GONZAGA UNIVERSITY
School of Engineering and Applied Science
Center for Engineering Design and Entrepreneurship

PROJECT STATUS REPORT
December 7th, 2023

Wildfire Resource Simulator

Prepared by:

Kevin Dang	Claire Yegian
Drew Bogdan	Vincent Do
Kole Davis	

1 Project Overview

The contemporary American wildfire fighting strategy faces skepticism, especially regarding the intentional non-action use of aerial firefighting resources. Fire Armada aims to address this skepticism by raising awareness about the potential benefits of deliberate fire suppression utilizing aerial firefighting. The goal is to convince the general public of the efficacy of this strategy through data-driven simulations that show the lack of effort and optimization from the current firefighting strategy.

Our approach involves developing a two graphic simulator that leverages existing flight data for firefighting aircraft and historical wildfire data in the United States between 2020 and 2022. The first simulator will visually represent the wildfire data and the flight paths of large firefighting air tanker. Additionally, the team will develop a machine learning model to identify and optimize dispatch strategies for firefighting air tankers in a second simulator. This second simulation will keep the same visualization of the fires and tanker bases, but change how the large air tanker aircraft are dispatched from their respective tanker bases.

The following information outlines the major features of the Simulator desired by Fire Armada and the responsibilities for each:

- Database
  - The database is used to store collected and clean data from any API calls and other sources.
- Visual Map
  - The visual map serves as a visual representation of fires, air tanker bases, and large air tankers on a map of the United States.
    - 1st stage: Static map
    - 2nd stage: Plot fire, flight, and tanker base data
    - 2nd stage: rolling map on a timescale
- Machine Learning Model (Dispatch Optimizer)
  - The model will optimize fire outcomes based on the model's distribution of aerial resources around certain regions. This involves the movements of these large air tankers from tanker bases to fires, time the fires burn, and how much area they burn.
    - 1st stage: functioning predictive model using the optimizer
    - 2nd stage: integration of the model's predictions onto the map

The project deliverables below encompass what the team intends to accomplish for Fire Armada:

- Data Retrieval
  - There will be interactions with the FlightAware API to collect flight path data and aircraft data, tanker base data will be parsed from publicly available pdfs from the US Forest Service, and fire data is public and found on National Interagency Fire Center's website.
  - Python Scripts will be made to format all the data gathered to be in the MariaDB database hosted on Gonzaga's server.
  - Data from the database will be transmitted to the front-end simulation utilizing API endpoints.

- Mapping Fire and Aircraft Data
    - The application database will pass necessary fire and flight data to the front-end application to map the data.
    - The web application will provide visual mapping of fire metrics, fire conditions, and aircraft data. Some examples of fire metrics would be, acres burned, acres burning when fire was found, etc. In general statistics, we can provide average times fires were put out in months and years based on the fire out and containment times.
- Machine Learning Model Simulation
    - The web application will have an implementation of a machine learning model to optimize the dispatch of large firefighting air tankers within their respective regions. The optimizer will attempt to simulate alternate dispatch times for air tankers at different air tanker bases within their regions to see if fire containment is faster and burns less acres.
    - The user interface and mapping will be refined to contain an approximate perimeter based on the acres a fire covers.

2 Summary of Work Accomplished

Our original release plan was to complete a database and fill it with data by the end of this semester. As of this writing, we have built half of the database, and filled it with some data, but the data collection is not complete. Our database contains two types of data: early flight data and fire point data. However, the flight data we have now is not useful without filtering. The fire point data provides statistics and location on all the fires in the year range we are looking at. To make up for this incomplete goal, we have also started and made significant progress on the front end development. The front end now has a fully working map rendered by the Google Maps API, and we are able to plot the fire points on this map and filter them by date. This front end is created using React, and interacts with a local JavaScript server's endpoints to request information from a MySQL database hosted on Gonzaga's server.

3 Working Prototype

The current alpha prototype consists of a working React web application that renders Google Maps onto the application. The team was able to collect the fire data and display it on the web application using the Google Maps API. A user can interact with the map, fullscreen the map, and see generic Google Maps icons of where fires were between 2020-2022.

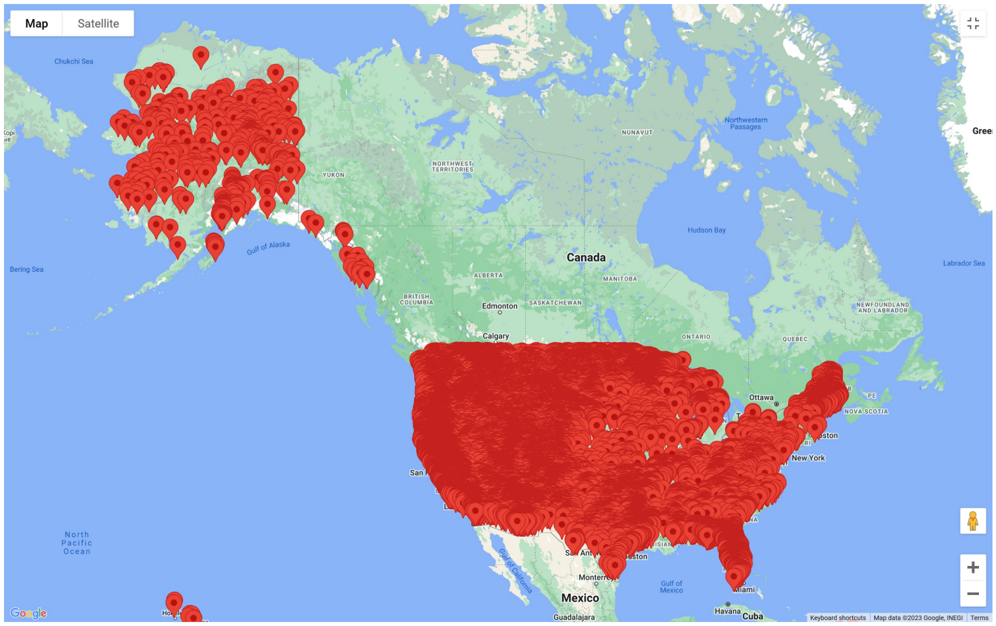

Figure 1: Fire points displayed using the Google Maps API.

The layout of the application, shown in the figure below, details the interactions between each part of the simulator. The React application interacts with a local JavaScript server's endpoints to request information from a MySQL database hosted on Gonzaga's server. The machine learning model has not been implemented yet, however, it would send its results to the server for the application to access.

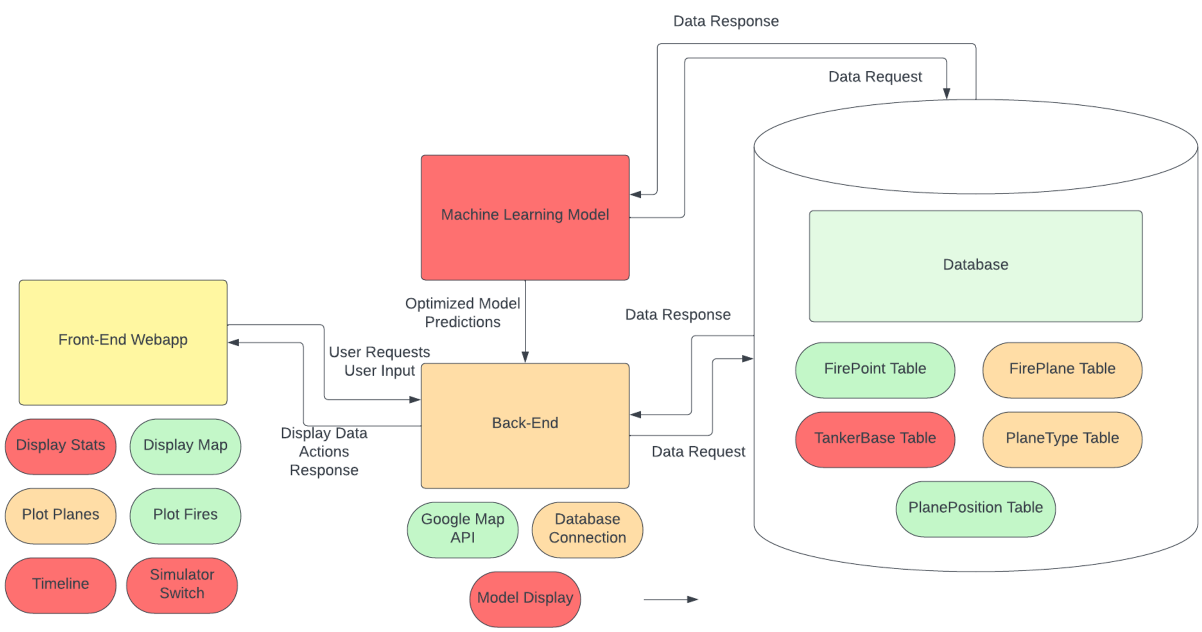

Figure 2: Model of all parts for the web application.

4 Revised Product Backlog

The revised product backlog includes sub-requirements for the major requirements from the previous backlog iteration. The majority of the new requirements break down and describe the steps towards completing the earlier project deliverables (database and visual map). The current data backlog involves work on the flight data (which has been difficult to gather completely), while the fire data collection has been completed. The machine learning model was redefined to be a fire plane Dispatch Optimizer as we gathered more knowledge about the aerial firefighting industry, and the corresponding backlog requirement was changed to better describe its features. Updating the product backlog with these additional requirements was done in biweekly sprint refinement meetings during the alpha development, which was necessary because the previous backlog only covered the high-level project requirements.

Table 1: Revised Product Backlog

Requirement	*Description*	*Major Feature*	*Priority*	*Estimate*
Flight Data Collection	Collecting and cleaning data on necessary metrics for aerial resources in the U.S.	Database	High	21
FlightAware API Requests for Data	Planning to spend the project budget on firefighting plane flight data via API calls.	Database	High	5
Create ADS-B Exchange API (Backup)	Creating an API to parse ADS-B Exchange flight data JSONs to tables. Testing works on sample flight data, still requesting the full dataset and creating endpoints.	Database	Low	8
Clean Flight Data	Preparing raw flight data to be inserted as database table rows. Flight data modules have been made but don't include all information yet, and the flight dataset is incomplete.	Database	Medium	2
Design Flight Data & Plane Schemas	Creating database tables for fire plane information and the historical flight paths taken. Schema for flight points has already been made with test data.	Database	Medium	3
Establishing Graphics	Setting up graphical representations of fire incidents and individual plane flight paths. Web application displays Google Maps of the U.S. and can plot fire point markers.	Visual Map	High	34
Location Data	Selecting necessary location-based data to represent aerial resources and fires on the map.	Visual Map	Medium	13
Fire Stats Display	Making calculations on data to display statistics about the given fire on the dashboard.	Visual Map	Medium	13

Fire Timeline	Mapping temporal fire data onto a timeline and implementing it into the map via a slider. Demonstrated ability to query certain dates which will translate into time ranges for fires.	Visual Map	Medium	55
Air Tanker Base Data Collection	Parsing and cleaning tanker base data from base directory document, which will be used as input for the dispatch optimizer.	Dispatch Optimizer	Low	3
Machine Learning Model Training & Dispatch Simulation	Training a machine learning model for the dispatch optimizer on fire, flight, plane, and base data to predict plane dispatch times, bases, and fires that minimize fire growth and burn time.	Dispatch Optimizer	Medium	21
Estimated Difference in Fire Size Based on Better Dispatch	Estimating the fire containment results based on optimized aerial resource dispatch.	Dispatch Optimizer	Low	13

5 Revised Product Release Plan

Below is our revised product release plan for the remainder of the school year. It details the major goal posts we plan to hit in the coming months up until the beginning of April. Changes were made to the target completion dates for these milestones to better reflect our current progress for the project, and what that means for the completion of future milestones.

Table 2: Revised Major Milestones

Milestone	*Description*	*Target Completion Date*
Gather Project Funding	By this point, we will have calculated and allocated the appropriate amount of budget for our project and its components.	First week of December
Complete Database	Our database is complete with all of the data we intend to work with for this project. Data that is cleaned and ready for other project components to utilize.	Third week of January
Functional Dynamic Map	A map displaying the US overlaying fire location/area data with data regarding the flight paths of aerial firefighting tankers and their tanker bases.	Second week of February
Working Machine Learning Model	A machine learning model using our collected data to optimize the flight paths of aerial firefighting tankers.	Third week of March
Integrated Machine Learning Model w/ Map	Completion of the project’s two simulators: the historical data simulator, and the tanker route optimizer.	Beginning of April

The following is the Sprint Release Plan for our project beginning after our final Project Plan is submitted and going until the final presentation of our project. Please note that sprints towards the end of the year are less precise than those beginning in a few weeks, and our plan may be adjusted as the project progresses. We will work on code commenting and basic documentation as we develop but will use our final sprint and the weeks before our presentation in May to clean up and add higher-level information to that documentation.

Table 3: Sprint Release Plan

Sprint Date	*Sprint Goal*	*Backlog*	*What we will demo*
4th Week in Oct to 1st week in Nov	Design and Create Database & Graphics Start	Working on Establishing Graphics and Data Collection	Database
2nd Week in Nov to 3rd Week in Nov	Start to Gather Data & Basic Window Requirements Gathering	Fire Data	Fire data in database and window
4th Week in Nov to 2nd Week in Dec	Continue to Gather Data & Start Basic Map	Flight Data	Flight data in database and window
3nd Week in Dec to 3rd Week in Jan	Continue to Gather Data & Basic Map	Fire and Flight Data	Full Data in database and window with map
4th Week in Jan to 5th Week in Jan	Clean and Refine Data & Basic Map Functions Begin Building and Training Model	Working on Establishing Graphics & Fire Stats Display	Clean data in database and window with map and functions
1st Week in Feb to 2nd Week in Feb	Begin Map Integration	Working on Location Data & Basic Simulation	Machine Learning Model and beginning of map and integration
3rd Week in Feb to 4th Week in Feb	Refine Model & Dynamic Map	Location Data	More of the map integration process
1st Week in March to 3rd Week in March (includes Spring Break)	Working Model Complete & begin Data Integration into Map	Basic Simulation for Flight Path Optimizer	Advanced model for prediction and last bit of map integration
4th Week in March to 1st Week in April	Refine model for large scale predictions, refine Map integration, first usability test	Estimate difference in Size Based on Flight Path Optimizer and working on Fire Timeline	Large scale predictions and display on the map & somewhat combined integration
2nd Week in April to 3rd Week in April	Full integration of map and machine learning model, second usability test, deployment, and cleaning up documentation	Establishing Graphics Complete & Fire Timeline Complete	Fully integrated Map and machine learning model
Final Push	Not a sprint, but a bit of time for tying up loose ends		Our final product, featuring our completed simulators and a functioning web-application

6 Project Risks

Our initially identified risks were API Query Limits, Database Host Outage, Unity Integration Version Control Errors, and Project Usage Ramifications. We have decided to remove Database Host Outage because it is not a large enough risk to warrant being addressed. Unity Integration Version Control Errors is no longer relevant because we have pivoted to using Google Maps API for the map feature. API Query Limits remains a risk, although it seems that API usage cost will be a larger limiting factor now than time. We have updated the risk to reflect that, and have secured a budget to mitigate it. Project Usage Ramifications remains a risk because the intent of our sponsor remains to create a commercial using the project to raise public awareness about the inefficiencies of contemporary American wildfire fighting. If the data, statistics, or model we produce do not align with the intent of the commercial, we are concerned that it might be misrepresented in a dishonest way because our sponsor is so focused on their narrative and seems to relegate factual evidence as a secondary priority. In order to mitigate this, we will be careful to cite accurate information and realign with our sponsor on the goals of the project. Additionally, we may add disclaimers to any final products that we remain concerned may be misconstrued.

We have added two new risks to the project: Inadequate Data and Non-Predictive Model. Inadequate Data is a risk because we need data containing sufficient descriptive and predictive attributes on a fairly granular time scale (minutes) for three recent years (for which data might not yet be fully published). It has already proven difficult to find and collect all of this data and then compile it in a comprehensive way. We may need to have alternative attributes or be willing to compromise on time scales or years. The fewer datasets we have to use, the easier it will be to combine them. We will need to be conscious of limitations and backup plans as we look for data. If we can't find the specific attributes we need or are unable to join the datasets, we will need to implement those backup plans (alternative attributes or scope compromise[1]).

A Non-Optimizing Model is a risk because we are assuming that our model will be able to optimize the system on the data we give it, and in doing so, will identify an inefficiency or area for improvement. If it is not able to optimize or doesn't find an area for improvement, we may not be able to achieve the project's goal of demonstrating inefficiency in the US's wildfire management system. We will need to be careful to choose data with predictive power to feed the model. We'll have to choose the model we use carefully to best fit the problem, and we will likely have to try a few different models or training sets to reach the final product. There is still a possibility that the risk will become reality if we can't produce results that demonstrate areas for improvement, or can't produce conclusive results at all. In either of those cases, we'll still have to present our findings, as they are. In the event that our final product does not reveal what our sponsor hopes it will, we can suggest other potential areas of inefficiency, additional data that may improve the model, and other routes forward for potential future contributors to our project.

[1] We may need to execute a scope change for this project anyway because our sponsor has changed their expectations for the project. If this does result in a scope change, the dispatch optimizer (ML model feature) would be pushed to a stretch goal and replaced with a similarly difficult feature, perhaps a more detailed statistical analysis.

7 Delivery and Maintenance Considerations

Meeting Information (no changes)

- The team will be working in 2-week sprints to complete project deliverables, providing daily asynchronous stand-ups in the team Discord server.
- Team meetings are held at least weekly on Mondays in BCSE 006.
- Other weekly and bi weekly meetings (held on Zoom and Microsoft Teams):
    - Meetings with our sponsor are set every other Wednesday at 4:30pm PST.
    - Meetings with our advisor are set on Mondays at 6:00pm PST.
- At the end of every sprint, we update our advisor and sponsors with a demo of new work during the meeting

Work Splitting (no changes)

- We conducted a work preferences survey for the team to determine what areas each member wanted to work on. The current work splits are as follows:
    - Front-End Development will involve working with designing the map and interface of the simulations.
        - Kole Davis
    - Back-End Development will involve data collection, data cleaning, database management between the client and the server, and model building.
        - Vincent Do
        - Drew Bogdan
    - Full-Stack Development will involve both front- and back-end development of the simulators.
        - Kevin Dang
        - Claire Yegian
- It is important to note that these work split projections are only preferences, as members of the team are expected to be flexible and assist with all areas of the project when needed.

Additional Tools (no changes)

- Work is tracked on Trello, both in cards and comments on those cards as specific areas of the project progress.
- Discord is used to communicate new concerns, suggest ideas, and set up smaller-group meeting times.
- Email is used to communicate with the advisor, DAB, and sponsor between meetings.

Method for Packaging and Deployment

- For the release package, we intend to ship the production build of the website application we are building along with database setup files.

Maintenance Considerations

- We do not have any maintenance considerations at this time.

8 Project Management Considerations

Our team has worked in two-week sprints to begin our project. We communicate daily with asynchronous Discord standups, and work together in small groups as needed between our weekly team meetings. We meet with our advisor weekly and our sponsor bi-weekly. The system we initially devised has worked well for us. We have had to iron out some technical coordination on standards for coding and Trello cards, and struggled at the beginning with doing the bulk of our work at the beginning of the week. We have since developed a guide for standards and improved on spreading work out more evenly. Moving forward, we feel confident with our teamwork system and will focus on continuing to collaborate.