation with “Spring REST Docs”
- Finish Configuration for long-term hosting site (AWS)

Individual Contributions

NAME	Individual Contributions	Hours this week	HOURS cumulative
Alex Herting	Generate report screen, planned request implementation, implemented tag definition for individual butterfly reports, logout functionality, debugged importing data, client communication, code reviews	20	166
Andrew Ahrenkiel	Sighting enter page now has calendar input option for admins, fixed styling bugs, collaborated to refactor the generate report aggregates, updated all pages to use proper domain images, updated the request for “update domains” to accept and change the logo and tagging images, participated in code reviews, started implementing changes based on client requests.	20	155
Charles Dougherty	Import previous butterflies functionality and styling, tag auto-filling, and solving crashing issues on the backend, brainstorming solutions to new issues	20	123
Jaret Van Zee	Discussed changes to the design document, reviewed back-end API performance, and started building import features for user data.	20	138
Carter Awbrey	Configured AWS Lightsail with DNS and CloudFront, updated deployment pipelines and GitLab structure, set up Docker for the Spring app, optimized MongoDB aggregations, built a new	30	172

	butterfly query system, implemented stream-based reporting, began migrating reports to streaming endpoints, and addressed server performance issues.		

Future Plans

For the frontend, we plan to complete the report generation logic as well as implement the different tagging systems into the data display page. We are going to look into archiving old data for domains that wipe data after a certain amount of time in order to keep data efficiency and practicality of the application. We are also receiving feedback from our client and other users that are testing the product that tell us what changes need to be made. We do not know exactly what we will be developing apart from the report generation, but we are sure that we will be making adjustments to the application as testing continues and grows to a wider user base.

For the backend, we just need to tidy up some things related to the reporting logic, optimize some data import features and add in a few miscellaneous APIs that we haven't gotten around to yet. Including, but not limited to, getting the next available tag, getting sighting statistical information.