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

PROJECT PLAN
October 9th, 2023

Wildfire Resource Simulator

Prepared by:

Kevin Dang

Claire Yegian

Drew Bogdan

Vincent Do

Kole Davis

Reviewed by:

Jasmine Jans
Faculty Project Advisor

Michael Marzetta
Project Sponsor/Liaison

Jesse Phillips
Design Advisory Board Member

Christy Marzetta
Project Sponsor/Liaison

1 Project Overview

1.1 Project Summary

The project aims to demonstrate the effectiveness of aircraft-based resources in combating wildfires. The goal of Fire Armada is to benefit fire agency management by developing better fire prevention and suppression tactics, as well as show to federal, local, and state representatives that the strategic usage of aircraft can improve response times and fire containment. Since potential Congressional supporters are hesitant about the reliability of this approach, this project serves as one means to convince them and the public by using mathematically sound and data-driven simulations.

The project will provide a graphic simulator that leverages existing flight data for firefighting aircraft and historical wildfire data in the United States in the last few years. The simulator should be capable of mapping aerial movements to corresponding wildfires and visually representing the progression of both, as well as estimating the time spent suppressing fires and other time in flight or refueling. Additionally, the project may provide optimization simulations to measure the outcomes of improved resource coordination, which may include early preventative fire detection, early response, and resource estimation for target metrics.

1.2 Project Objectives

- Build a historical simulator that can visualize past data of aerial firefighting vehicles and the progression of fires throughout the country.
    - The data for the movement of the aerial fire fighters should be researched and filtered by the team using data such as tail number, call signal, or radar footprint through the ADS Exchange or similar historical registry.
- Use the historical simulator to identify the following metrics.
    - Value-added time (dropping suppressant vs flying waiting, filling, on hold)
    - Non-value-added time (time grounded, time in-flight, time refueling)
- Build a second optimization simulator which can simulate a better distribution of aerial resources to result in better fire containment using
    - Data from a given historical fire
    - A machine learning model to predict containment based on resource management
- Utilize both simulators to estimate resource distribution required to achieve targeted fire suppression metrics.
- Run simulations on historical US wildfire data and forecasted models to achieve improved resource strategies.

1.3 Project Stakeholders

The current stakeholders for this project are the following:

- Developers (Gonzaga University):
    - Kevin Dang
    - Vincent Do

    - Kole Davis
    - Claire Yegian
    - Drew Bogdan
- Sponsors (Fire Armada):
    - Michael Marzetta
    - Christy Marzetta
    - Additional board members:
        - Kelly Maloney
        - Rick Taylor
        - Bill Hockett
- Faculty Advisor:
    - Jasmine Jans
- Design Advisory Board Member:
    - Jesse Phillips
- Congressional Individuals:
    - Local Eastern Washington Politicians
- Target Users:
    - Interested Parties

1.4 Project Deliverables

- Project Plan Deliverable
    - Details of the project will be documented and delivered to the sponsor.
- Data Retrieval Deliverable
    - Interactions with APIs and fire data from live databases and fire metrics are passed into a database.
    - Data from databases will be pulled into the front-end simulation to only display fire metrics.
- Mapping Fire and Aircraft Data Deliverable
    - The simulation should be able to display accurate information for the user. The application databases should be able to pass necessary fire data to the simulation.
    - The application will provide some visual mapping of fire metrics, fire conditions, and aircraft data.
- Machine Learning Model Simulation Deliverable
    - The simulation will have the minimum viable product of the sponsor's wishes.
    - All documentation will be updated for any relevant information on the application.
    - The application will have some implementations of a Machine Learning Model to simulate future fires and predict fire spread.
    - The user interface and mapping will also be refined fire simulation based on the predictive model.

1.5 Project Scope

Currently this project specifically has no existing aspects. There are multiple other slightly similar projects, but also differ on certain ideas that Fire Armada wanted in this product. But as for what code or project work has been done, we only have the project proposal from Fire Armada. The rest of the project will be developed from scratch this year and likely continue for years to come.

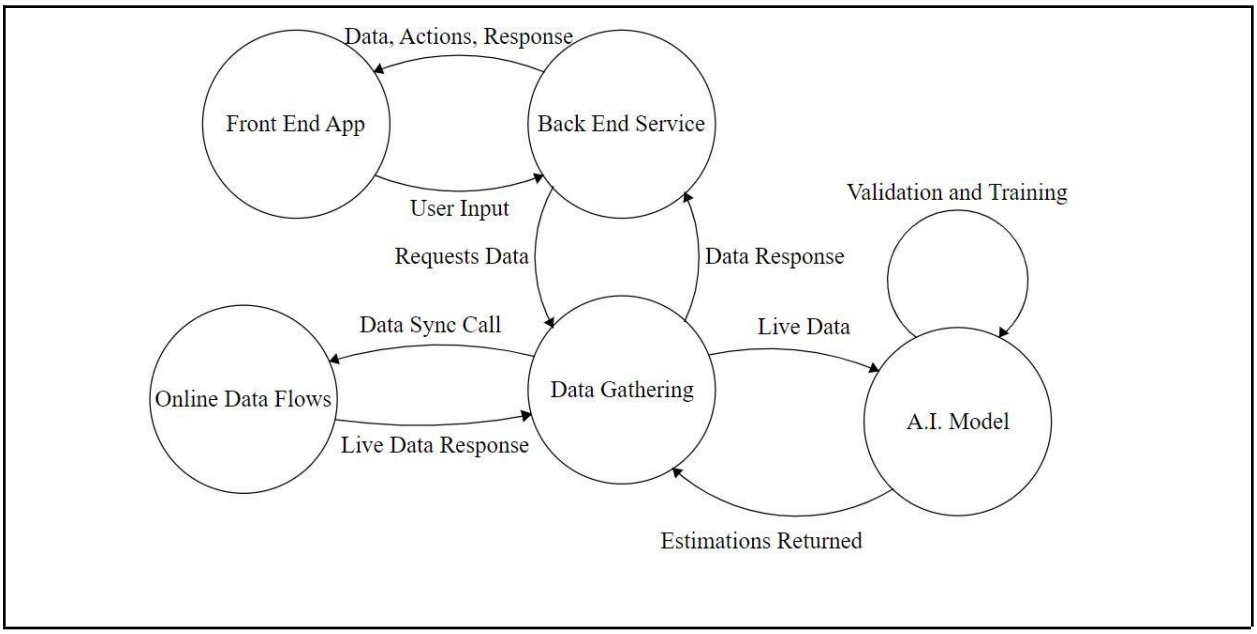

Figure 1: Context Diagram

1.6 Related Work

Existing Systems:

- Watch Duty | https://www.watchduty.org/how-it-works/overview
    - Mobile application offering real-time wildfire alerts and reports throughout the western United States. Provides information related to wildfire perimeters, air quality monitoring, wind direction, and so on.
    - Run by volunteers manually obtaining information from 911 dispatch centers and official first responder reports.
    - Similarities: Wildfire monitoring and awareness, information systems
    - Differences: Not specific for aircraft, only uses real-time data, restricted to western US states, volunteer-driven data.
- FIRMS | https://firms.modaps.eosdis.nasa.gov
    - Uses satellites to gather, document, and display real-time fire hotspot data across the United States, Canada, and other spots across the globe.
    - Similarities: Actively collects and charts wildfire data
    - Differences: Utilizes satellite technology that is outside of this project's scope, provides global fire data

2 Project Requirements

2.1 Major Features

The project aims to visually demonstrate that better distribution and allocation of aerial firefighting resources will result in more efficient and effective fire containment. To accomplish this, we will collect data on aerial resources (planes) and fire histories in the US. The data will then need to be cleaned and stored in a central location, for which we will use our first major feature: a database. Using that data, we will build our second major feature: a map of the US to visually represent fires and their containment. In parallel, we will build our third and final major feature, a machine learning model for predicting fire outcomes based on distribution of aerial resources. Once we have a static map and a predictive model, we will begin to integrate the two features to build a comparative map (perhaps two maps side by side) showing how a historical fire was contained vs how it could've been contained with improved resource distribution. This map (or maps) will integrate a timescale to show fire progression and containment over time.

Based on Fire Armada's goals, we considered a few other major features that we have not included as targets for our project. The first of these features was real-time mapping of active fires. We believe that we don't have the time or resources necessary to obtain or handle data at this scale. The second of the rejected features is a forecast simulation for fires that have not yet occurred or are currently being fought. We do not believe that we have the necessary background knowledge in fire science or the necessary understanding of other contributing factors (wind, type of ecosystem, drought conditions, etc.) to create a forecast with reasonable accuracy or attention to aforementioned contributing factors. The final feature we did not include in our targets is the identification and coordination of successive sequential aerial bombardments when necessary and cost-effective. We do not have the policy or contracting knowledge or access to the financial data necessary to achieve this feature.

In order to determine major features, we reviewed available data and contextual information for firefighting. This gave us an understanding of what data we have at our disposal, and therefore what questions we may be able to answer, and it also made us aware of our limitations in terms of expertise.

Table 1: Major Features

Feature	*Description*
Database	Database to store collected and cleaned data from API calls and other sources.
Visual Map	Visual representation of fires and resources on a map of the US. ● 1st stage: static map ● 2nd stage: rolling map on a timescale
Machine Learning Model	Predicting fire outcomes based on distribution of aerial resources. ● 1st stage: functioning and predictive model ● 2nd stage: integration into map (best resource scenario side-by-side with the actual progression of the historical fire)

2.2 Initial Product Backlog

This project has 3 major features that we plan to create. Within each of these major features there are a couple of requirements that we want to see. For each feature the requirements have set forward is something we want completed before we set the feature as completed in the project.

The following are requirements for the Database:

- Fire Data Collection
    - We will consolidate the fire data we have access to into a database and clean the data in the process to maintain all data under the same units.
- Flight Data Collection
    - We will gather flight information on fire fighting planes and get their flight paths, idle time and distance to active fires. This data will be collected and used to display in the Fire Plane Data Requirement.

The following are requirements for the Visual Map:

- Fire Timeline
    - The Fire timeline will show the path of fires as time passes and shows how the fire either grows, is contained, or shrinks. This data will be displayed on the map and time is controlled by the user to see dates and times during the day.
- Establishing Graphics
    - Set up a map of the US, likely using game engine graphics and match latitude/longitude values of the collected data to the map.
- Location Data
    - Alongside the Timeline, flight data of fires and planes will need to be displayed on top of the fire data to show the location of the fire and where the plane took off from, its flight path, and its idle time.
- Fire Stats Display
    - This will show the stats of the fires at the current time: the acreage that is on fire, was burned, trees lost, and possibly a statistic on smoke created.

The following are requirements for the ML Model:

- Basic Simulation for Better Attack
    - The simulation will be trained on how fires are attacked and take into account the time that planes are idle versus how long the fire is active. It will then learn to show how the fire would have best been attacked (early detection and improved response time). This will show the plane trajectory changes on a different simulator tab.
- Estimated difference in size based on better attack
    - Following the information the first model is trained on with the flight trajectory, the model will show the difference in land burnt had the fire been put out in the more efficient manner that the original model found. This will give statistics on the saved acreage and trees that did not get burned.

Table 2: Initial Product Backlog

Requirement	*Description*	*Major Feature*	*Priority*	*Estimate*
Fire Timeline	Mapping temporal fire data onto a timeline and implementing it into the map.	Visual Map	Low	55
Establishing Graphics	Set up the graphic representation.	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	Low	13
Fire Data Collection	Collecting and cleaning data on necessary metrics for fires and their progression in the US.	Database	High	34
Flight Data Collection	Collecting and cleaning data on necessary metrics for aerial resources in the US.	Database	High	21
Basic Simulation for Better Attack Vector	Use data to create an ML model using historical fire data and resource use to predict better resource distribution.	ML Model	Medium	21
Estimated Difference in Size Based on Better Attack Vector	Estimate the containment results based on improved resource distribution.	ML Model	Medium	13

2.2 Additional Features

- Simulation(s) Integrated Onto Website
    - Integration/Embedding of the two simulations onto the Fire Armada site to be public for any interested parties.
    - If the third simulation described below was to be completed, it would also be embedded into the Fire Armada site.
- Forecast/Prediction Simulation
    - Build a third simulator that can utilize existing datasets to predict the best possible times to allocate firefighting resources in a given zone to minimize risk of fire outbreak
- Economics Data
    - Estimate the costs of resource allocation spreads and compare different approaches based on economic value.

3 Design Considerations

3.1 Initial User Interface Design

The user interface will consist of one page containing an interactive map of the US displaying historical data and firefighting predictions with improved aerial resource allocation. The user will be able to switch between tabs of the map to view historical versus simulated data. Using fire mapping and plane tracking, one tab of the map will use a timescale to demonstrate how a historical fire played out over time and how aerial resources were used in that fire. The second tab will show the same fires adjusted with our machine learning model's aerial resource improvement predictions and the resulting impact on the fire outcomes. Each fire will have a point that can be selected by clicking on it and will expand to a window showing specific metrics for that fire and its corresponding aerial resources. A side panel will display overall statistics (either fire-specific or general to firefighting strategy based on what we decide on).

This mockup includes our three main features from the Project Requirements: a map, a machine learning model, and a database. The map feature will be implemented in both the historical and simulation maps. The machine learning model will be implemented in the simulation map. Our database feature will be accommodated in the back-end but will also be used for the metrics, statistics. and mapping on our user interface.

Our sponsor would like a visual representation of fire resource allocation (particularly aircraft) in order for an accessible demonstration of how improved resource distribution could improve fire outcomes through earlier detection and response times. To this end, a map has been a part of our plan since the beginning of the project. We decided on a one-page user interface to keep the product simple and to-the-point in conveying information.

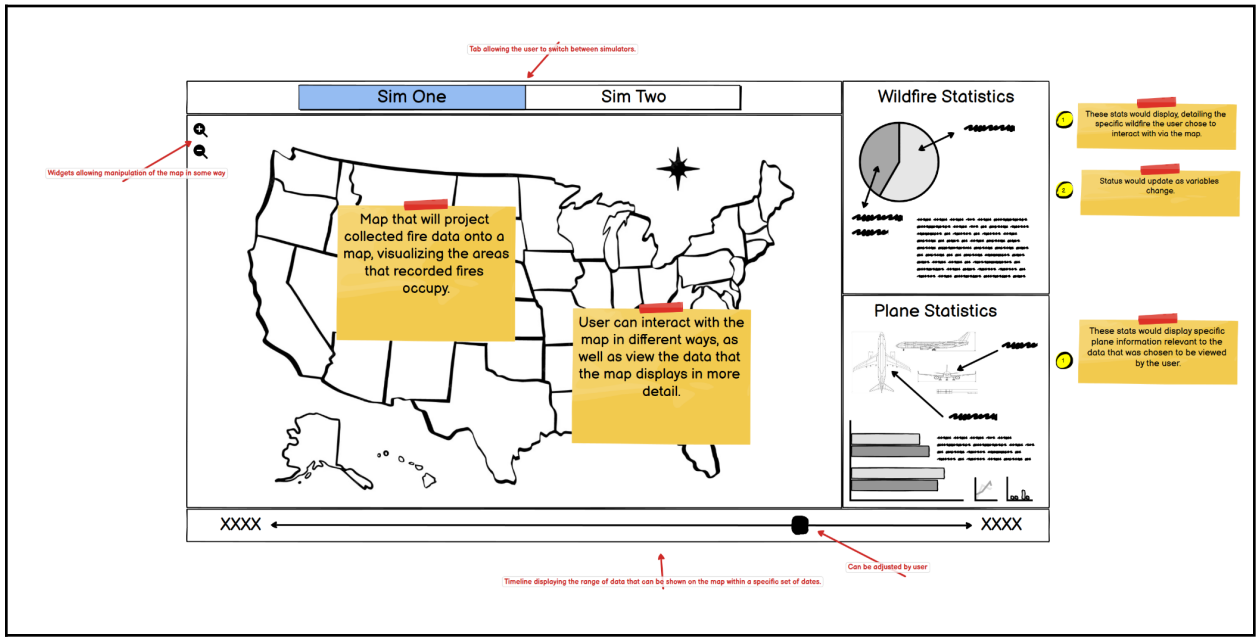

Figure 2: General UI Mockup

3.2 Initial Software Architecture

We will have a static relational database for which we will pull and clean data manually. The map will be built using a game engine (we plan to use Unity). The map, pop-up windows, and statistics side panel will be implemented in the front-end. The back-end will handle any operations that need to be passed from the front-end and the data it needs.

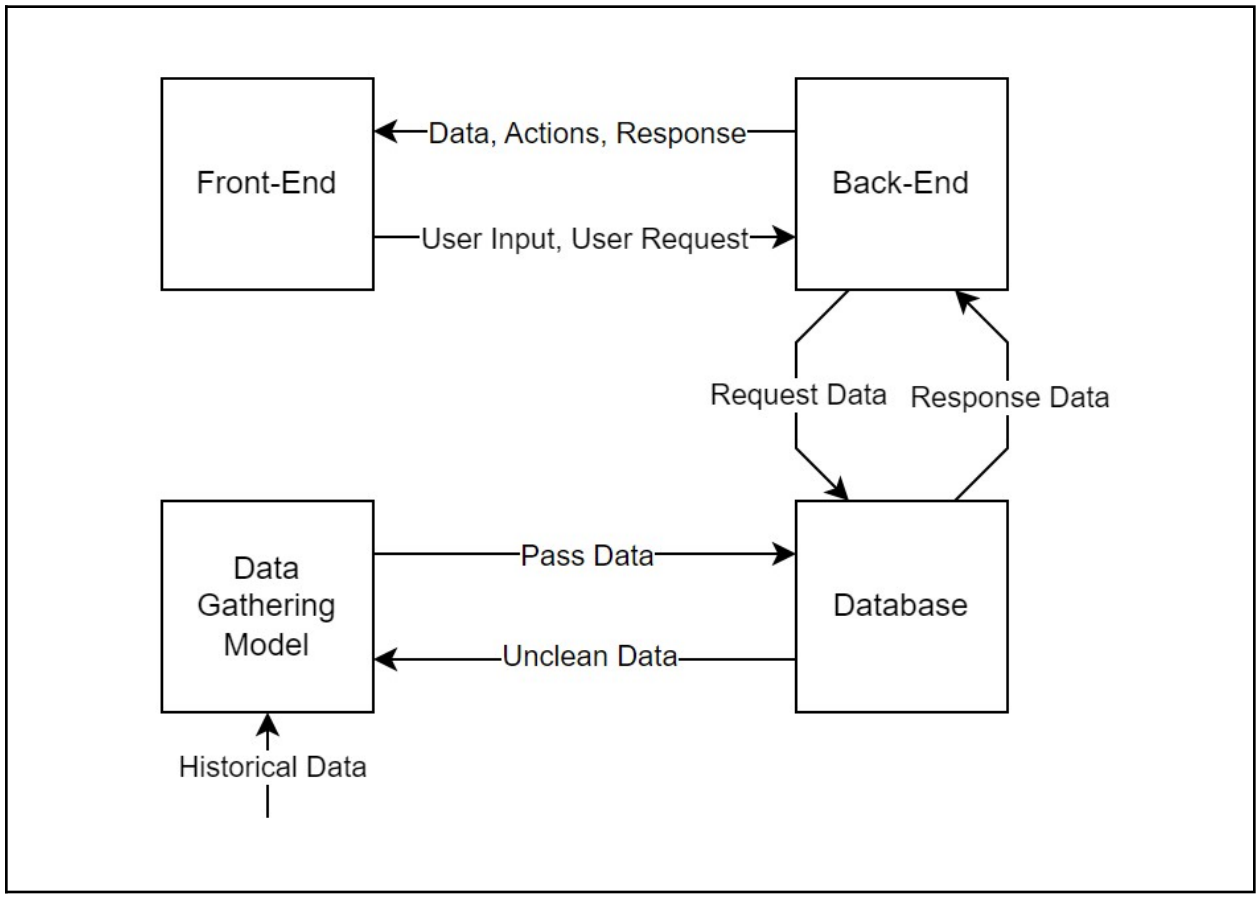

Figure 3: System Block Diagram

3.3 Development Environment, Tools, Languages, and Libraries

Languages:

- Structured Query Language (SQL)
    - We will be using SQL for our database. This system is very quick and can store a lot of data for cheap space requirements. This would be run on a Database Management System. The system has not been decided yet, but the two front-runners are Db2 and SQLServer.
- Python
    - Very quick and efficient language for its use of libraries. Easy to combine and work with APIs.

- C#
    - This will be used with Unity to set up the back- and front-ends for the application.

Libraries:

- Pandas
    - General data management library for Python.
- Keras
    - Specialized machine learning library with built-in connections to TensorFlow.
- TensorFlow
    - Deep learning and neural network Python library that will be used for our machine learning & deep learning predictions.

Tools & Integrated Development Environments (IDEs):

- Unity
    - Game engine we will be using to create and run the application. Has a large suite of tools to speed up the process and help us create a seamless working application.
- PyCharm
    - PyCharm is an IDE for Python. Very helpful for developing API solutions, has lots of valuable plugins for development.
- Visual Studio (and Visual Studio Code)
    - General development IDE with a wide range of tools and language support.
- Rider
    - C# development tool.

3.4 Initial Software Test Plan

- Since we will be dealing with hundreds of thousands of datasets based on fire locations and flight data, we would want to verify data in the database if we are pulling data from APIs.
    - We can create a data integrity auditor that would run SQL queries to test the tables in the database and check for certain discrepancies.
- We can also write unit tests in case of regressions that can occur when implementing new features to the simulator.
- Since it is possible that this simulator will be on the Fire Armada site or somewhere accessible through the Fire Armada site, it is important to secure any API endpoints that are used to gather and transmit data.

4 Project Risks

1. **API Outage(s)**
    - Why is this a risk?
        i. If one of the APIs we use goes down, it could disrupt simulation data collection and cause certain project features to stop working.
    - What actions can be taken to prevent the risk from happening?
        ii. We can store the API data we collect in our own separate database and have our simulators pull from that system instead of directly pulling from the APIs.
    - How will the risk be monitored?
        iii. We can monitor API uptime and outage reports while developing project features.
    - What events will trigger that would require the team to mitigate the risk?
        iv. One or multiple API outages occurring would trigger our need to mitigate this risk.
    - What will the team do if the risk becomes a reality?
        v. If one or multiple APIs we are using go down, our system will cease API data requests and instead use the info already pulled for the duration that the API(s) are down.
2. **Database Host Outage**
    - Why is this a risk?
        i. If we have not backed up the data we have collected, we could lose our data and our simulations would be unable to make database connections and run.
    - What actions can be taken to prevent the risk from happening?
        ii. We can create local backups of the database to prevent data loss.
    - How will the risk be monitored?
        iii. We can monitor our database machine from time to time and keep an eye on outage reports if they are available.
    - What events will trigger that would require the team to mitigate the risk?
        iv. Our database host either being faulty or going down completely would trigger our need to rely on local backups.
    - What will the team do if the risk becomes a reality?
        v. If our database host goes down, we will have data backups in place that we can feed into our simulators as a substitute.
3. **Unity Integration Version Control Errors**
    - Why is this a risk?
        i. Caution must be exercised when using GitHub in tandem with the Unity game engine, as Unity is not natively compatible with GitHub and could break the project if not properly incorporated.
    - What actions can be taken to prevent the risk from happening?
        ii. We can put in place the proper environment structure Unity needs to be compatible with GitHub, as well as maintain a standard Unity environment version and gitignore file among the team. In addition, we can declare a branch we will use to build and run the project. We can also revert repository states if the situation calls for it as a last resort.

- How will the risk be monitored?
    iii. We can monitor each other's commits and pull requests to the repository and what Unity files will be changed in those commits.
- What events will trigger that would require the team to mitigate the risk?
    iv. A loose commit could result in version control errors that we would need to address via reverting the repository branch(es) to a previous working state.
- What will the team do if the risk becomes a reality?
    v. If errors such as these are encountered, we can work as a team to fix the issue or revert the repository.

4. Project Usage Ramifications

- Why is this a risk?
    i. It is unclear what the impact will be when our project is finished and is used for its intended purpose: as general propaganda to make the public masses aware of this situation. Regardless of what form these impacts will take, our names will be tied to the project itself, so we should care about them.
- What actions can be taken to prevent the risk from happening?
    ii. We can ensure that the project we create is not used in ways that were not specified or proposed by the sponsor at the beginning of the project's design proposal.
- How will the risk be monitored?
    iii. We can monitor what this project gets used for (commercials, videos, websites) off and on, as well as touch base with our sponsor about future plans to utilize the project and/or the data it projects.
- What events will trigger that would require the team to mitigate the risk?
    iv. An event such as the project or project data being used in a way that was not initially agreed upon, in a potentially illicit manner. This would require us to attempt to mitigate this risk.
- What will the team do if the risk becomes a reality?
    v. We can talk with our sponsor about repercussions, or potentially work on publicly condemning that the project or its data be used in an illicit fashion.

5 Initial Product Release Plan

5.1 Major Milestones

Our major milestones will be the completion of our Project Plan, amassing a workable amount of data, building a functional map, constructing a machine learning model to predict on our data, and finally integrating the ML model and the dynamic map. These milestones encompass our three major features: a database, an interactive map, and a ML model. We ordered them in this way because we need the plan to begin development, the data to begin working with the ML model, and both the ML model and the map prepared in order to integrate the two. The map and the ML model will be developed in parallel because they are fairly independent of each other until we begin integration.

Table 3: Major Milestones

Milestone	*Description*	*Target Completion Date*
Complete and Present Project Plan	We will have a fully revised and approved plan outlining our project, goals, and rough task timeline.	October 18th & 19th
Workable Amount of Cleaned Data	Cleaned data encompassing all 50 states for planes and fires with the minimum necessary metrics and ready to begin development with the ML model (not necessarily fully collected, but enough to start working with).	End of January
Functional Dynamic Map	A map displaying the US with the ability to represent fires, planes, and their changing locations over time.	Middle of February
Working Machine Learning Model	A ML model using our collected data to make predictions about fire outcomes (does not need to be our final, fully refined model, just something we can work with).	End of February
Integrated ML Model with Map	Two simulations, one with historical data and one with predictive data, both displayed dynamically on a timescale.	Beginning of April

5.2 Initial Sprint Releases

The following is our 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. We will also use that time for usability testing and deployment.

Table 4: Sprint Release Plan

Sprint Date	*Spring 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	Fire Data	Fire data in database and window
4th Week in Nov to 1st Week in Dec	Continue to Gather Data & Start Basic Map	Flight Data	Flight data in database and window
2nd 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	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 Building and Training Model & Start Map Integration	Working on Location Data & Basic Simulation	Machine Learning Modeland beginning of map and integration
3rd Week in Feb to 4th Week in Feb	Refine Model & Dynamic Map	Location Data	Integrated Map
1st Week in March to 3rd Week in March (includes Spring Break)	Working Model Complete & begin Data Integration into Map	Basic Simulation for Better Attack Vector	Advanced model for prediction and integrated map
4th Week in March to 1st Week in April	Refine model for large scale predictions & refine Map integration	Estimate difference in Size Based on Better Attack Vector 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 into ML & usability testing, 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

6 Maintenance Considerations

The initial build of this project will not require much maintenance overall. This is a static build of the program and will not be a live service, but future goals for this project include a live service dashboard. Because of this, there will be a large portion of future maintenance considerations to understand.

Due to this service not being a live dashboard for our project scope, there will not be much maintenance on the data end. The reason for this is because our plan originally will be to gather all the data in many API calls and store the information in our own database. Once we have this built, the maintenance would just be to keep the database up and running with consistent backups. However, on future additions to this project, important maintenance would need to be taken on to make sure the data coming in from the APIs is clean and merges well with the already existing database.

Other maintenance that would need to be considered for the future goals of the project would be keeping the map and details up to date and working well with the new data as it comes in live. This also includes the machine learning model that needs to be kept in check and working. With new data being added into the database, the ML model would need to be checked and retrained to understand the new data that would then be present for it to learn on.

7 Project Management Considerations

- Meeting Information
    - 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 meetings:
        - Meetings with our sponsor are set on Wednesdays at 4:30pm PST.
        - Meetings with our advisor are set on Mondays at 6:00pm PST.
    - At the end of every sprint, we plan to update our advisor, DAB member, and sponsors with emails of the work that was done. If necessary, meetings can be held when implementations are not clear to the respective parties.
- Work Splitting
    - 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 interface of the simulations.
            - Kole Davis
        - Back-End Development will involve data collection, data cleaning, and database management between the client and the server.
            - Vincent Do
            - Drew Bogdan
        - Full-Stack Development will involve both front- and back-end development of the simulators.
            - Kevin Dang
            - Claire Yegian
- Additional Tools
    - None as of now.

Team Member Bios

Kole Davis

EDUCATION

Bachelor of Art, Computer Science & Computational Thinking — **August 2020—May 2024**
Software Development Concentration — Gonzaga University
Art Concentration — 3.7 GPA

WORK EXPERIENCE

Microsoft Hololens Application Research — **April 2022—Present**

- I am actively involved in a Gonzaga University faculty-led research team that is developing augmented reality (AR) software for the Microsoft Hololens 2. I am currently spending my time developing and testing an application designed to scan the surrounding environment to detect and notify the wearer of any tripping hazards among other environmental notification systems.

The Spur Group, Inc. — **Jun 2020—Aug 2021**

- The Spur Group is an industry-leading Strategy and Technology Consulting Firm that focuses on Go-To-Market strategy revenue acceleration. Based in Bellevue, Washington, The Spur Group works with a broad array of companies, including SAP, Microsoft, Salesforce, VMware, Dell, Zoom, Visa, Google, Cisco, Capital One, and many others.

  Application Development & Data Analytics Intern (Remote)
  Worked on front-end website development projects (HTML, CSS, JavaScript). Created, debugged, and maintained Web Applications using frameworks (React, Bootstrap). Developed bots using Microsoft Power Automate. Worked with Microsoft Azure and Excel. Cleaned, analyzed, and presented data to fellow employees and clients.

PROJECTS

Gonzaga 2nd Annual Hackathon — **Oct 2022**

- Developed a Unity augmented reality project for HoloLens with three teammates.
- Attempted to make a game based on Pac-Man where the player collects items and avoids a yeti.
- Developed 3D assets in Blender and created scripts using those assets

Spokane Mayor's Cyber Cup III — **Feb 2022**

- Participated in this cyber security event with a team of four and tackled a plethora of capture the flag tasks from binary and assembly to Linux systems and SQL.
- Placed first in the lower division and third overall in the competition.

SKILLS

Java	JavaScript (jQuery, Node.js)	React.js
2D/3D Modeling	HTML & CSS	Git (GitHub)
C++ (gtest)	Unity/Unreal Engine	C#

OTHER INTERESTS

Games / Applications Development

- People commission me to create mods for them in Unreal Engine games, and I am actively working with both Unity and Unreal on personal projects.

Kevin Dang

EDUCATION

Bachelor of Science, Computer Science — **Aug. 2020—May 2024**
Software App Development Concentration — Gonzaga University
Mathematics Minor — 3.46 GPA

WORK EXPERIENCE

Software Developer Intern, Expeditors — **June—August 2023**

- Migrated and updated existing legacy utility to new codebase adding front-end web pages to help display discrepancies between databases from different servers for one of their applications.
- Created secured REST endpoints for communication between front and backend using Java, Spring Boot, and AngularJS while adding multiple stretch goal features after completing Minimum Viable Product.
- Worked with product owners to refine feature requirements and develop technical solutions utilizing SAFe workflow practices.

Research Assistant, GU EASL — **Oct. 2023—Present**

- Develop, Maintain, and Improve a hazard detection system for the HoloLens 2 utilizing C#, Unity, and Mixed Reality Toolkit (MRTK).

Teaching Assistant, Gonzaga University — **Oct—Dec 2022, Sept—Dec 2023**

- Grader for Fall 2022 Computer Science I and Fall 2023 Database Management Systems courses.

PROJECTS

Gonzaga 2nd Annual Hackathon — **Oct 2022**

- Developed a game in Unity utilizing the HoloLens 2 to develop an AR game based on Pac-Man where the player collects watermelons while avoiding a yeti.
- Investigated C# scripting in Unity, the Unity file structure, and the built-in properties with the Unity editor.
- Learned Build and Deployment to the HoloLens 2 and effective management of Unity repositories.

Spokane Mayor's Cyber Cup III — **Feb 2022**

- Collaborated in this cyber event with a team of four friends and solved many capture the flag tasks from binary, assembly, to Linux systems, and SQL.
- Our team places first in the Lower Division but third overall in the competition.

Space Guardians — **July 2023—Present**

- Developing a top-down shooter survival game in Unity where you play as a spaceship shooting alien ships with an arsenal of different weapons.
- Utilizing and learning Agile workflow, 2D spriting, and Unity scripting for a 2D game.

SKILLS

Java (Hadoop/Spring)	JavaScript/TypeScript	SQL (MySQL)
Python (pandas)	React/AngularJS	Git (GitHub/GitLab)
C++ (gtest)	NoSQL (MongoDB)	C# (Unity)

OTHER INTERESTS

- I mostly like to spend my time developing personal fun game projects, reading books, working out, and playing video games with friends.
- I love spending time and having fun with my family when I get the chance to go back home.

Drew Bogdan

EDUCATION

Bachelor of Science, Computer Science **September 2020—May 2024**
Concentrations: Data Science, Computer Security Gonzaga University
3.6 GPA

WORK EXPERIENCE

Data Engineer, BNBuilders **May 2023—August 2023**

- Completed projects under multiple departments for many different topics.
- Created a large SQL Server live sync to keep 3D building model data updated, and a PowerBI sheet with statistics of the building model data and how it changed over time.
- Created multiple machine learning models to help assist in predicting outcomes of bids and expected new employment required for specific projects.

Teaching Assistant, Gonzaga University **September 2021—Present**

- Teaching Assistant for 224 Software Development under Dr. Crandall class since sophomore year.

PROJECTS

Discord Music Bot **February 2022—Present**

- Created a bot on discord to download and playback audio from Youtube or Spotify sources into a discord voice channel.
- Added a custom queue system to manage the downloading of songs on a separate thread to have an almost seamless transition from one song to another.
- Started working on basic chat bot abilities to have basic conversations.

Gonzaga 2nd Annual Hackathon **October 2022**

- Worked with a group of 3 to create an Augmented Reality Pac Man Game with Unity on a Microsoft HoloLens
- Learned to work with C# Scripting, Unity, game objects, and objects spawning in a world.

Spokane Mayor's Cyber Cup III **February 2022**

- Competed in a local cyber security capture the flag competition with a team of four and completed tasks from binary patching to SQL injection, website scripting, and even social engineering.
- Achieved first place in the lower division but received enough points to be third overall.

Gonzaga 1st Annual Hackathon **October 2021**

- Made an Online Poker Application with a team of three
- Spent time making the packet structure, and the parser to send and receive game information across the network towards a central server.

SKILLS

Java (JavaFX, Swing)	C# (Azure)	Bash Scripting
Python (Tensorflow, Keras)	SQL (MySQL, SQLServer)	C
C++	Git (GitHub)	PHP

OTHER INTERESTS

I like to go on hikes, play hockey and work out a lot. My coding hobbies are making video games, small Discord bots, or web bots and doing some Hack the Box boxes. I also have been trying to write a book for the last 8 years, but the actual story has changed multiple times, but I'm finally on one I like and am working towards finishing. Recently I have gotten into cooking and have been trying all sorts of different dishes.

Claire Yegian

EDUCATION

Bachelor of Science, Computer Science **September 2020—May 2024**
Concentration: Data Science Gonzaga University
Minor: Spanish 4.0 GPA

WORK EXPERIENCE

Commerce Architects **June 2023—August 2023**

- Built the backend framework for a shared state, multi-user planning poker web application using AWS Lambdas, Gateway, and DynamoDB as well as Terraform for automated deployment.
- Collaborated with two other interns using agile for project management to take an app from concept to production.
- Worked closely with full-time developers to overcome blockers and learn new tools.

Web Designer, Gonzaga University **February 2022—Present**

- Made a website mockup from scratch using HTML, CSS, and JavaScript.
- Brainstormed and executed page layout and information distribution for the University site.

PROJECTS

Gonzaga University Complex Systems Research Group **February 2022—May 2022**

- Contributed to natural language processing research projects, one concerning translation mapping and another concerning orca whale call identification models.
- Collaborated weekly with a team of students and professors to monitor progress and brainstorm solutions for obstacles.

Data Science my-scikit-learn **September 2022—December 2022**

- Built a simplified scikit-learn imitation project in python with a table builder class and simple linear regression, k nearest neighbors, dummy, native bayes, decision tree, and random forest classifiers.
- Applied classifiers to a data science project on Spotify music popularity trends.

SKILLS

Java	AWS	Object Oriented Prog.
Python	Terraform	Git/GitHub/GitLab
C++	HTML/CSS/JavaScript	Agile Project Management
X86-64 Assembly	React	

OTHER INTERESTS

I'm a Trip Leader for GU Outdoors and love spending my free time backpacking, mountain biking, and skiing. I play for Gonzaga's club team as a goalie and am a Boston Bruins fan. I love to travel! I took a gap year to travel in the Himalayas and Patagonia and studied abroad in Spain. I love speaking Spanish and hope to live outside of the US for a while, maybe starting next year.

Vincent Do

EDUCATION

Bachelor of Science, Computer Science — **Sept 2020—May 2024**
Software Security Concentration — Gonzaga University
Mathematics Minor — 3.9 GPA

WORK EXPERIENCE

IT Specialist Intern at Trimlite — **June—August 2023**
- Developed tools for current product offering visualization and analysis using data exports from ERP software and helping maintain the BOM for production and sales.
- Created software for accounting to expedite manual entry of credit forms into bank processing.

Teaching Assistant at Gonzaga University — **Sept—Dec 2022, Sept—Dec 2023**
- Grader for Fall 2022 Data Structures and Fall 2023 Database Management Systems courses.

PROJECTS

Gonzaga 2nd Annual Hackathon — **Oct 2022**
- Developed a Unity augmented reality project for HoloLens with three teammates.
- Attempted to make a game based on Pac-Man where the player collects items and avoids a yeti.
- Learned how to work with the Unity scene editor, game objects, and C# scripts and experimented with spatial audio.

Spokane Mayor's Cyber Cup III — **Feb 2022**
- Participated in this cyber security event with a team of four and tackled a plethora of capture the flag tasks from binary and assembly to Linux systems and SQL.
- Placed first in the lower division and third overall in the competition.

Notepad± — **Oct—Dec 2021**
- Created a Notepad web application in the MERN tech stack for Web Development final project.
- Built features that allowed users to register and log into accounts; save, edit, and upload their text files; and change the web app theme.

Gonzaga 1st Annual Hackathon — **Oct 2021**
- Developed Poker in Java alongside two other teammates.
- Coded game logic (scoring, card & hand values, etc.) classes and managed GitHub branch setup.

Duplicity Game — **May 2020—June 2021**
- Part of a 3-person team developing a hobby multiplayer game project on the Roblox platform.
- Wrote custom chat, leaderboard, and related GUI logic scripts on a client-server model in Lua.
- Implemented features using server-to-client communications to prevent common game exploits.

SKILLS

Java (Hadoop)	JavaScript (jQuery, Node.js)	SQL (MySQL)
Python (pandas, PyTorch)	HTML & CSS	Git (GitHub)
C++ (gtest)	NoSQL (MongoDB)	C#

OTHER INTERESTS

Piano
- Received music lessons from 1st grade to high school.

Martial Arts
- Attended Vovinam classes from middle school to high school.
- Performed form demonstrations at community summer festivals.

Appendix

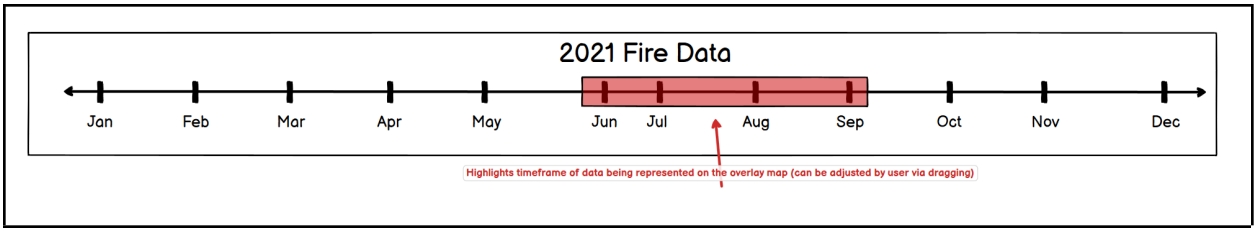

Figure 4: Map Timeline Component

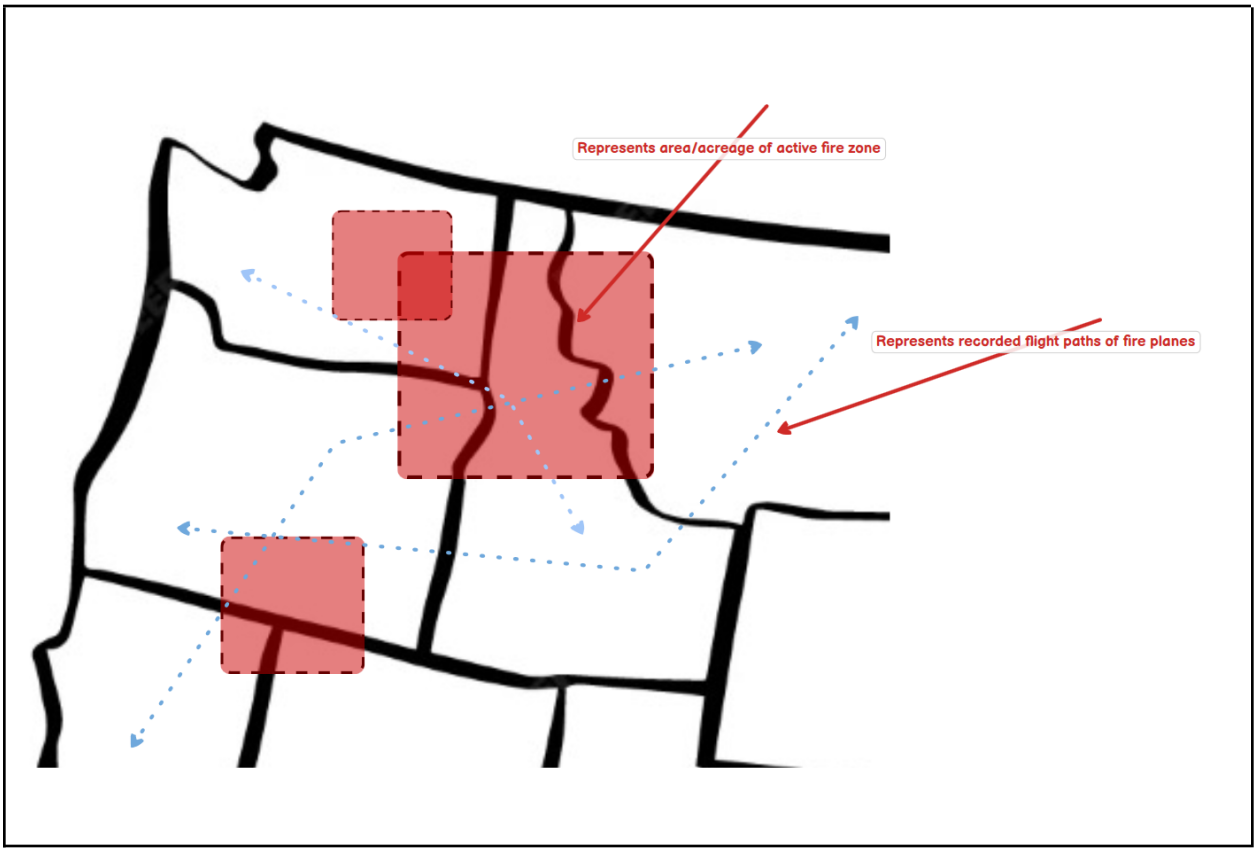

Figure 5: Map Data Overlay Functionality