SpaceMonitor

PRDv2

Team: 2B || !2B

Company Name: PowWow Energy

Authors:

- Daniel Shu (danielshu@ucsb.edu)
- Richard Rodriguez (richardmrodriguez@ucsb.edu)
- Anna Ivannikova (anna03@ucsb.edu)
- Yiyang (Meredith) Xu (yiyangxu@ucsb.edu)
- Bryan Wu (bryanwu@ucsb.edu)

Introduction

Farms are trying to make the most efficient use of their resources, but there could be room for improvement. Inefficiency not only wastes resources and costs money but also contributes to climate change. Therefore, a cheap and accessible method for evaluating the efficiency of water usage should be provided to farms in order to help them identify and execute on areas in their crop fields that excessively use water. Our project will provide a public web tool that shows problematic areas where water is not used efficiently. Members of the public such as farmers, researchers, and Resource Conservation Districts (RCD) will be able to use this information to help conserve water, minimize the effects of droughts and water shortages, and reduce the cost of agriculture.

There currently does not exist any solution that uses geospatial remote sensing data to find areas where water use across crop fields is not uniform and where water use is relatively high for the specific crop. Satellite data is much cheaper to obtain than other sources such as high resolution drone imagery, making our solution more cost effective. With this in mind, our greatest innovation takes shape by leveraging satellite data and machine learning techniques to develop our web tool. Our model will look for areas of water use that are either uneven, indicating a farmer is not uniformly watering the field, or relatively high compared to other similar conditions, possibly indicating overwatering.

SpaceMonitor will bring together and analyze different data sets such as ETa (evapotranspiration) data, crop census from DWR, groundwater depth, and elevation data. The ETa and crop census data comes from satellite imagery that is provided by PowWow. The elevation data comes from publicly available sources, and the groundwater depth data comes from monitoring equipment within wells. Using unsupervised machine learning, our algorithm will predict if a certain area is water-efficient or not. Our algorithm will attempt to cluster together fields that have similar water needs and assign an efficiency score to each one relative to other crops within its cluster. We also hope to present the results of our algorithm to the user in an easy, readable way not only so that they may understand the results, but also so they can give feedback. Any users will be able to see inefficient areas since they will be highlighted with a different color. They could also click on any highlighted area, and a pop up window will output ML statistics about this particular farm. In order to make more efficient use of our server's computational power, we will periodically cache the results produced by our algorithm. This feature only works under the assumption that the field data can be accurately processed, loaded, and displayed on a user's device with trivial delay.

One of the main challenges we encountered during our work with the project occurs in the data. The data is very noisy in that two farms growing the exact same crop in relatively the same location could have completely different water needs. Additionally, each data layer may not be equally important in determining the efficiency of a farm. We will have to determine which data layers should be given more importance, and this could also differ based on the geographical location.

Team Goals

Our initial goal is to create a minimum viable product that can output the machine learning analysis on the maps, highlighting inefficient areas. Users will be able to visualize the different levels of the output and hide the layers in which they are not interested in. They will also be able to locate themselves and find the closest farm since the map will have a search box. Our secondary goal is to create a private cabinet for users which will remember the areas which a specific user wants to be monitored over time.

Our plan for sprint one was to create the foundations of the frontend, backend, and database. We began researching some of the technologies and learned how to use some of the tools such as React and Django.

For sprint two, we will create stubs for the outline of our classes and begin to plan how they will all interact. At this point, the complete end-to-end functionality should be stubbed out.

For sprint three, we implemented the full pipeline. Our website grubs data from the backend (using API call), draws polygons of the farms on Google map, applies clustering machine learning algorithm to analyze which areas have not being efficient, and highlights inefficient areas with different color.

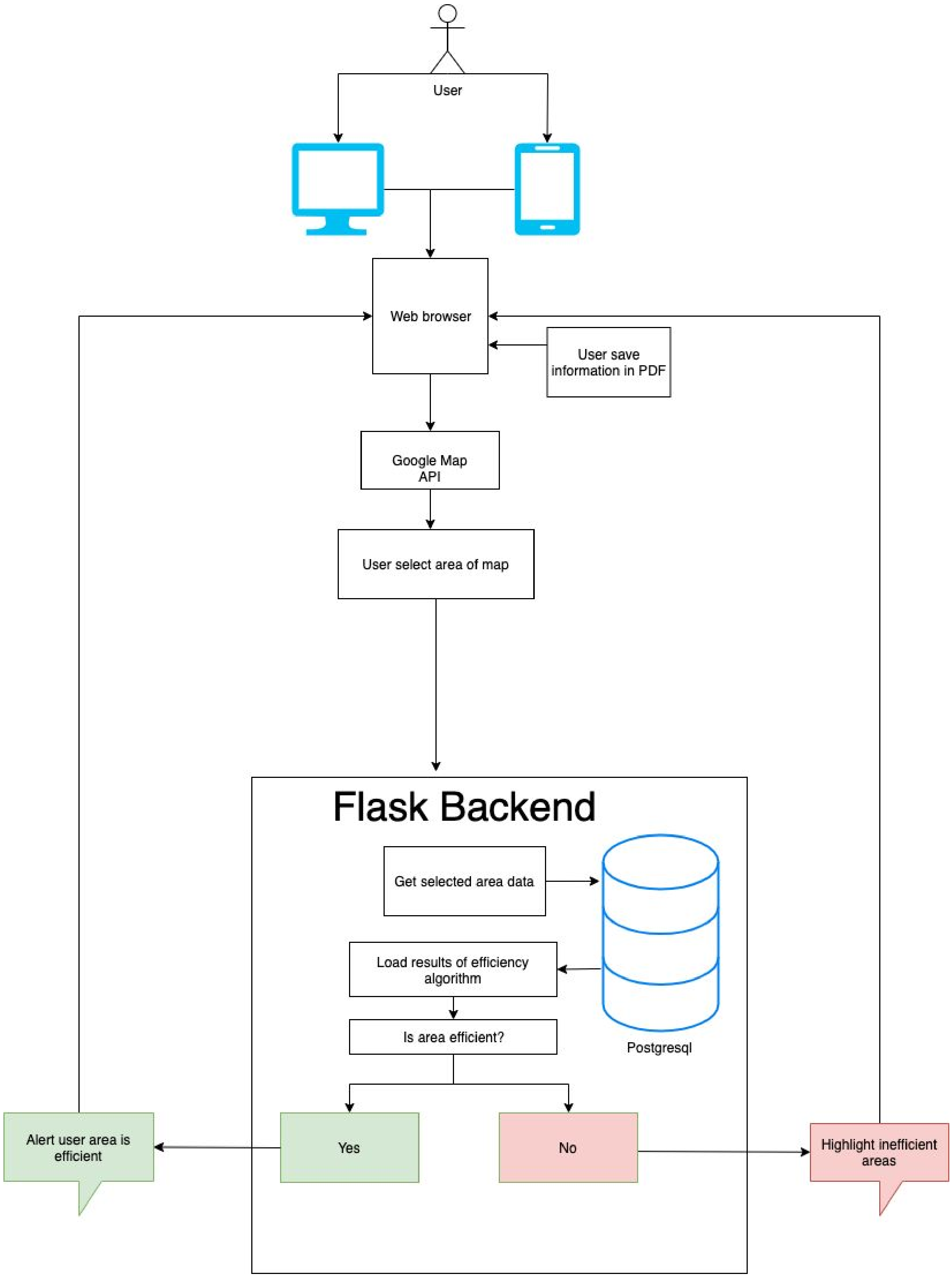
For sprint four, we will focus on converting Google maps to Mapbox, writing different unit tests, improving clustering algorithm and storing the results into the database, finding the way to store API key into .env file, deploying the website to Heroku, and preparing our minimum viable product for our presentation.

Our team objective is to foster a work hard, play hard culture where each team member is interested in the project and will be proud of their contributions. The time our team spends together during the class meetings and daily scrums will always serve as an environment for personal growth where we are constantly learning and applying both technical and nontechnical knowledge like the use of our tech stack and team communication practices respectively. Moreover, we hope that the fruits of our labor will help inform and motivate the public on methods to improve inefficient water usage in the agriculture sector.

Assumptions

- The publically available datasets will never be privatized.
- Users will always be able to see the ML results on the map
- Data used is accurate and precise
- Users will accept the website being publicly available
- Our target audience are farmers in Fresno County and Kern County (due to the limited data sets), researchers, Resource Conservation Districts (RCD), and anyone who’s interested in agricultural data
- The UI is intuitive and provides easily understandable processed data

High Level Diagram



UI Design

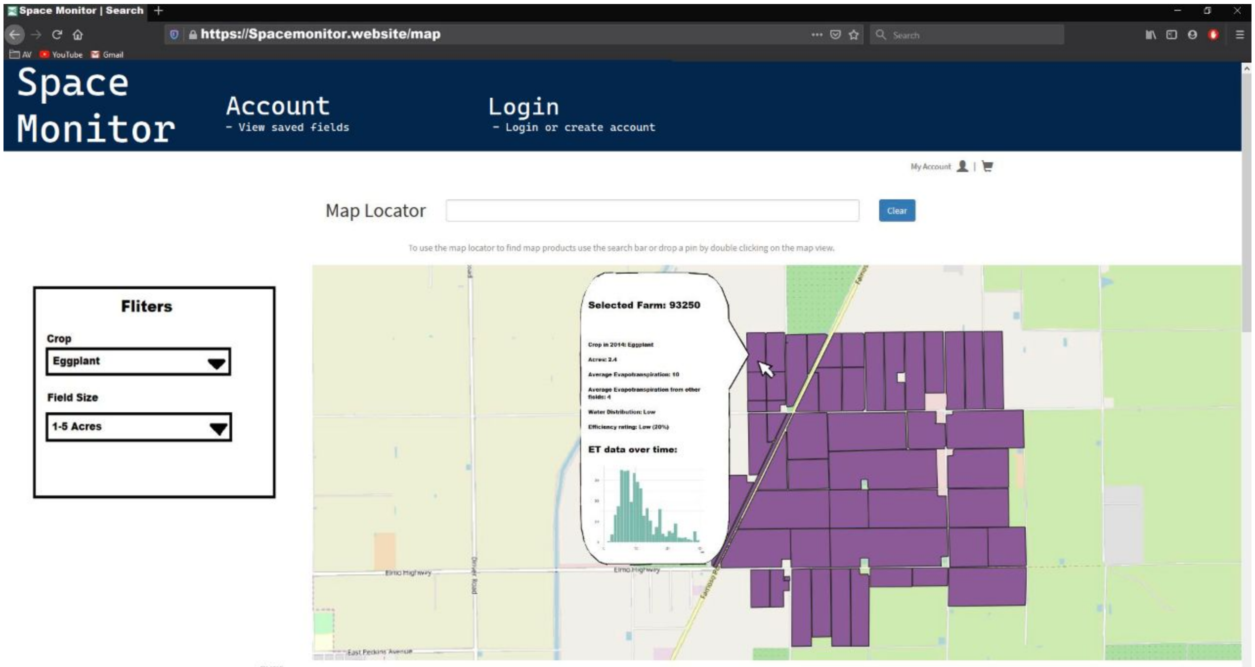


Figure 1. Main page seen when opening website. Users can click on a specific farm and view certain statistics on that selected farm, filters can be applied to only show certain fields

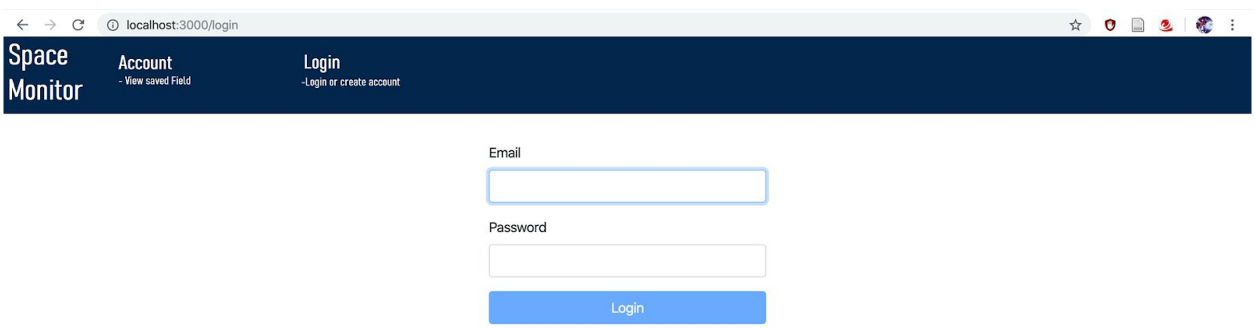


Figure 2. A login page for users to access their account and view saved fields/farms

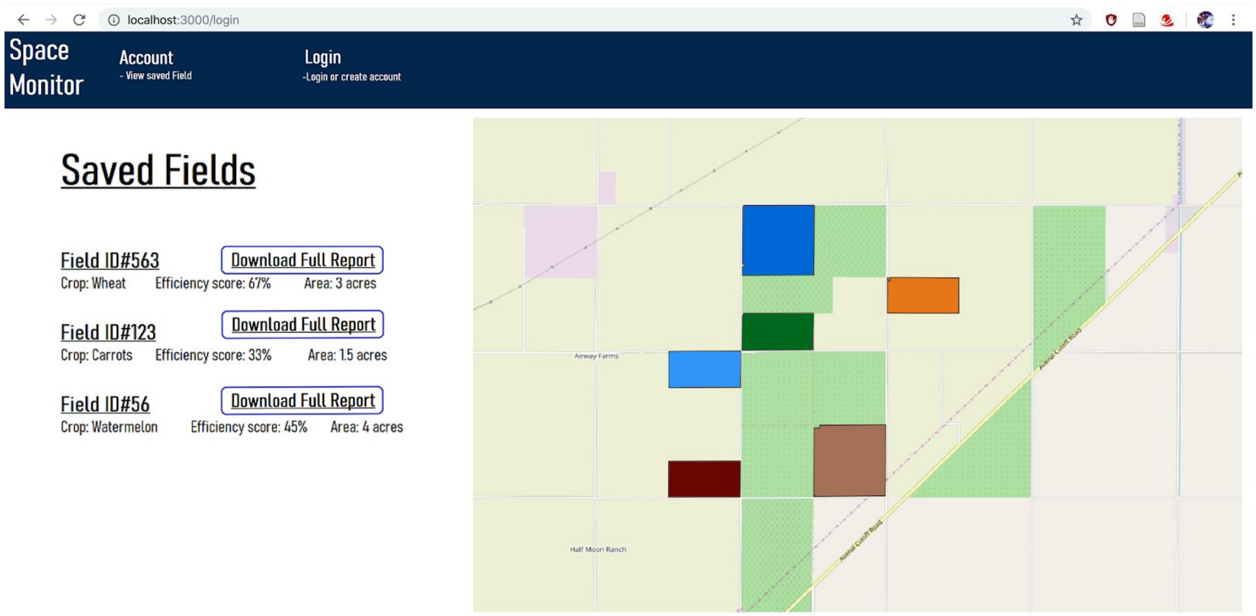


Figure 3. Users can see fields that they have saved for quick lookup. Users can also download a pdf report on a specific field.

Functional & Non-Functional Requirements

User Stories:

1. As a user, I can browse and zoom in on different portions of the map so that I can view information only in areas I am interested in. (~2 days)

    Acceptance Criteria: The map is correctly displayed on the home page and the user can zoom in, zoom out, and click and drag the map to move the view.

    Trello:
    https://trello.com/c/rdSBuJ6f/10-user-story-1-user-can-zoom-in-and-browse-different-portions-of-the-map

    Commits:
    https://github.com/aivannik/powwowEnergyCapstone/commit/9a0ee0208dba49a3b48a6e0c15c35ea7af14b18d

2. As a developer, I can run a clustering algorithm on a specified area of the map so I can see which fields are inefficient relative to similar nearby fields. (~2 days)

    Acceptance Criteria: After user selects a portion of the map, the algorithm will run and highlight areas where water use is relatively higher.

    Trello:
    https://trello.com/c/t4D2LJMC/12-user-story-2-run-clustering-algorithm-to-see-where-water-is-being-used-inefficiently

    Commits:
    https://github.com/aivannik/powwowEnergyCapstone/commit/ea44133638aaad2ee76fbb90011e7d416415c86c

3. As a user, I can download a PDF report of my farm for a specific year so that I can have a record of how efficient my farm is. (~1 day)

    Acceptance Criteria: Pressing a button will download a pdf document that lists all the results from the map.

    Trello:
    https://trello.com/c/3rhDcfeD/9-user-story-3-generate-pdf-report-of-water-inefficiency-for-a-specific-field

    Commits:
    https://github.com/aivannik/powwowEnergyCapstone/commit/7ff7fd136fc57a8811347d196225eb35d7d82378

4. As a user, I can filter the map by crop and see which areas are growing what crop. (~2 days)

Acceptance Criteria: The map can take a parameter to specify the crop filter then show data/areas with that crop

Trello:
https://trello.com/c/jCq71zKz/15-user-story-7-filter-which-crops-are-displayed-on-the-map

Commits:
https://github.com/aivannik/powwowEnergyCapstone/commit/28957b7248615c3ead0754d5451c95960cc5d8ba

5. As a user, I can choose which data layer(s) to display on the map (elevation, water depth, etc) so that I can filter the information and only see what I am looking for. (~ 2 days)

Acceptance Criteria: Pressing different buttons will display the data over the map as an overlay or remove it.

Trello:
https://trello.com/c/kr1Xc5LY/16-user-story-8-filter-which-data-will-be-displayed-on-the-map

Commits:
https://github.com/aivannik/powwowEnergyCapstone/commit/28957b7248615c3ead0754d5451c95960cc5d8ba

6. As a user, I can click on a certain highlighted field on the map and a popup window will appear so that I can see information specific to one field instead of the whole map. (~ 2 days)

Acceptance Criteria: A popup will be visible

Trello:
https://trello.com/c/qPnecPIC/19-user-story-11-select-a-field-on-the-map-and-a-popup-window-will-display-all-the-information-specific-to-it

7. As the frontend, I can plot the statistics about a particular field through line graphs or histograms on a popup window that displays this field’s water use and efficiency over time so that the data is displayed in a visual way. (~ 2 days)

Acceptance Criteria: A popup with information that displays the evapotranspiration data over time will appear for any field that is clicked.

Trello:
https://trello.com/c/UamEkoZ1/21-user-story-12-as-a-user-i-can-read-a-popup-window-that-displays-this-fields-water-use-and-efficiency-over-time

8. As the efficiency analyzer, I can take some criteria (weather, ETA) and give an efficiency score to each field so that I can label which fields below a threshold are considered inefficient. (~ 4 days)

Acceptance criteria: Every field will have some numerical value score and a threshold will be chosen.
Trello:
https://trello.com/c/hAJkBKVY/24-user-story-15-as-the-efficiency-analyzer-i-can-take-some-criteria-weather-eta-and-give-an-efficiency-score-to-each-field

9. As a user, I can search and display data by year for a particular area. (~2 days)

Acceptance Criteria: Efficiency score is based on ETa and other agricultural data, which differ year to year. Be able to display the results of the algorithm for any specified year.
Trello:
https://trello.com/c/36WAdBtn/17-user-story-9-display-data-for-a-specified-year-for-a-given-area

10. As an admin, I can input and change data into the database so I can rerun the algorithm and generate a new report for the area. (~ 1 day)

Acceptance Criteria: the users can see the updated data and download a new pdf
Trello:
https://trello.com/c/eFu0hFCd/18-user-story-10-the-admins-can-alter-data-to-generate-a-new-report-for-an-area
Commits:
https://github.com/aivannik/powwowEnergyCapstone/commit/bbfed8a1502fc4de2a5667f0ae6e7496c0f7a9c2

11. As a user, I can click on a field and see all other fields that have been clustered with it so that I know which other nearby fields are similar. (~ 1.5 days)

Acceptance Criteria: When a field is clicked, the user will be able to see the other clustered fields because they will all be colored.

Trello:

https://trello.com/c/k62AAddL/22-user-story-13-as-a-user-i-can-click-on-a-field-and-see-all-other-fields-that-have-been-clustered-with-it

Commits:

https://github.com/aivannik/powwowEnergyCapstone/commit/adceee06fa1b9e133cd9c8721b8499681ef6f7fc

12. As an efficiency analyzer, I can input data for any geographical area and be able to give an efficiency score so that the app is general and works on more than just California. (~1 day)

Acceptance Criteria: There are no hardcoded values and the algorithm will predict inefficient areas as long as the data is in the correct format.

Trello:

https://trello.com/c/kMy7wEYm/14-user-story-6-run-algorithm-on-a-specific-area-of-map

13. As a user, I can sign up and create an account so I can begin saving and personalizing the information from the app. (~ 2 days)

Acceptance Criteria: A username and password can be set up for a specific user, they can login to the website. Each registered user has personal data depending on their usage history

Trello:

https://trello.com/c/cAVhsoZD/11-user-story-4-users-can-create-an-account-to-personalize-and-save-information-from-the-app

Commits:

https://github.com/aivannik/powwowEnergyCapstone/commit/f147adf4eb4ff0221e820155e25f183812131a4a

14. As a user, I can log into my account using the username and password I created before so that I can see the information saved in my account (specific map area, etc) (~2 days)

Acceptance Criteria:

scenario 1: User with account

Given the user already has an account

When the user enters the correct username and password to login

Then it will take the user to a page where the user can see saved information in the account

Scenario 2: User without account

Given the user hasn't created an account yet

When the user tries to login

Then it will prompt the user to sign up first

scenario 3: User with account

Given the user already has an account

When the user enters incorrect username and/or password to login

Then it will prompt the user to try again

Trello:

https://trello.com/c/GxYRR2aH/20-user-story-5-i-can-log-into-my-account-using-the-username-and-password-i-created-before

Commits:

https://github.com/aivannik/powwowEnergyCapstone/commit/b2cc97ba7d53700d680240f7a803714b8d193765

15. As the backend web server, I can cache results produced by the algorithm so that the frontend does not take too long to compute and load the results. (~ 3 days)

Acceptance criteria: Be able to zoom in and render all the information in under a second.

Trello:

https://trello.com/c/jgpIQXHg/23-user-story-14-as-the-backend-web-server-i-can-cache-results-produced-by-the-algorithm

Issue:

https://github.com/aivannik/powwowEnergyCapstone/issues/30

Commits:

https://github.com/powwow-capstone/backend

16. As a server, I can precompute which fields are inefficient so that it saves time and the backend does not have to compute it every time it needs to service an API call. (~ 2 days)

Acceptance criteria: The inefficient fields are only computed once and sent to the frontend whenever it makes an api call.

Trello:

https://trello.com/c/z61LpEm3/25-user-story-16-as-a-server-i-can-precompute-which-fields-are-inefficient

17. As a developer, I can easily incorporate new data into my algorithm so that the framework is flexible and works for more than the default dataset. (~ 1 day)

Acceptance Criteria: The model design is flexible enough so that if new farm data is introduced, it will not dramatically change the algorithm.

Trello:
https://trello.com/c/EYLQxQIb/26-user-story-17-as-a-developer-i-can-easily-incorporate-new-data-into-my-algorithm

18. As a database developer, I can take geographical data and correctly associate it with the corresponding farm in the same location so that we can compute statistics for water use on a per-field basis. (~ 1 day)

Acceptance Criteria: When entering new data or ETa data from previous years, there is minimal or no change in the algorithm.

Trello:
https://trello.com/c/Y2Tovcnk/27-user-story-18-as-a-database-developer-i-can-take-geographical-data-and-correctly-associate-it-with-the-corresponding-farm-in-the

19. As a user, I can search a farm's address/city/zip code in order to easily find the farm on the map. (~ 3 days)

Acceptance Criteria: The map view will shift to the farm associated with the address user entered.

Trello:
https://trello.com/c/6UZnLLCV/28-user-story-19-as-a-user-i-can-search-a-farms-address-city-zip-code

20. As a developer, I can commit code changes and have them automatically deployed to the production website. (~ 2 days)

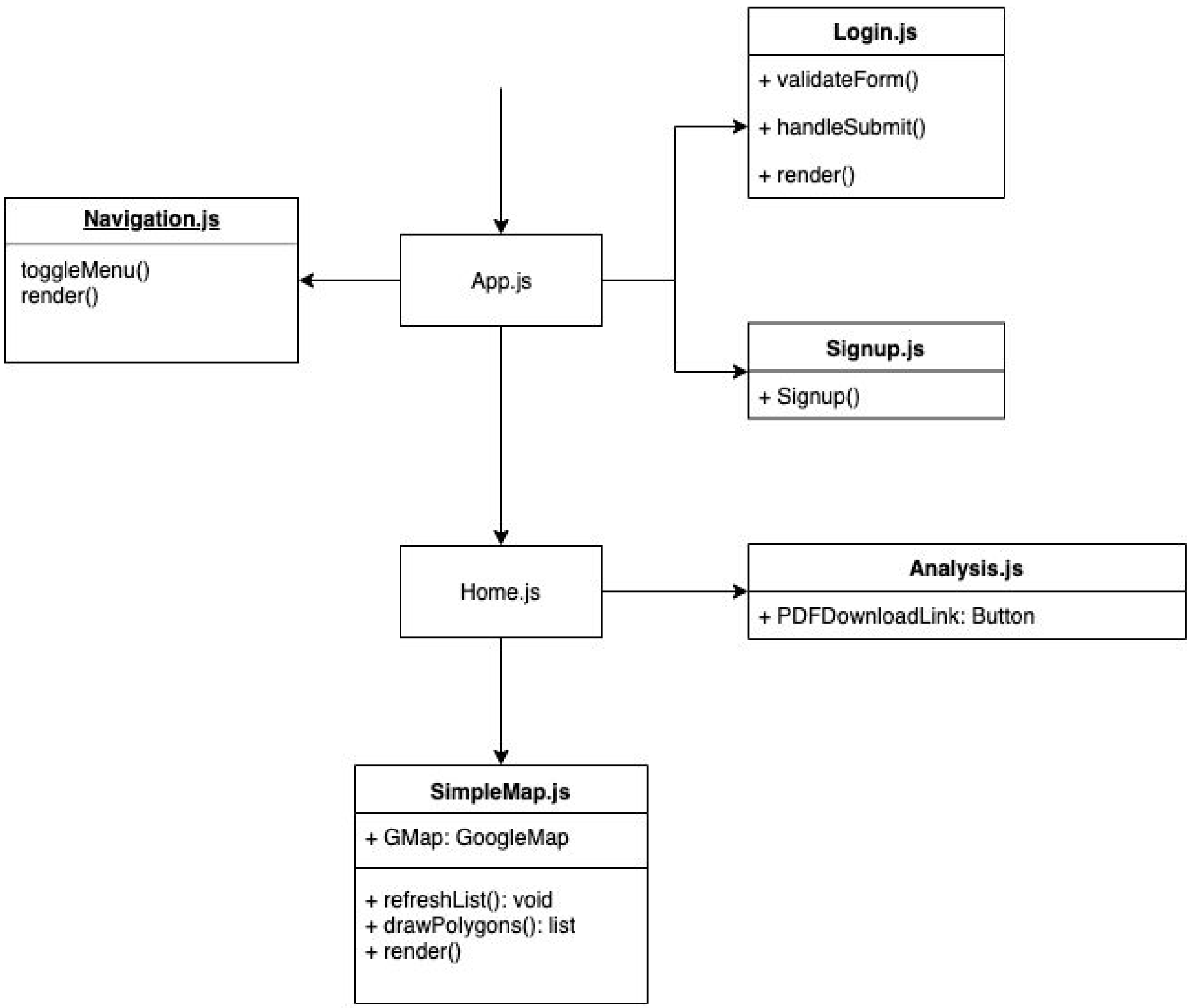
Acceptance Criteria: Github commits to the master branch are automatically deployed to Heroku production instance of website.

Trello:
https://trello.com/c/d87IMqYZ/29-user-story-20-as-a-developer-i-can-commit-code-changes-and-have-them-automatically-deployed-to-the-production-website

System Models

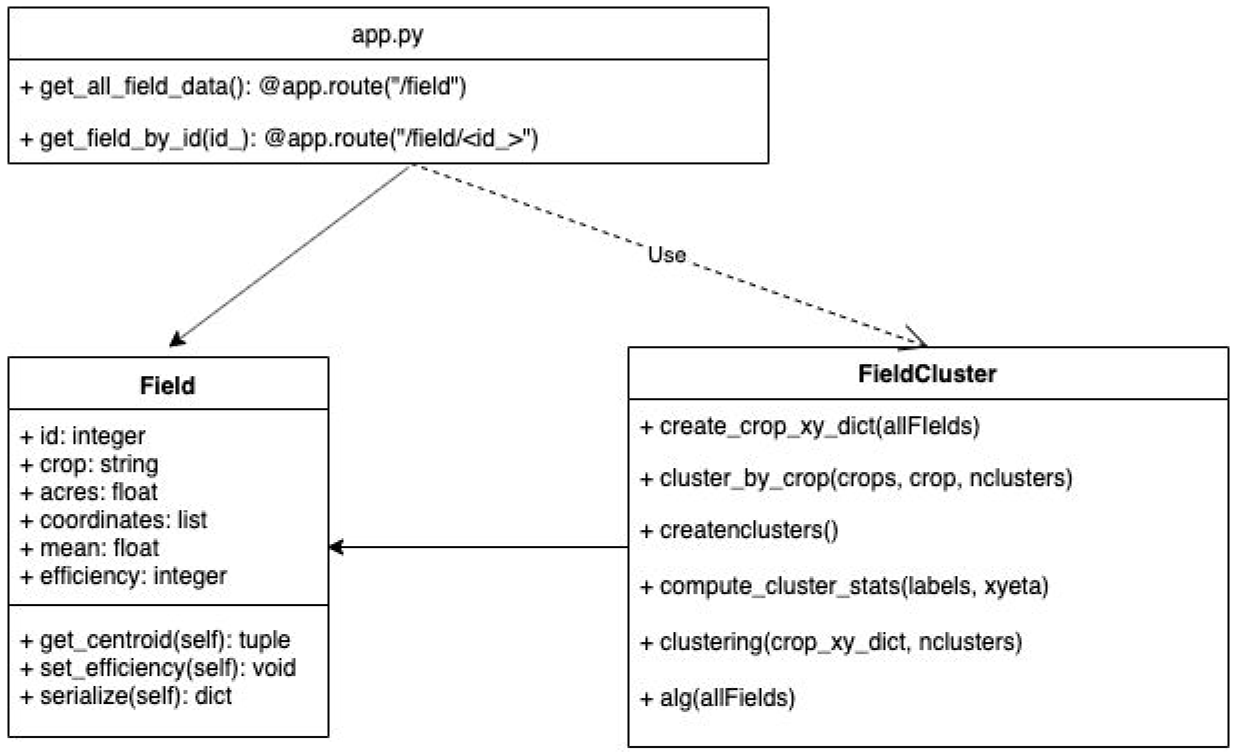
Frontend UML

The app enters the frontend through App.js. The main homepage is Home.js, which has two components: SimpleMap.js and Analysis.js.



Backend UML

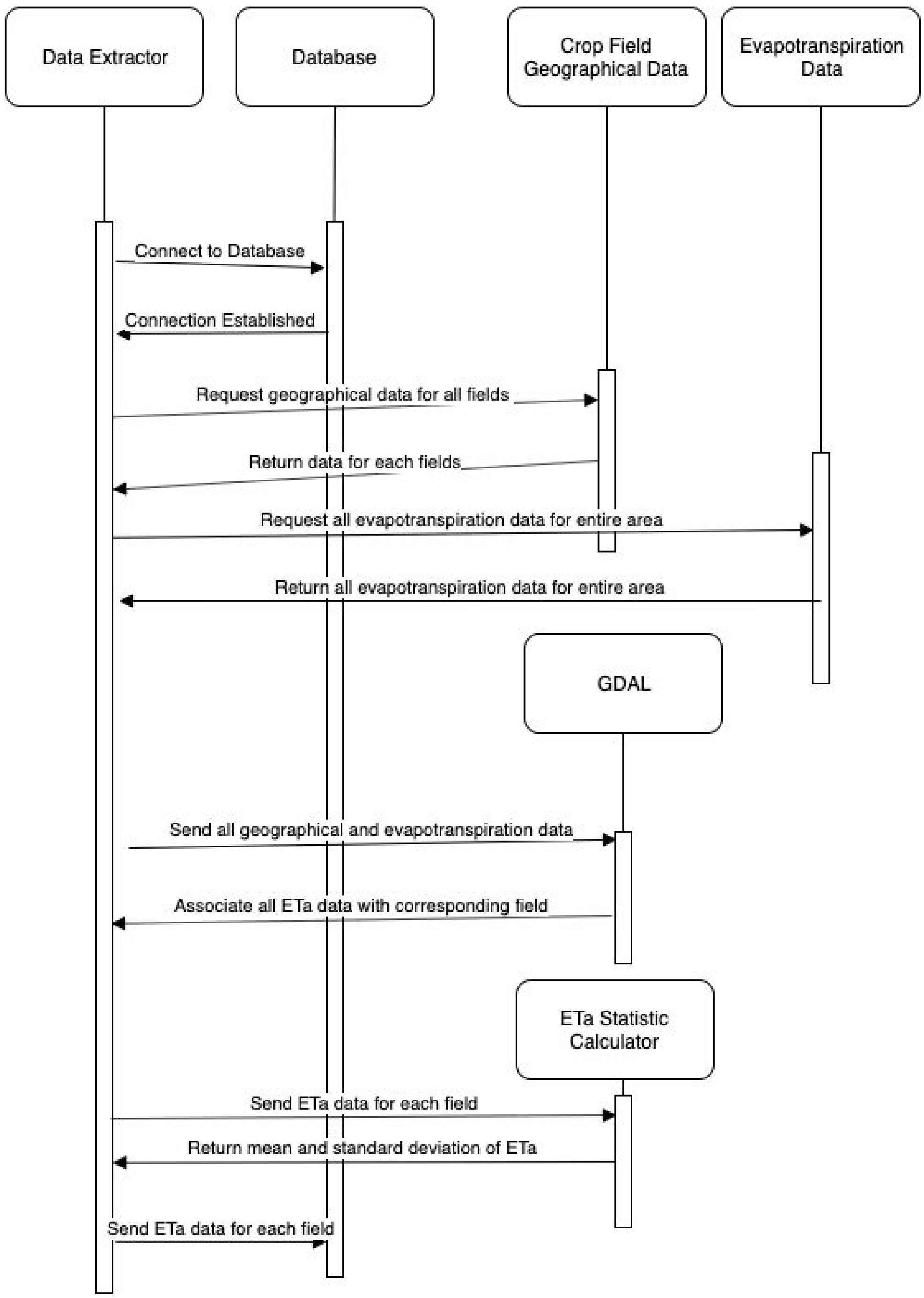
The backend uses Flask to interact with the database and service API requests from the frontend. app.py sets the API routes, and fields.py is a model for each crop field. A clustering algorithm is run on all the fields to determine inefficient areas.



Module Sequence Diagrams

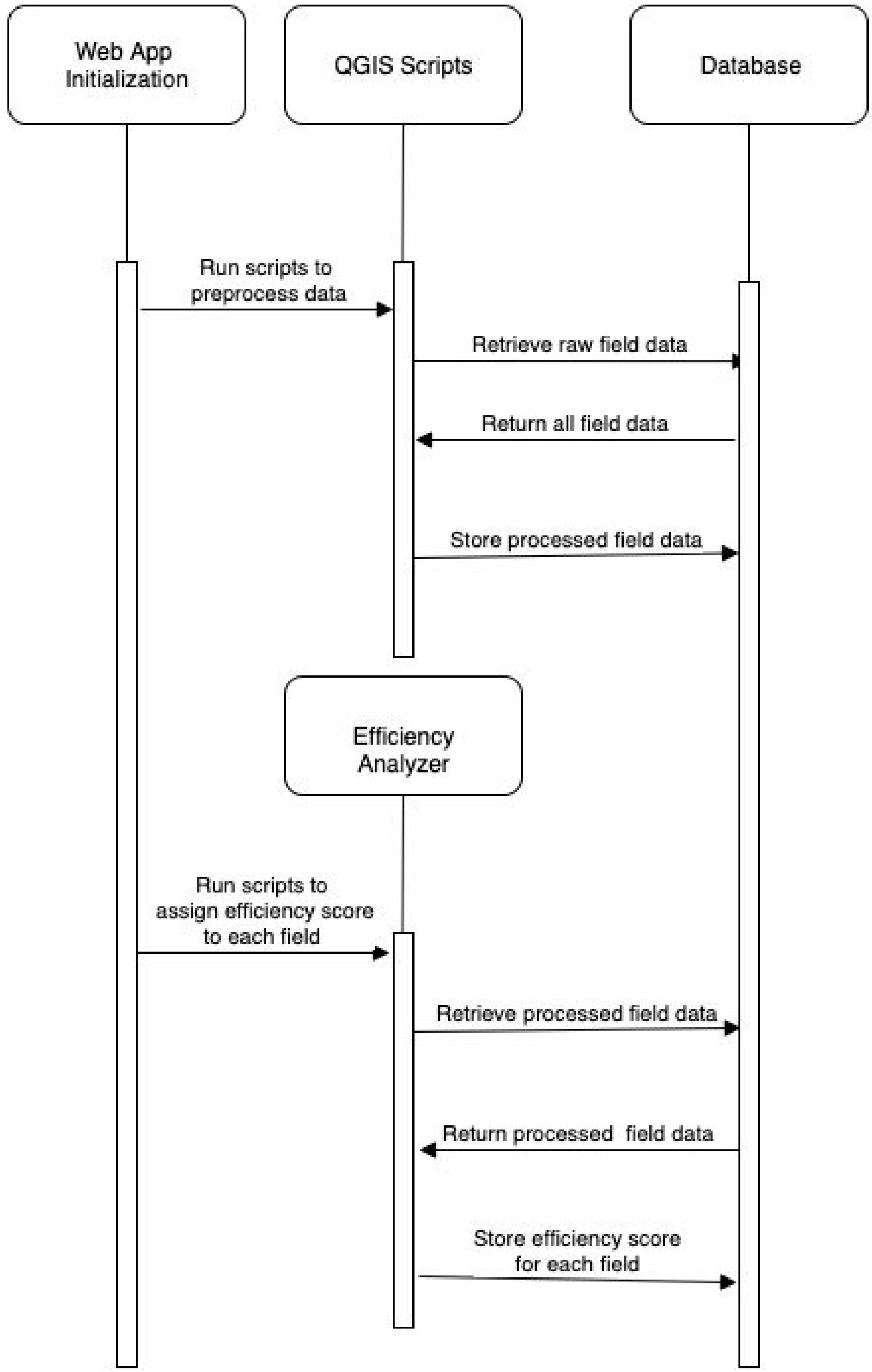
Extract data and store in database

The data for each field is spread over multiple files. In order to consolidate all the data and be able to associate the evapotranspiration (ETa) data to the correct geographical location, we must use GDAL, an external library that can parse the files, and store them in the database.



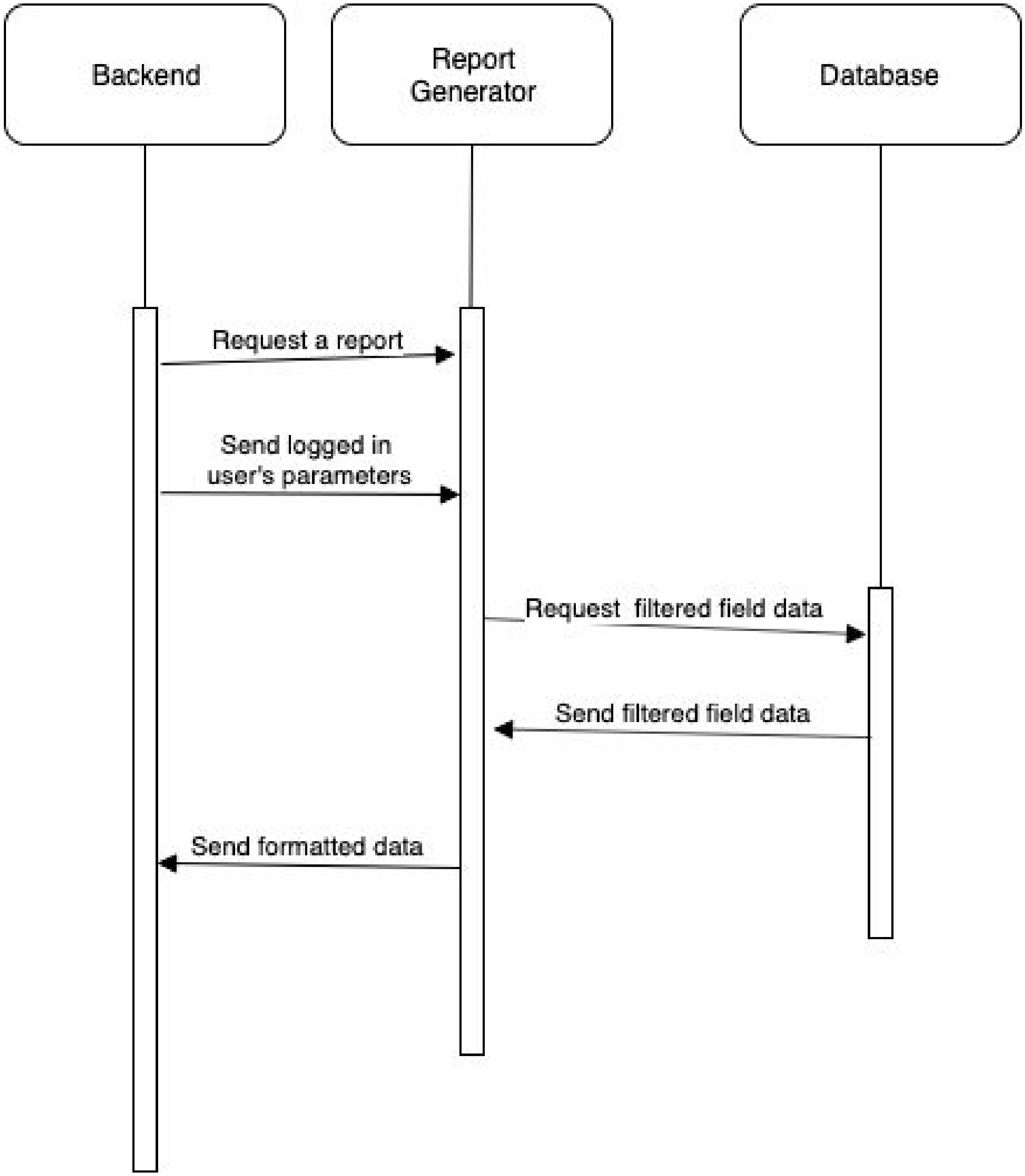
Initialize Web App

First, process all data and store each field within the database. All the field data is static, so upon initializing the web app, run the efficiency algorithm to determine which fields are efficient and which fields are inefficient. Store these results within the database.



Generate a Report

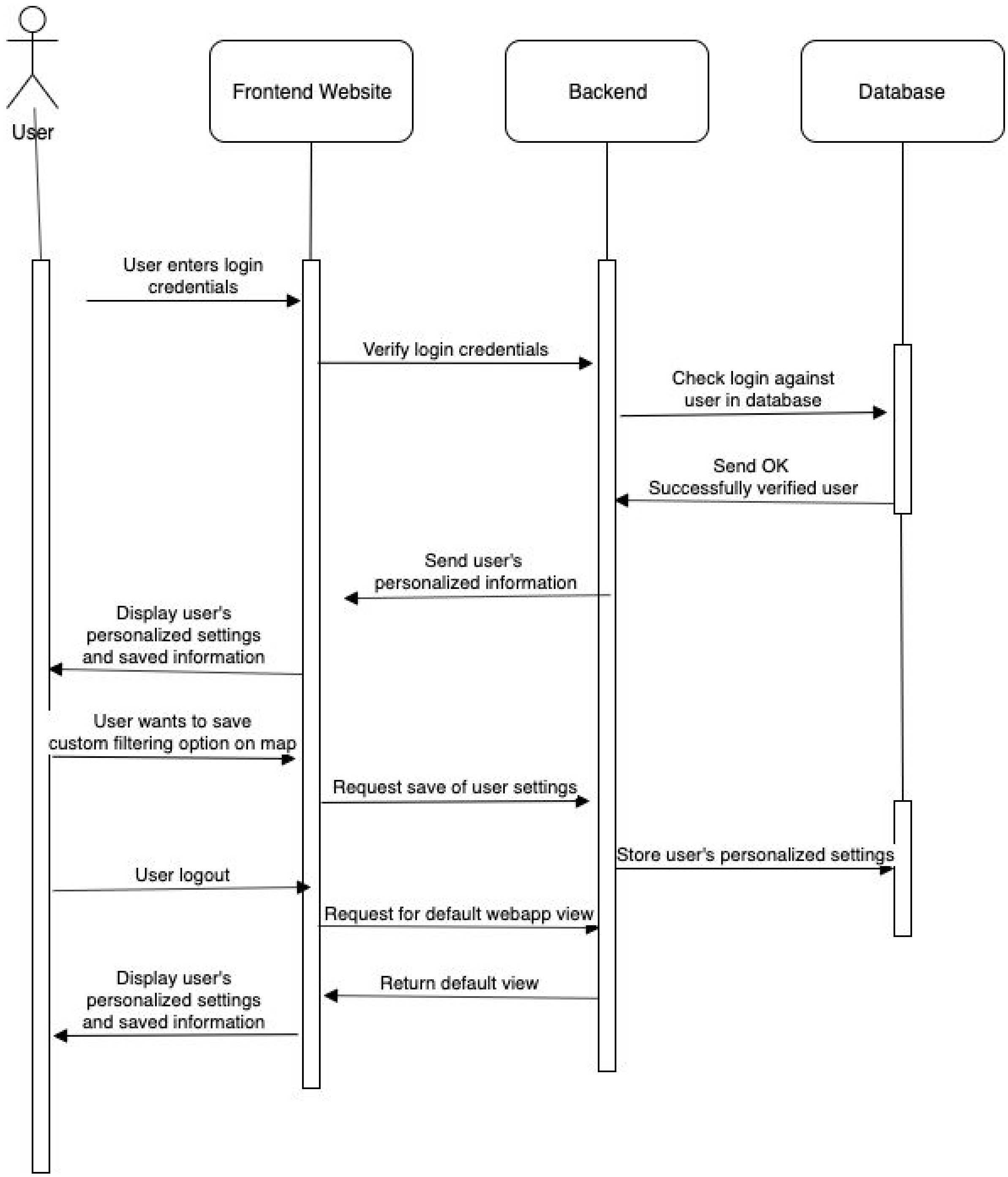
A logged in user will have filter settings and parameters for the map. These will be used to generate a report, where the efficiency depends on what filters were applied. The report is then made with statistics and graphs based on these settings.



User Sequence Diagrams

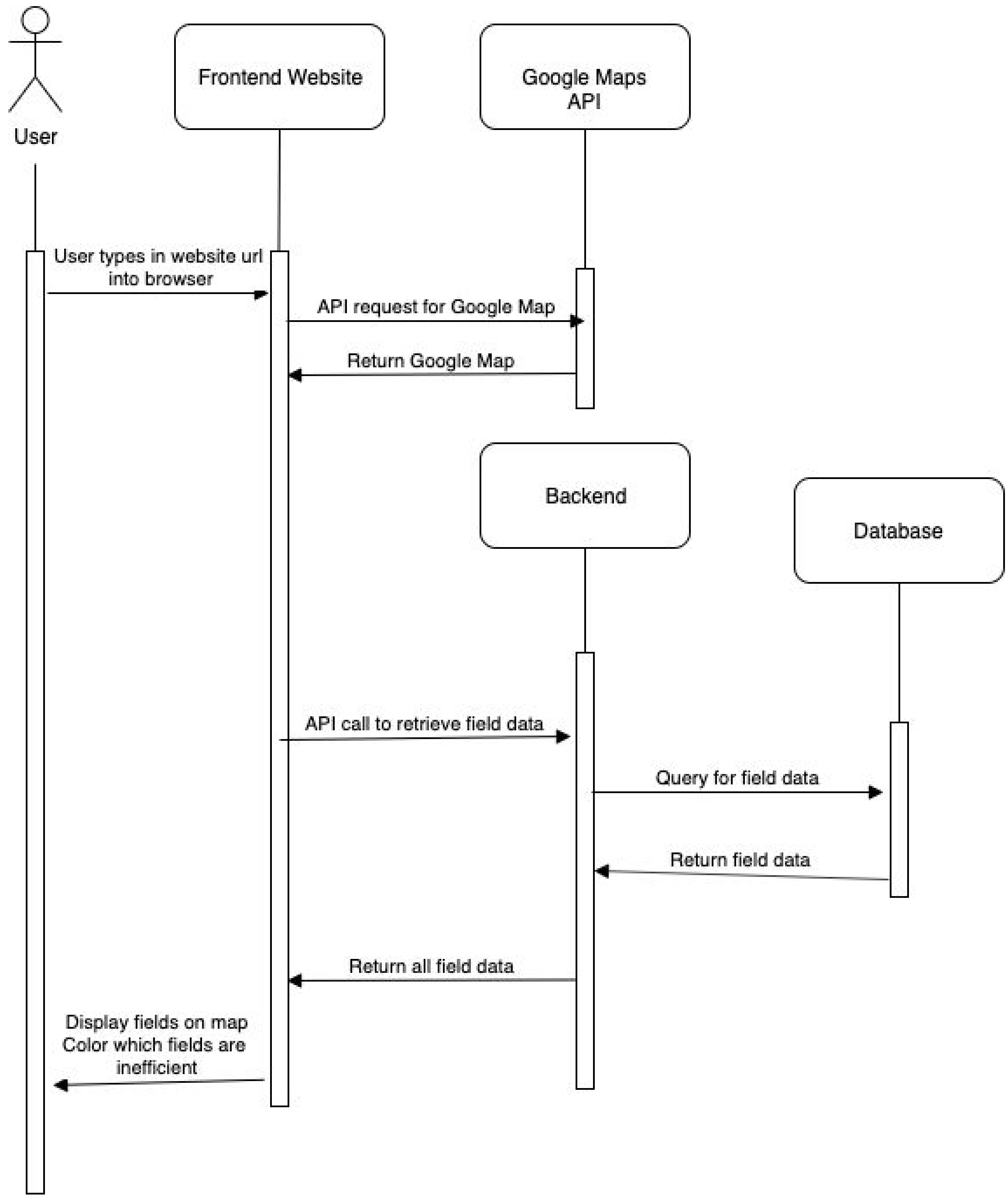
User Login to account

When the user logs in to his/her account, any personalized filtering or view options of the map are loaded and displayed. If the user wants to save a new filtering, it will be stored in the database. When the user logs out, the default view is restored.



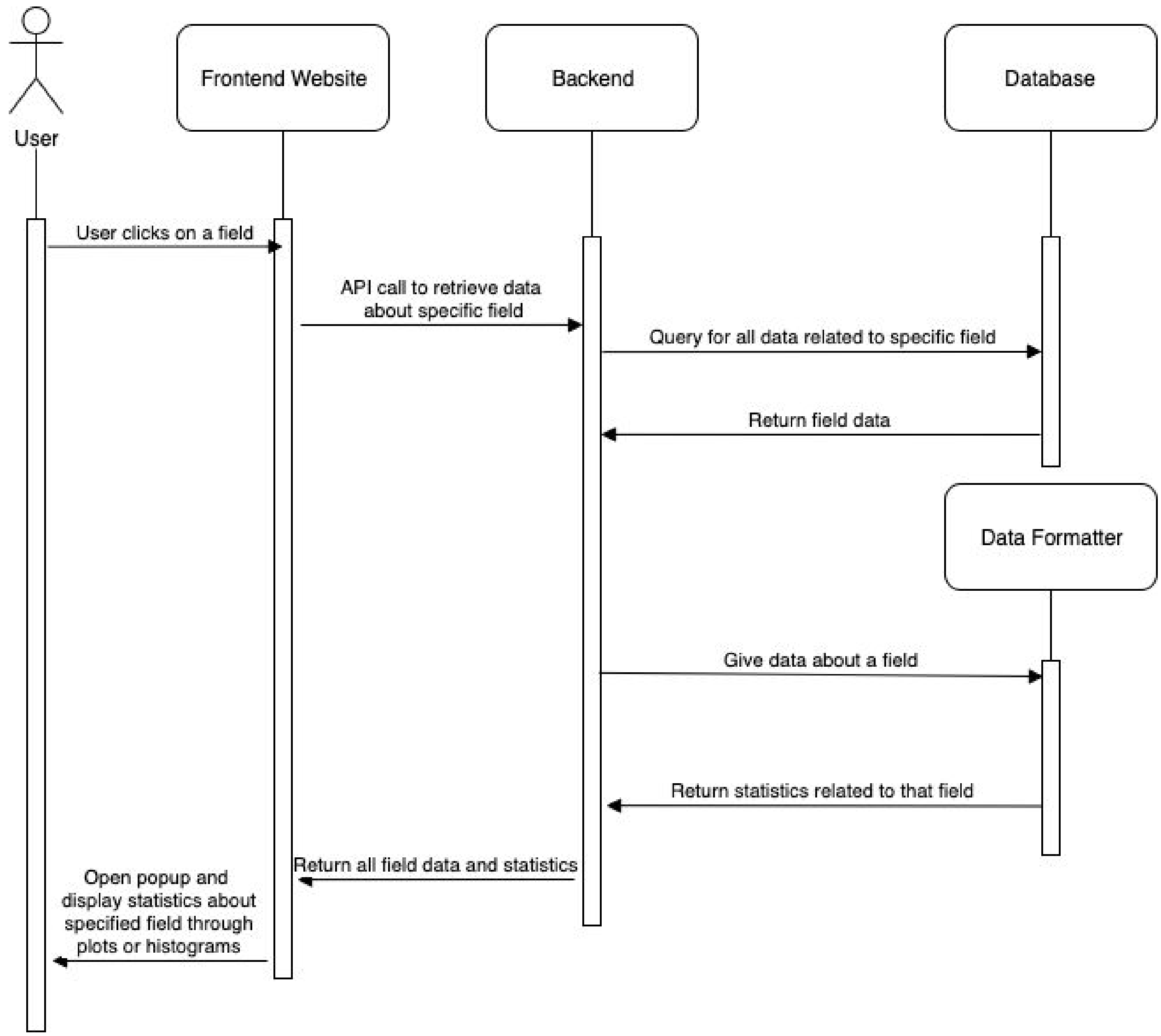
User access web app

When the user views the website, the efficiency algorithm has already been run upon initialization. Backend will load this data and send it to the frontend.



User clicks on a field

When the user clicks on a field, display all statistics related to that particular field.



Appendix

Technologies employed:

- Pandas
- Python
- QGIS
- Flask
- React
- OpenCV
- Google Maps API