AlgaeEstimator

Allen Rocha
Computer Science
University of Wisconsin-Parkside
rocha021@rangers.uwp.edu

Jose Martinez
Computer Science
University of Wisconsin-Parkside
marti265@rangers.uwp.edu

Antero Nevarez-Lira
Computer Science (API)
University of Wisconsin-Parkside
nevar001@uwp.edu

Wyatt Wells
Computer Science
University of Wisconsin-Parkside
wells039@rangers.uwp.edu

Zaid Altahat
Computer Science
University of Wisconsin-Parkside
altahat@uwp.edu

Abstract

The design and development of a mobile application to predict harmful algal blooms (HAB) is described. Its main feature is to predict the potential of HABs based on the data provided by the user. Users create an account using an email and password. All submitted test results are stored in an online database, Google's Firebase . The secondary feature uses the data stored on Firebase to draw out a map that allows users to view other users' submitted experiments in an easy to understand design.

Introduction

Algae is used when referencing a large group of microorganisms that are not closely related to each other. Algal blooms are rapid accumulations of algae populations in a body of water that can cause discoloration of the body of water. Blooms, often caused by runoff from fertilizers entering a body of water, can affect its entire ecosystem by blocking the sunlight from the organisms below it, depleting the lake of its oxygen, and releasing toxins into the lake. Even when algae blooms do contain species that can release toxins into the bodies of water, these blooms can create “dead zones”, where the low oxygen levels makes it uninhabitable by aquatic life. Across the world people are feeling the effects of HAB and they may not even be aware of it. Officials have closed many lakes due to the presence of many HAB. Lake Erie, Utah Lake, Hells Canyon in Idaho, Pyramid Lake and Shasta Lake are just a few of the impacted lakes. Although there are few toxic algae species when viewing the thousands of spices [1], a study led by Florida Tech revealed that a neurotoxin was present during bloom periods and non-bloom periods along the Florida Atlantic coast [2]. These toxins released into the bodies of water can severely harm or even kill animals and humans. After an analysis of the liver cells from dead bottlenose dolphins, Florida Tech researchers discovered the 119 dolphins had the neurotoxins brevetoxin and saxitoxin during algal blooming periods and non-blooming periods [2]. Of the discovered HAB species two of them, *Karenia brevis* and *Pyrodinium bahamense*, produce the neurotoxins [2]. Although humans are not necessarily in constant contact with dolphins, it is very possible that neurotoxin entered the bottlenose through biomagnification. Biomagnification is when certain substances enter into bodies of water, and then move up the food chain in progressively greater concentrations as they are consumed by predators. The fears of possibly poisoned fish creates a substantial impact on local economies that are driven by fishing, tourism, and impacts marine based jobs [3]. In 1983 and 1984, there was a mass dieoff of the grazing sea urchin *Diadema antillarum* all across the Caribbean coast after years of increased presence of HAB [4]. Immediately after the mass deaths, the algae population continued to rapidly increase but their productivity decreased by twofold [4]. Along the Jamacian coasts, scientists have linked HABs towards the rapid deterioration and bleaching of the coral reefs [5]. Although HABs can be caused by runoff, blooming periods have also been linked to the overfishing of herbivorous creatures [5]. Where eutrophication is possible, algal blooms can cause great risks to both the environment and creatures around it. Therefore, an application to allow people with or without knowledge of HAB, algae, or the ability to predict HABs is not only useful for the scientific community, but the greater public as well.

Design Approach

The apps have several components, the Android app, the iOS app, the backend API, and the admin dashboard. Each component was worked on by a different team. The apps were done natively using Kotlin and Java on the Android side and Swift 5 for the iOS app. The backend API was built using Nodejs running as Functions on the Google Compute Platform. The database is hosted on Firebase. The admin dashboard is built using Vue and is hosted on Firebase storage.

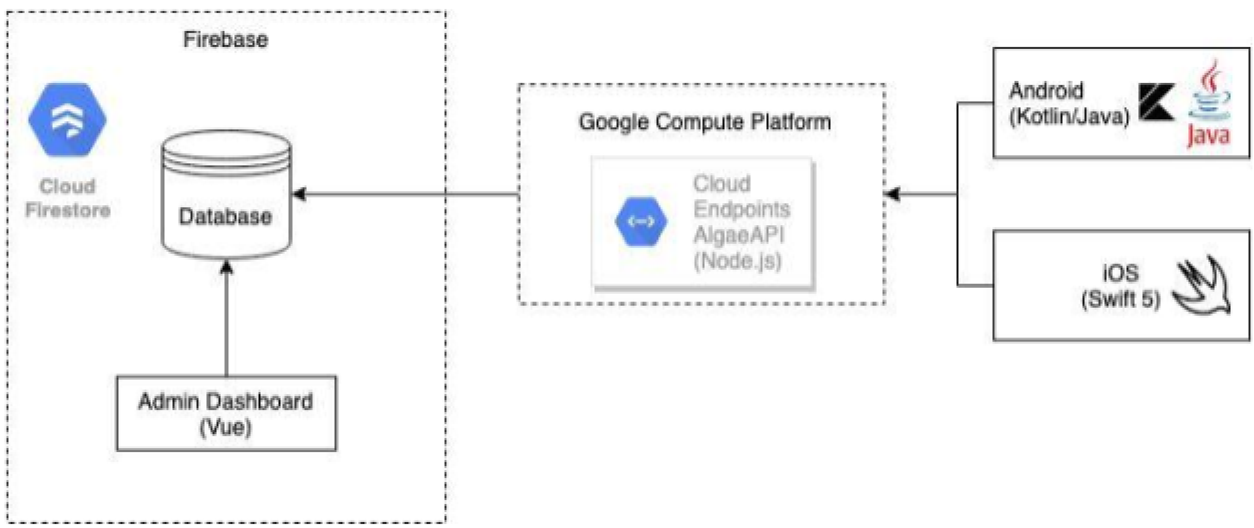

Figure 1-1: Data Design

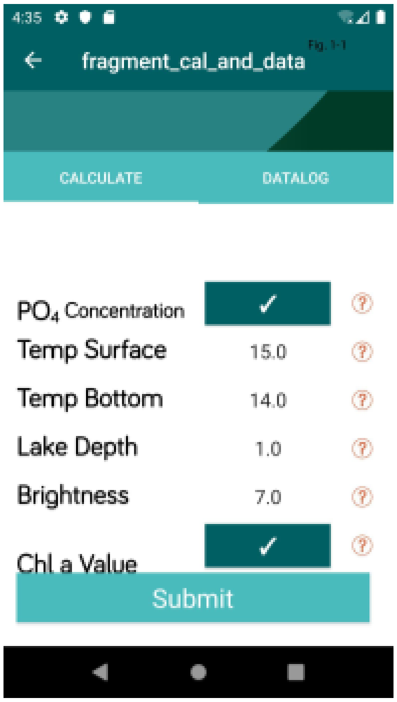

Figure 2-1: Data Input

The AlgaeEstimator application was originally launched in 2016 by University of Wisconsin - Parkside students [6]. Our objective was to redesign the app to make it more user friendly. The integration of Firebase allowed us to redesign the application so that users can keep all their experiments in the cloud. Firebase allows users to securely create an account, and securely log in and out of their account .With their created account, the users are able to create experiments and save their experiments to their account so that you can access those experiments from any platform. As shown in Figure 2-1, one screen allows the user to input data. The PO_4 Concentration button opens a menu that allows the user to either enter specific PO_4 or let the app make an estimate based on the environment (Urban/Farmland, Natural/Lawns/Sandy), a visual estimate of the amount of plants in the body of water, and the frequency of algae blooms in the lake (if the user has that information).

Figure 3-1: Data Results

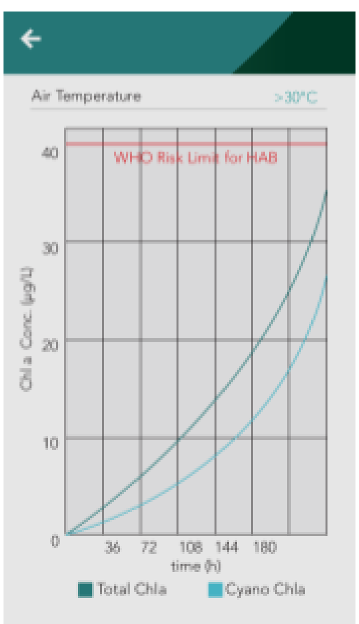

Figure 3-2: Graph

The “Temp Surface” field allows the user to enter the surface temperature of the lake in degrees Celsius . The “Temp Bottom” field allows the user to enter the temperature of the bottom of the lake in degrees Celsius. “Lake Depth” allows the user to enter the depth of the lake in meters.

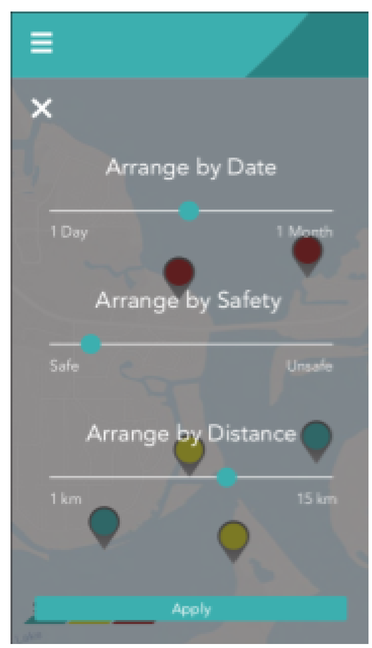

Figure 4-1: Map Settings

“Brightness allows the user either manually enter the brightness in lux, or allow the phone’s camera to measure it. “The Chl a Value” button opens another menu that allows the user to either manually enter the values for Total and Cyano Chl a, or again allow the app to make an estimate. This estimate is based on Secci Depth in meters and Dissolved Oxygen (percentage). Next to the fields to enter the data are tiny question marks that describe what the corresponding field is, and information on how to gather the data.

Figure 3-1 allows the user to see the results, and a graph (Figure 3-2) of the data calculated. The main requirement for this application is prediction of HAB. To accurately predict for HAB the user must take quantitative measurements, requiring specific field tools with them. This requirement limited the target audience to only those that knew how to correctly use those field tools. With the newest version of the application, the new home page has a way to visually estimate for potentially HAB in which the user compares their location with images of diverse levels of algae provided within the application. This new feature increases the target audience that allows not only lake/water management institutions to effectively use it but also, adult users into camping or water related activities. Other new major features implemented are Firebase integration, a map, and an FAQ.

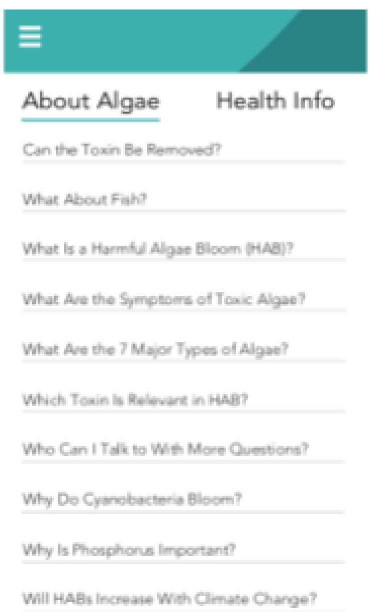

Figure 5-1: FAQ

The user can navigate to the newly implemented map page by pressing the “Map” button. This page allows the user to view the test results sent at locations of other users’ as markers colored as blue for safe, yellow for unsafe, and red for dangerous. The markers safety scale is dependent by the phosphate levels, history of HAB, and the ratio of cyanobacterial Chlorophyll a per liter. The location markers’ color is generated automatically and is sent to Firebase along with the experiment when the user submits their data. Along with the map, users can filter out markers' safety, distance (a max of 50 km), and select a range of dates to show. When pressing the generated map marker, you are able to view that experiment.

The user navigates to the FAQ (Figure 5-1) by pressing the "About Algae" button and displays a page with many of the often-asked questions about algae and algal blooms. Each of the questions are interactable and lead to an answer to the asked question. Along with the frequently asked questions, the "About Algae" section also gives information on who to contact if you need help identifying HABs.

Backend

During the research phase, one of the main concerns was storage: How could the application function without using the device's internal/external storage drives? The best solution was Firebase. Known as Backend-as-a-Service platform, Firebase provides a wide range of tools for mobile and web development, one of those being the real-time database. Integrating the real-time database meant that the user could send all of the specifications and their current location to be stored and have access to the data in, as the name suggests, "real-time" across any device that the application is installed on. Firebase also takes advantage of the fact that it is backed up by Google, and provides a storage solution that utilizes Google's Cloud Storage. This meant that the user could take as many pictures of potential HAB without having to worry about reaching device storage limits.

In its current state, the application functions as follows: The user creates an account which is handled by Firebase authentication for security. Once logged in, the user is able to submit all recorded data, including location of surveyed area (latitude and longitude), which is then sent to the real-time database. Afterwards, the user is prompted to upload an image of the location, which is sent to Firebase Storage. As it stands, this setup works, however, it is not the most optimal solution. Two of the possible obstacles are API call limitations and storage size limitations. Due to these concerns, we are currently working with the Software Engineering class API team to redesign the backend in order to lower the amount of API calls made at any given time, as well as limiting or possibly removing the need to store physical files, as a way to make the application as efficient as possible.

Mobile iOS

The iOS started off by having to update their application to work with Xcode version 11. The app consists of multiple screens as seen below.

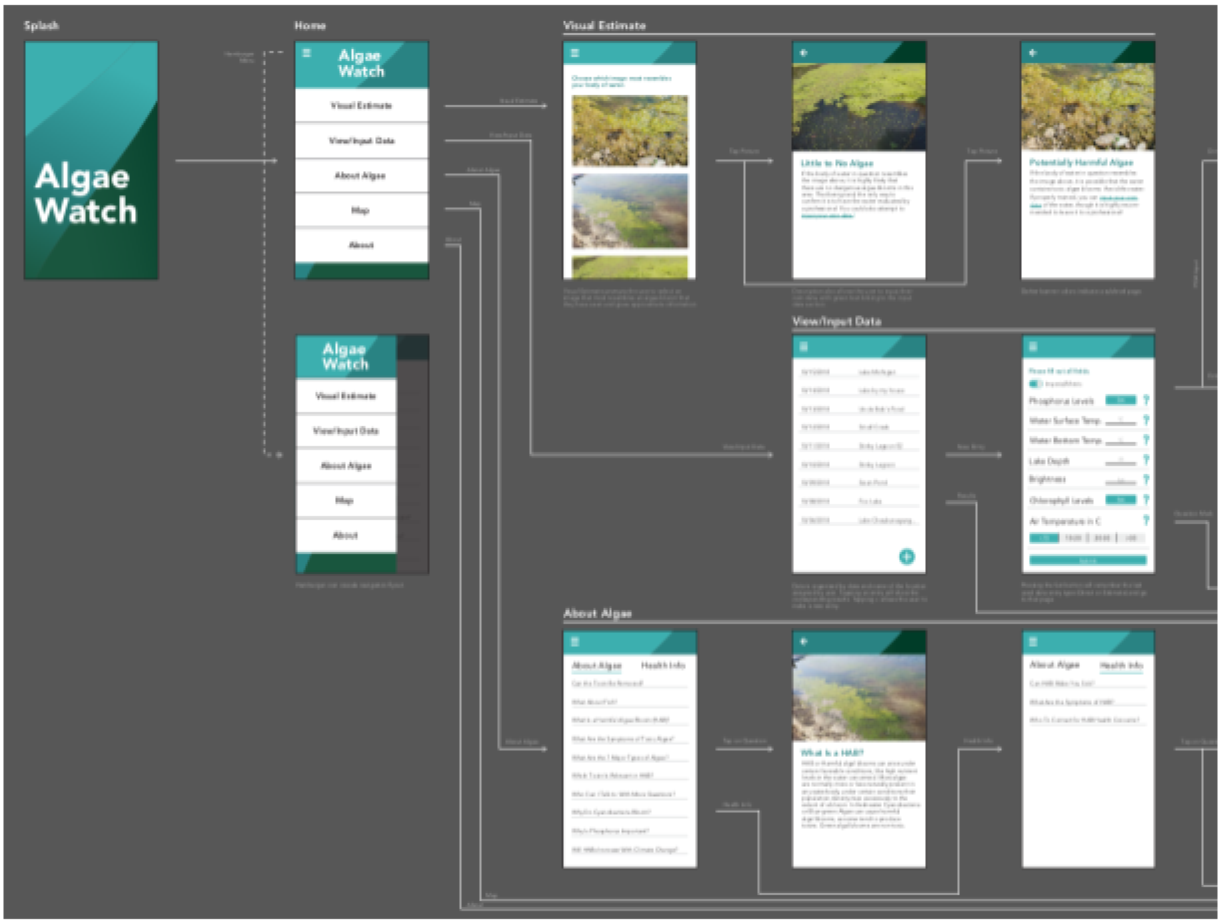

Figure 6-1: UI Overview

Implementing a home view to control the flow of the application as well as adding a map, about section, and login views. The login view was an added decision after our team received the UI storyboard. Along with this view (Figure 6-2) we included a signup and password reset option. Implementing firebase, we are able to add a new user to the database and send them an email to reset their password.

The map implemented from MapKit for iOS, asks for permission to access the location of the user. This is done so that the application can add a point on the map of the tested area. The "About" view defines key terms, answers key questions, the purpose of the application,

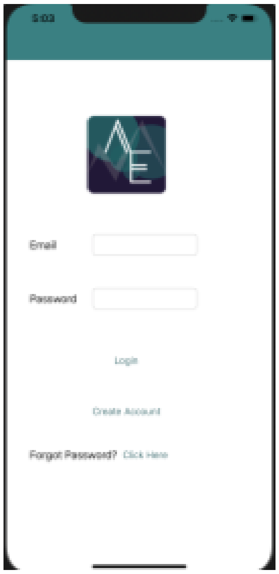

Figure 6-2: Login

disclaimers, and allows the users to give feedback on their experience. All this information is retrieved from the application's Firebase or sent to Firebase. Login as well as signup takes the user's input information and adds it to firebase, when login in the information is pulled from firebase for authentication.

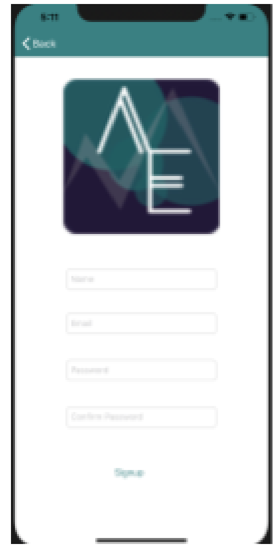

Figure 6-3: Signup

Mobile Android

One of the first aspects of the Android version of the application worked on was the upgrade to version 28. Some Android specific features implemented into the development of the application is the MapView which helps offload the work of creating a custom map for this application. This map has a reference image in the bottom left to help quickly identify each marker by its color. As stated previously, the map features markers whose colors change depending on the result of the test. When taking a photo when submitting an experiment, the application also

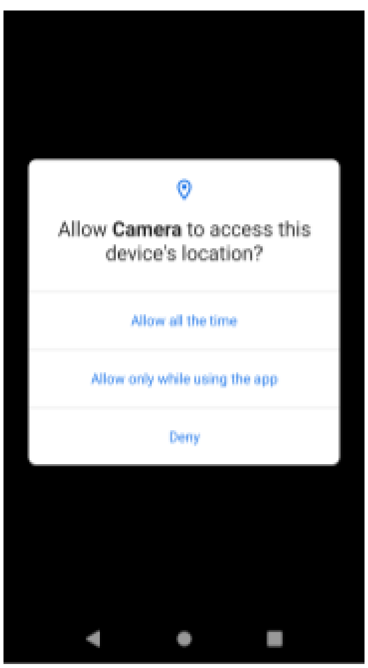

Figure 7-1: Map Permissions

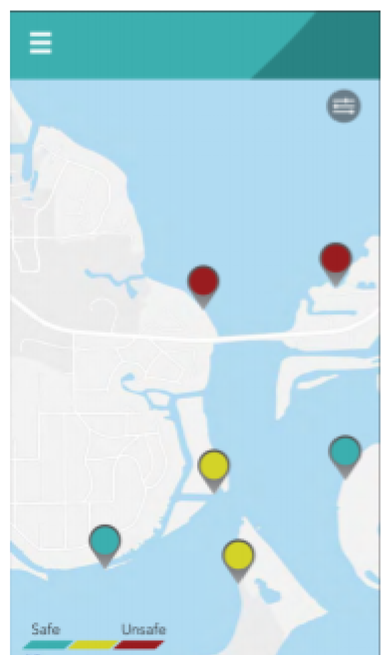

Figure 7-2: Map Markers

Process used

The use of the Agile Scrum process was instrumental in making progress where sprints ran for two weeks. Atlas Jira is the tool used to manage the project which was key to stay organized. Microsoft Teams is the communication platform used by the team. Scrum is an agile approach for developing innovative products and services. We have decided to use the scrum approach because of its simplicity and flexibility. The use of an agile approach helps a team respond to unpredictability. Agile approach combines iterative and incremental approaches.

Challenges

This project was created by another group of students from 2016 and was not updated since its first release. So, when the new development team first opened the project files everything was outdated and required many upgrades. For the first few weeks both Android and iOS teams had to work towards upgrading their versions of the application, the most current SDK. After that we also had the issue with two of the group members never working on Android development, so they had to learn quickly working with the easier aspects, to continue with the more difficult aspects of the application. Another major hurdle we had to overcome understanding the poorly documented or sometimes undocumented code. That made the process of upgrading more difficult due to certain deprecated libraries that were key to the application, and we would not know their purpose as to find replacements.

Our first goal was to simply update the UI with the design created and provided by students at the University of Illinois – Chicago (UIC). The UI they created had a more modern feel compared to the older version which was very outdated. But this UI had some issues since it was created without thinking about the functionality, so we had to improvise and, in some cases, refer to the client for clarification on some design specifications.

During our first meeting with the client, after we had upgraded the project files to their most current version, she had mentioned the existence of a bug she was facing with the android version of the application. As she reported, after successfully entering the data for three tests the application would not allow you to return to the data entry page. Although after our meeting we had tried to replicate the bug, but we were unable to and all throughout the project we never ran into that issue. At the end we had concluded it might have been some incompatibilities of the phone running the outdated application, and since we rewrote most of the navigation code, we continued to test the updated navigation instead.

Results

The current working product is able to do all essential functions, data entry, calculations, plotting the prediction HAB graph, viewing the map and seeing the location markers, account creation, user login, password resetting, retrieving data from the database, and sending data to the database. We are currently working on optimising the database usage, because it is currently inefficient. Currently, it doesn't cause any problems, but if the scope of users were to grow, it could, so we are working on this as a preventative measure. Therefore, another SEII group that specialises in API development, is developing an API for us to optimise this process.

Related Works

Most applications devoted to HAB are not focused on encompassing all aspects of testing for HAB or predicting HAB. One similar application would be Bloomwatch, but it has some features missing. Bloomwatch also has the ability to qualitative analyze lakes, that is the only way to check for HABs. The major feature it lacks is the ability to do quantitative tests and it refers users to order kits. Another missing feature is the ability to view other users test results. Bloomwatch does have the ability for users to upload their tests, but they do not allow users to share their test results with each other.

Future Works

The client's primary goal was to make the application more user friendly, especially for those outside of the biology community so that the tool is more accessible. The changes made are a step in the right direction, but there are some other features that the client suggested that weren't within the scope of this semester.

Firstly, the client mentioned they had a vision for the application to have some sort of social media-like functionality where users could share data, comment on each other's posts and findings, and lessen the gap between the everyday user and the scientific community. As this was far out of scope of 1-2 semesters of work, Jose Martinez and Zaid Altahat suggested implementing a twitter hashtag, where the application could generate a url for the data, which could be tweeted with a hashtag. This would save a lot of development time, and delegate a place for discussion of the application's datasets, on an already developed and public platform, that is very widespread and user-friendly. The client agreed that this was a reasonable compromise.

Then, the client requested that users be notified when an algae bloom happens in their area. This range could be adjusted, and the user would be able to set notifications for certain bodies of water, or ranges from other points rather than the user's current location. This would be a significant quality of life improvement, and increase the use-cases for the app, but we have not yet had development time to implement this.

Conclusion

This application is an important tool for the public to be informed about the water safety of their local bodies of water. The ability for users to create accounts and store their experiments online, create experiments and graphs, qualitatively test for HABs, view other users submitted nearby experiments, view the experiments as a map with colored markers, and read useful information regarding algae and HABs. Development of this application will continue to add the features described in the previous section. An application to predict and track HABs will continue to be useful for everyone across the world.

References

[1] Ferris, Robert. “Why Are There so Many Toxic Algae Blooms This Year.” CNBC, CNBC, 26 July 2016, https://www.cnbc.com/2016/07/26/why-are-there-so-many-toxic-algae-blooms-this-year.html
[2] Spencer E. Fire et al. “Comparison of during-bloom and inter-bloom brevetoxin and saxitoxin concentrations in Indian River Lagoon bottlenose dolphins, 2002–2011”, Aquatic Toxicology, 31 August 2019. DOI: 10.1016/j.aquatox.2019.105371
[3] International Atomic Energy Agency. “Harmful algal blooms and associated biotoxins.” IAEA, IAEA, 24 Feb. 2020, https://www.iaea.org/topics/harmful-algal-blooms
[4] Hughes, Terry Szmant, Alina M. Steneck, Robert Carpenter, Robert Miller, Steven, (1999), Algal blooms on coral reefs: What are the causes?, Limnology and Oceanography, 6, doi: 10.4319/lo.1999.44.6.1583.
[5] Goreau, T. J. 1992. Bleaching and reef community change in Jamaica: 1951–1991. Am. Zool. 32: 683–695.
[6] The AlgaeEsitmator was originally created by University of Wisconsin - Parkside students Harleen Kaur, Hanh Le, and Francisco Mateo