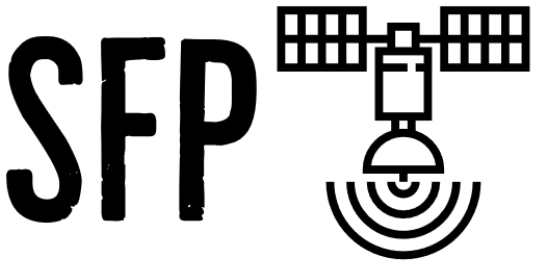
Team Satellite Fire Patrol

Software Design Document v1.1
February 16, 2024



Team Members:
Aaron Santiago, Ricardo Chairez and Zachary Hallemeyer

Clients:
Benjamin Wiebe, Dr. Camille Gaillard, and Dr. Christopher Doughty

Mentor:
Saisri Muttineni

Introduction

The Hawaiian islands hold immense ecological and cultural value but are currently threatened by increasing hot and dry conditions caused by climate change. This threat was evident in August 2023 when multiple wildfires across Maui resulted in at least 106 deaths, billions of dollars in damages, and ecological destruction. These fires begin in grasslands, which are vulnerable to high temperatures due to increasing drought and low moisture. Other Hawaiian ecosystems such as coral reefs and tropical forest canopies are also threatened by rising temperatures resulting from climate change.

Issues with Current System

- Lack of warning system for areas sensitive to excessive heat.
- Ecosystems are dying due to high temperatures
- No mitigation system for major loss of life or infrastructure

With climate change being such a major threat to the Hawaiian islands, we look to high-resolution thermal remote sensing as a way to monitor ecological threats. It has the potential of alerting researchers and natural resource managers when temperatures are approaching dangerous levels, so they can help mitigate potential damages. A thermal infrared image of Maui from NASA's Landsat sensor on August 7th, 2023 showed grassland temperatures ranking in the top 4% of the last decade. The information to detect a potential ecological disaster was there, but failed to reach the people it needed to.

Team Satellite Fire Patrol is partnering with clients Dr. Christopher Doughty, Dr. Camille Gaillard, and PhD student Benjamin Wiebe to create a GUI web application that uses real-time satellite thermal data to identify warning signs in the Hawaiian islands and send alerts to the proper authorities. The app will automatically aggregate and process satellite thermal data from multiple sources, compare historical averages to highlight temperature anomalies, and present the data in a user-friendly interface for a range of resource management applications.

Our Essential Features

- Retrieve most recent satellite data from NASA Ecostress
- Display most recent and historical temperatures of Hawaii
- Alert users if temperatures exceed a custom threshold

In our development, we must consider the technological challenges we face and what technologies we will use to solve them. In this document, we will explore the design of the web application that allows for easier environmental monitoring in Hawaii.

Implementation Overview

Overview

For the implementation, the project will be divided into three main components: a website application, a cloud hosted server, and a database. The web application will be made with a framework called Vue.js.

This web application is where the land surface temperatures can be seen on a virtual map. The virtual will be in a satellite style, allowing the user to move and focus on areas of Hawaii that interest them. The virtual map will be rendered with the help of an API called Mapbox. Along with this, the user will be able to create custom alerts in the web application, with both the user information and alerts being stored in our database.

The database will be contained on NAU's monsoon servers. The monsoon servers have 5 terabytes available for the web application to use. The information that needs to be stored is temperature data, both historical and current, as well as information about users and their custom alerts.

Finally, a backend program hosted on the monsoon servers will be responsible for gathering land surface temperature measurements from NASA Ecostress. Once new data is gathered, it will be processed and stored in monsoon servers. As the new information is stored, the web application will be able to request the information to display

Technologies

The team has researched different technologies and frameworks available to use in implementation and has decided on the following:

- **Django** - Django is a robust web development framework that is designed for rapid development. As the backend of the web application will be handling large amounts of data processing and sending, a framework that is optimized for large data was needed. Django has powerful object relational mapping which allows for simpler and more importantly, quicker, data interactions. Another factor in using Django, is that Django was built and maintained with security in mind. The web application will be storing limited information about users, so security is vital for the project.

- **Vue.js -** Vue.js is a front-end framework that allows for flexible and rapid development of the web application's front end. It has the capability to effectively render data into the virtual DOM
- **Mapbox** - Mapbox is an API developed for web applications. The API allows for custom rendering of virtual maps, making it ideal for this project.

Architectural Overview

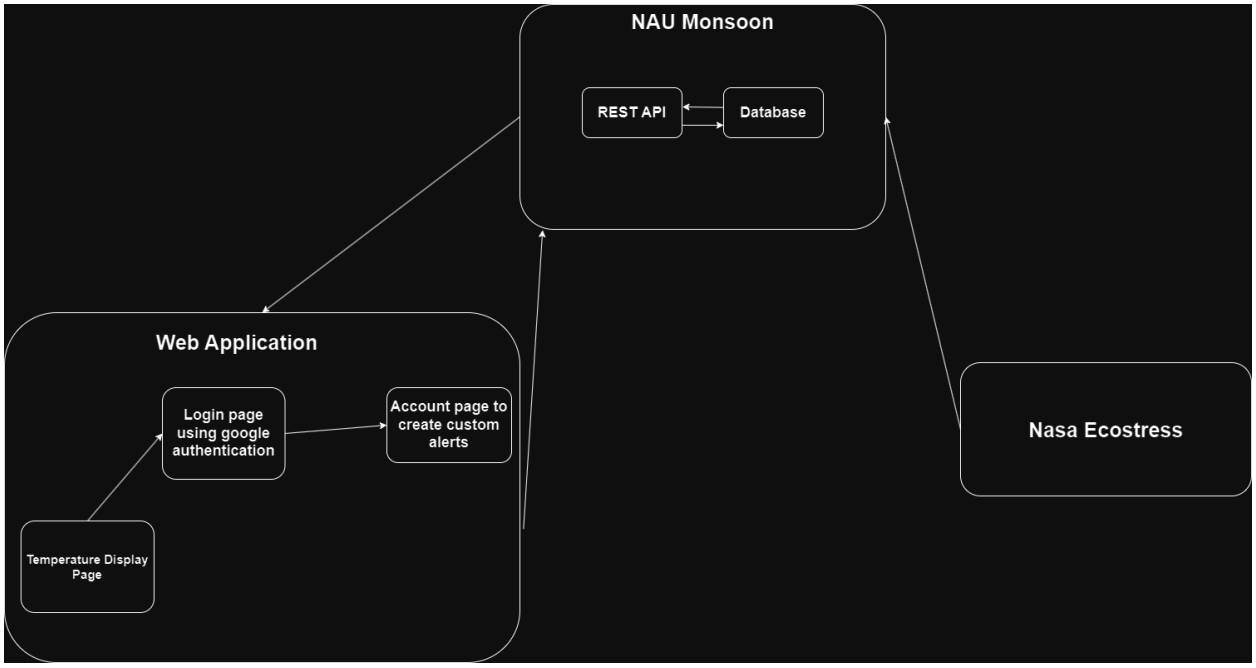


Figure 1: Architecture Diagram

Our system will be composed of about two separate parts that must communicate with each other as efficiently as possible to provide the best possible user experience. The first component of our system is the back end, which will be hosted on NAU's monsoon and have these crucial tasks. First it must prompt and store the data from NASA Eco stress, then it organizes the data and places relevant information into folders based on years. It will also hold our backend framework Django, which will communicate with our front end whenever a user requests data on the map to send the correct data for those coordinates. The data will be requested and sent out of the RD port on NAU's monsoon.

Then there will be the main website, which allows users to login using their Google credentials and save up to 5 areas to be monitored. The main website will communicate with the backend and be created using Vue.js as a framework. The front end will be communicating with the RD port to show new data on the website as soon as it is requested. Users will be able to save 5 locations using a select tool, which will be saved with their account. The front end will update users if their thresholds are reached or if their areas have been updated through SMS or email using OneSignal.

NASA Ecostress API will be providing up-to-date information consistently every day to the monsoon database. Setting up our system this way allows for us to organize the raw data from NASA as soon as the satellite completes an orbit. The monsoon database will organize and

calculate the averages and push its data to the data portal for the website to grab and use for the maps.

The web application will be hosted on ThermalWatch.org and will contain multiple pages and be responsible for the overall user experience. Signing in with Google accounts will enable users to save and monitor 5 locations with a selection tool. Mapbox is similar to Google Earth in both its user interface and information. This tool allows users to see an accurate satellite view/or graphical representation of Hawaii.

Mapbox also allows for graphical overlays on its maps, which we will use to display the temperature data that has been published from monsoon. The design will be easy to follow and use, which will make it as accessible as possible to the public. The website will also be responsible for using the OneSignal technology to send notifications to users when their temperature thresholds have been reached or surpassed. The notifications will send the user a notification with the information about which zone was breached and a link to access the website.

The database management system we will be using is called Django, which is a backend framework which is efficient at handling large amounts of data. This system will be stored in Monsoon to handle the large amounts of data we will be receiving and organizing our 4 years of historical data. With this system, we can easily organize and manage the data to complete necessary calculations such as averages and cloud cancellation efficiently.

Module and Interface Descriptions

Database - NAU Monsoon

Data Retrieval - The land surface temperature data will be retrieved from NASA Ecostress directly. In order to accomplish this, NASA's API, earth data, will be used. When the web application is first run, historical data from the last five years will be retrieved. After this initial task, the program will request new data every hour.

Data Processing - The data retrieved from NASA's API is in h5 format which needs to be converted to geojson in which each latitude and longitude point has a temperature. The data must also be processed to remove clouds as well as outliers. For this, quality control masks will be utilized and outliers will be discovered with the use of historical data. Furthermore, the metadata including date, bounding box, and such must be created.

Data Storage - With our well-thought-out framework, the data will be stored in our 2 terabytes of storage on our Monsoon. In our data processing, the data will be organized into separate folders based on the date that the data was retrieved. This clearly separates the relevant information into neatly organized folders in our storage, which aids us in our data requesting and calculations.

Custom Alerts - As new data is successfully processed and stored, the custom alerts from users will be checked with the new data to see if the new temperatures exceed the custom threshold. If the custom thresholds are exceeded, then the user will be notified along with the coordinates of the exceeding temperatures. This will be accomplished using OneSignal, which sends out notifications via SMS and email.

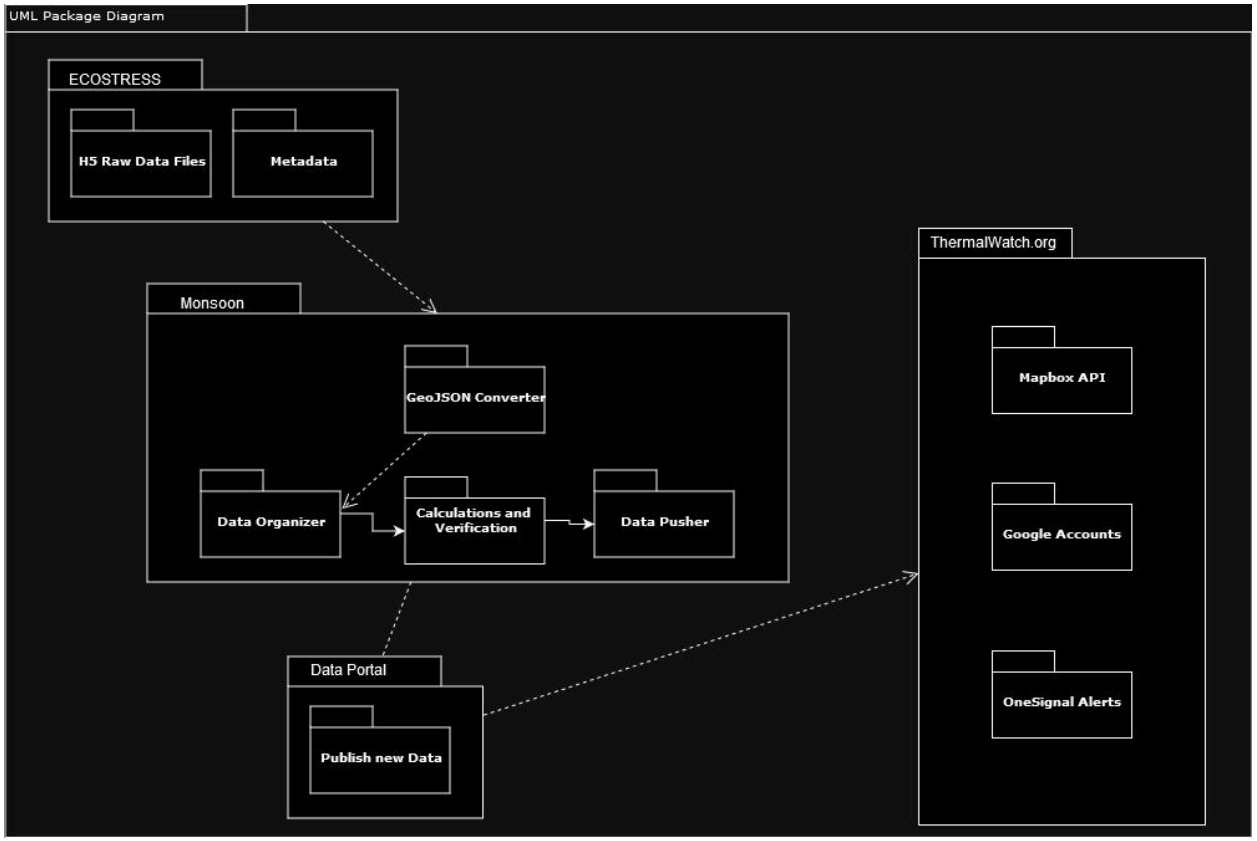


Figure 2: UML Package Diagram

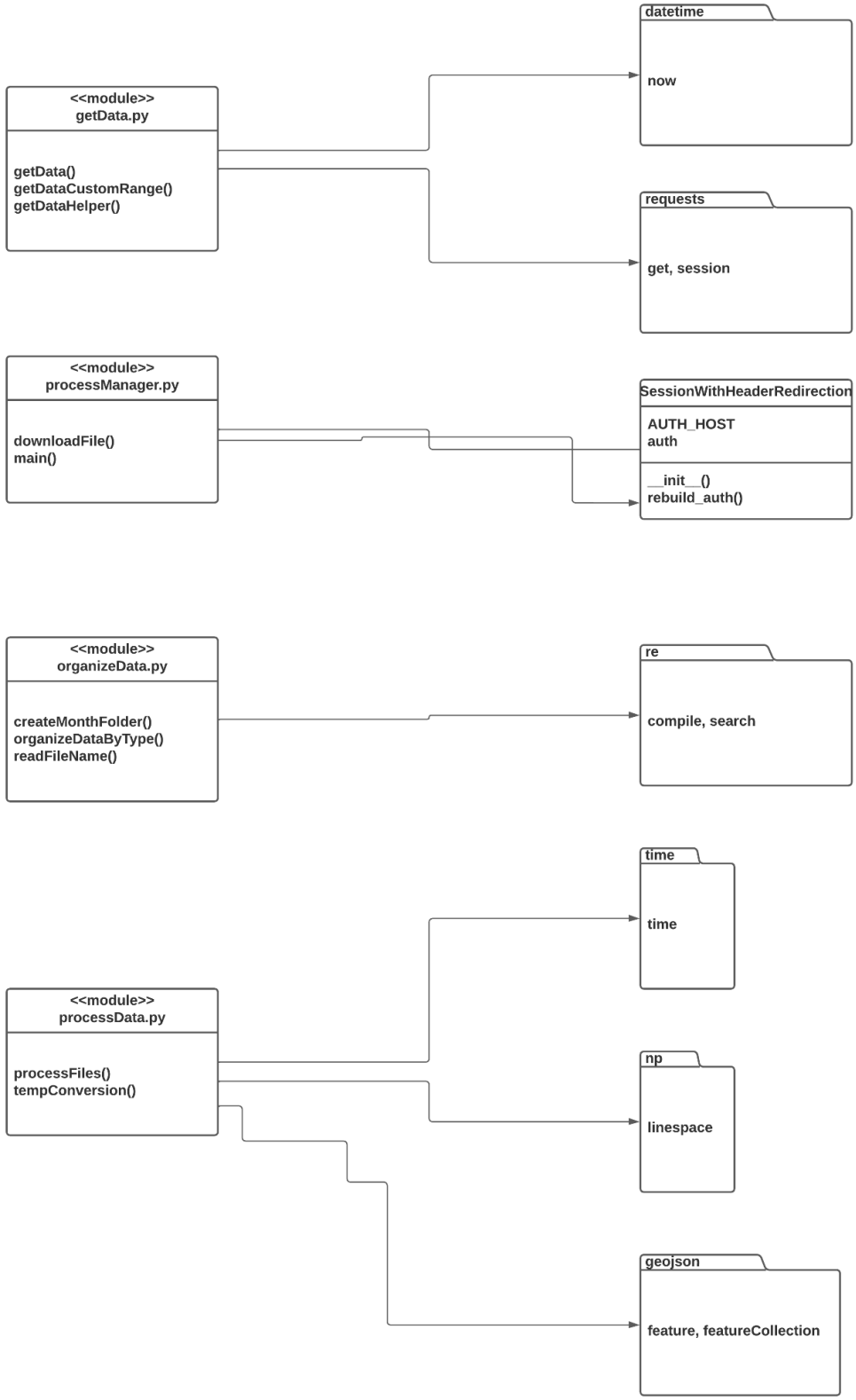


Figure 3: Functions Diagram

REST API - NAU Monsoon

Data Request - The front end of the web application will need to request the data from the database utilizing a data portal configured for Monsoon. In order to accomplish this, the web application will have to request a specific bounding box of what coordinates the user wants data for. The web application will also need to specify the time period desired for the temperature data.

Data Send - The database on Monsoon will need to place its processed data into a location available to the web application. In order to accomplish this, the database will need to 'publish' the processed data. Publishing is a Monsoon concept, which effectively means that the data will be available for anyone that accesses a custom URL.

Website

User Interface - The user interface will be made up of two main screens. The home page will consist of a virtual map with temperature overlaid on top of the virtual map. There will be an option to change what time the temperature was gathered from. The other screen will be the account screen, in which users will be able to set up custom alerts.

Custom Alerts - In the account screen, users will be able to set custom temperature thresholds for custom regions.

Google Sign In - Our website will be using google accounts so we do not have to manage and keep track of the user sign in information, which could be a security risk. By using the Google accounts API, we will be able to implement an easy sign in system that is secure and accessible to most of the public, since Google is a very popular product.

Mapbox - Mapbox is an open source mapping service that uses high quality satellite imagery similar to that of Google Maps or Apple Maps, and it is mainly used in navigation applications. We will be harnessing the imagery and map UI to show Hawaii in an impressive looking user interface.

Implementation Plan

Spring 2024 Capstone
This semester will include the bulk of the software development process. We will be splitting our development process into two parts. The first part will focus on the backend design of the project. For our backend design, we will be setting up NAU's monsoon to manage and organize the data from NASA's ECOSTRESS in a database in a way that is easy to access from. The second part of our project will focus on the Frontend of our project by creating an intuitive and beautiful design.

Data Organization ~ January 2024
We will be using Python and the Django framework to successfully manage and organize the data in the Monsoon server. We will be creating scripts that automatically request the data from the NASA satellite and process it to determine if the temperatures we are currently viewing are outliers.

Alert System ~ February 2024
We will be relying on an API to function for our alert system. The script for this will be included in the backend of development, so it can notify people as soon as possible.

User Authentication ~ February 2024
Instead of managing usernames and passwords, we will be incorporating google account's API to ensure users can easily log in and access our web application.

Web Application ~ February 2024 – May 2024
The web application is a crucial part of the project because it will be the part that users interact with. Users will be able to request data and control which regions they want to monitor through the web application. Our website will be using Mapbox AI because of how easy it is to implement our data with it. We will be showing historical temperature data and current data on the website on a 2D/3D map that has an overlay for the temperatures.

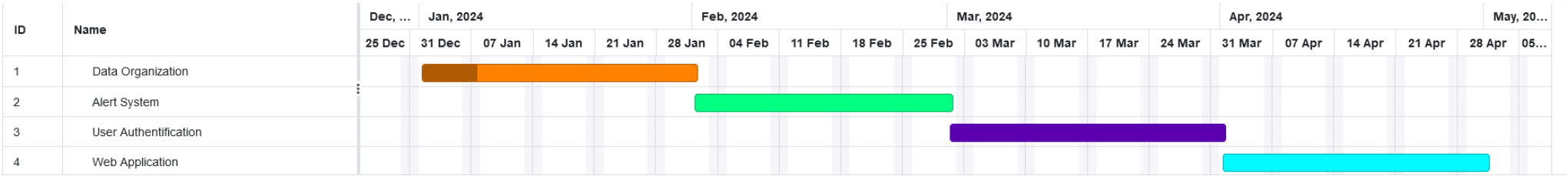


Figure 4: Schedule

Conclusion

With the increase in wildfires and rising coral reef temperatures, the Hawaiian islands are vulnerable to the threat of climate change. Team Satellite Fire Patrol is working on a project with clients Dr. Camille Gaillard, PhD student Benjamin Wiebe, and Dr. Christopher Doughty to create a software that helps identify warning signs in the Hawaii area from real-time satellite thermal data.

Our software will implement features in relation to these points:

- Historical data
- Real time data
- User authentication
- Custom alerts
- Overlaying thermal data

We believe that we will integrate these features separated into the front end and back end using NAU's Monsoon.

- NAU's Monsoon Integration
- Front End Creation using Mapbox

In conclusion, our proposed solution will involve multiple open-source technologies that will take advantage of NASA's data to create an early warning system for wildfires to allow for the mitigation of ecosystem loss and wildfire damages. With our planned steps detailed in this document, we will complete our project to the best of our ability and create a helpful and useful final product.