PRD Version One

Project Title: SpaceMonitor

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 the 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.

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. A user will be able to visualize the different levels of the output and hide the layers in which they are not interested in. 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 will implement the stubs and have finished the primary use case, which is to identify what areas are being inefficient in water use.

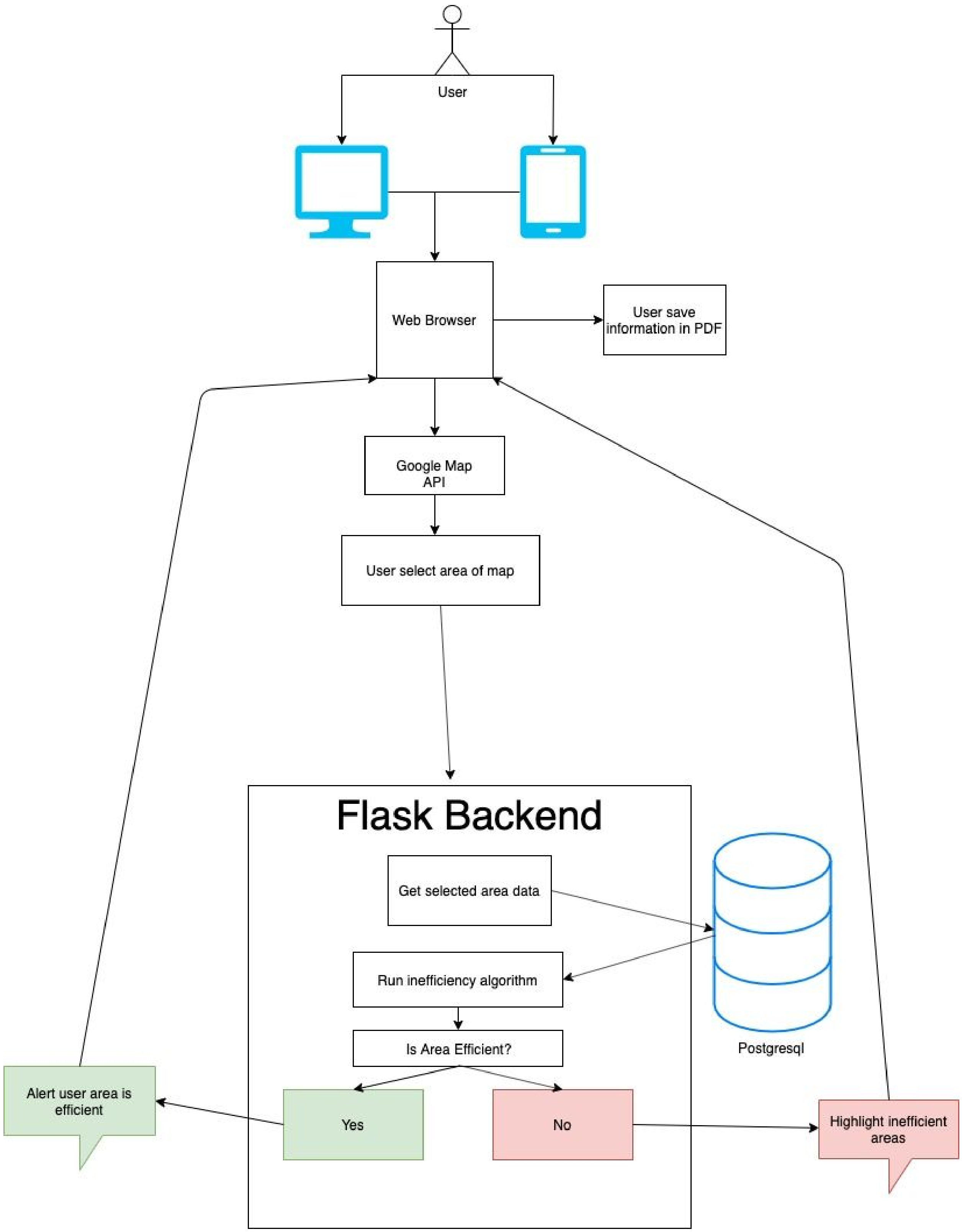
For sprint four, we will focus on finishing other features and preparing our minimum viable product.

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 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.
- Data used is accurate and precise
- Users can always login or signup
- Users with an account can see their saved areas on the map
- All users can access our map/website

High Level Diagram



Functional & Non-Functional Requirements

User Stories:

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

    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.

2. As a user, I can run a clustering algorithm on a specified area of the map so I can see where water is used inefficiently.

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

3. As a user, I can generate a PDF report of my farm for a specific year so that I can have a record of how efficient my farm is.

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

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

4. As a user, I can sign up and create an account so I can begin saving and personalizing the information from the app.

    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

5. 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)

    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

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

6. As a user, I can view a map of a specific area so that I will see what I am concerned about.

   Acceptance Criteria: The map can zoom in and focus on a specific area instead of showing everything

7. As a user, I can filter the map by crop and see which areas are growing what crop.

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

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

8. 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.

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

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

9. As a user, I can search and display data by year for a particular area.

   Acceptance Criteria: when a user searches for a particular year, he can see the data only for this year

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.

Acceptance Criteria: the users can see the updated data and download a new pdf

https://github.com/aivannik/powwowEnergyCapstone/commit/bbfed8a1502fc4de2a5667f0ae6e7496c0f7a9c2

11. As a user, I can select a field on the map and a popup window will display this field's water use and efficiency over time, so that I can see information specific to one field instead of the whole map.

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

Appendix

Technologies employed:

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