Inferno Alert

A Project Report
Presented to
The Faculty of the Computer Engineering Department

San Jose State University
In Partial Fulfillment
Of the Requirements for the Degree
Bachelor of Science in Software Engineering

By
Deep Patel
Jay Solanki
Nimay Patel
Soumya Trivedi

April 13th, 2022

Advisor: Professor Ahmed Banafa

APPROVED FOR THE COLLEGE OF ENGINEERING

DocuSigned by:
Prof. Ahmed Banafa
5945D5EC752C447...

Professor Ahmed Banafa, Project Advisor

Dr. Wencen Wu, Instructor

Dr. Rod Fatoohi, Computer Engineering Department Chair

ABSTRACT

Inferno Alert

By Deep Patel, Jay Solanki, Nimay Patel, Soumya Trivedi

The United States has witnessed more than 1.8 billion acres of land getting burnt this year alone and the number is suspected to increase (CNN). These fires gulp the vegetation disturbing the ecological balance in nature and destroying everything along the way. Predicticting such fires beforehand will not only reduce the cost to combat these fires but also save thousands of lives.

Existing fire alert applications don't alert people until the fire is ablaze. They don’t predict future instances mainly due to the fear of the loss of business in the tourism sector. Businesses will suffer a loss if such alerts are provided in advance; As in the case of Lake Tahoe: Visitors were unaware about the susceptibility of a wildfire breakout and restaurants were afraid that they would lose money if they told visitors not to come..

The project aims to build up on the existing fire detection models and datasets to fabricate an application that alerts people nearing a fire risk zone. This alert might save billions of dollars and livestock. The project takes into consideration the history of that place and the environmental factors such as: wind, FFMC, temperature, DMC etc. to make as accurate a prediction as possible displaying its results on a web application.

Acknowledgments

We would like to express our gratitude to our project advisor, Professor Ahmed Banafa, who guided us throughout this project and to our Project Class Instructor Professor Wencen Wu for providing us the necessary material for the completion of this project.

.

Table of Contents

Chapter 1 Introduction
1.1 Project Goals and Objectives
1.2 Problem and Motivation
1.3 Project Application and Impact
1.4 Project Results and Deliverables
1.5 Project Report Structure
Chapter 2 Background and Related Work
2.1 Background and Used Technologies
2.2 Literature Survey
2.3 State-of-the-art Summary
Chapter 3 Project Requirements
3.1 Domain and Business Requirements
3.2 System (or Component) Functional Requirements
3.3 Non-functional Requirements
3.4 Context and Interface Requirements
3.5 Technology and Resource Requirements
Chapter 4 System Design
4.1 Architecture Design
4.2 Interface and Component Design
4.3 Structure and Logic Design
4.4 Design Constraints, Problems, Trade-offs, and Solutions
Chapter 5 System Implementation
5.1 Implementation Overview
5.2 Implementation of Developed Solutions
5.3. Implementation Problems, Challenges, and Lessons Learned
Chapter 6 Tools and Standards
6.1 Tools Used
6.2 Standards Used
Chapter 7 Testing and Experiment
7.1 Testing and Experiment Scope
7.2 Testing and Experiment Approach
7.3. Testing and Experiment Results and Analysis
Chapter 8 Conclusion and Future Work
8.1 Conclusion
8.2 Future Work

References

List of Figures

Figure 1. Process Decomposition Diagram Page No. 08
Figure 2. Process Summary Diagram Page No. 09
Figure 3. Domain Class Diagram Page No. 11
Figure 4. State Machine Diagram Page No. 11
Figure 5. Architecture Design Diagram Page No. 15
Figure 6. Interface and Component Design Diagram Page No. 17
Figure 7. Structure and Logic Design Diagram Page No. 18
Figure 8. Software Development Process Diagram Page No. 24
Figure 9. Test case lifecycle diagram Page No. 26
Figure 10. Scenario Diagram for testing process Page No. 27
Figure 11. Implementation Diagram for testing process Page No. 28
Figure 12. Test Result Summary Page No. 29
Figure 13. Test Cases Completion Status Page No. 30

List of Tables

Table 1. Functional Requirements Page No. 12
Table 2. Non-Functional Requirements Page No. 13
Table 3. Technology & Resource Requirements Page No. 14
Table 4. Sample table of Test case Design Page No. 27
Table 5. Cost of Software Testing Page No. 29

Chapter 1. Introduction: - (Contributor: Jay Solanki)

1.1 Project Goals and Objectives

(1) Project Goals: -

- Our goal is to develop a machine learning algorithm that can make predictions using non-confidential climate data that is available via non-profit research organizations.
- We plan to build up on the existing fire detection models and datasets to fabricate a web application that allows people to view the possibility of fire in specific areas selected by them.
- Our goal is also to help people save their own lives by making an early evacuation.
- As such, we intend to make this web application accessible to both web browsers and mobile applications and keep it free of cost to use.

(2) Project Objectives: -

- Our objective is to reduce the cost to combat wildfires and to save thousands of lives by letting people know in advance the places with high fire risk that they might want to avoid during wildfire season.
- As we all know, for firefighters, it is a two pronged approach to deal with fires: One is to keep fire under control and other is to evacuate people. Hence, it is also our purpose to help firefighters and local authorities in early evacuations and relocations by informing civilians of the danger they are in.
- It is also our objective to make sure that they can only access the prediction results on our User interface and we do not, under any circumstances, plan to make any climate data in our dataset accessible to our end users.

1.2 Problem and Motivation

The problem we are dealing with is wildfire. Wildfires destroy millions of hectares of terrestrial ecosystems annually and have led to a substantial increase in ecological and economic costs over the recent years [5]. This year the fires in California totaled a whopping 7,758 [13] fires burning enormous acres of land let alone the destruction caused to life, property, and economy. Our team is motivated to tackle the issue of destruction caused by wildfires thus we came up with our project: Inferno Alert. Our project predicts the incoming wildfires and thereby alerting people in that area. The project aims to save wildlife and lives of people who unknowingly fall prey to the flames. We are in the process of addressing this problem and are on route to creating a web application that predicts wildfires. Another problem that we have addressed is learning the required skills for this project. Additionally, we are already done with finding and cleaning the dataset. We have also made progress on the UI of the web application.

1.3 Project Application and Impact

Our initiative/project warns citizens and firefighters ahead of time so that they can prepare for wildfires and cause less damage to society. With Inferno Alert, we can limit the spread of flames and the problems arising from it. Our project's goal is to make individuals aware of their surroundings and inform them when necessary to prevent any mishap.

The impact on society, industry is huge since the project notifies for a probable fire beforehand which results in saving livestock, infrastructure and lives of people. Academically there can be research done to increase the accuracy of our prediction model. Our project gives to academics for research and reference.

1.4 Project Results and Deliverables

Project Results:

Upon completion of the project, the end-user should be able to predict the wildfire beforehand, taking into account the user-entered coordinates (Latitude and Longitude) of a location, which should be compatible with any browser. Completing the Inferno Alert project will provide the user the option to enter the coordinates. The possibility of fire for those coordinates will then be displayed to the user on a UI map.

Project Deliverables:

- Deliverable 1: This includes the activities and procedures we did over the week, as well as the document's weekly reporting. We create a list of what we've done so far, what we still need to do, and what the next steps are. Completing the assigned tasks on Jira software and scheduling the bi-weekly group meetings with the project advisor to meet the project deadlines.

- Deliverable 2: We have already made the different types of Project prototypes in the project life cycle by using Adobe XD to test the website's structure, theories, and ideas regarding Web Application Layout and gather user feedback based on the different prototypes models through usability testing to improve the final product.
- Deliverable 3: After the project prototype was done we started the coding part where we chose a precise and accurate wildfire dataset with proper independent variables. We are currently working on choosing a Machine learning Model which can predict the wildfire most accurately. We have also started working on User Interface Components using frontend technologies.

1.5 Project Report Structure

The Introduction of this report is followed by the Background and Related Work section. This section talks about the background of this project. It includes works that are related to this project and also provides the literature search and state-of-the-art summary. The third section describes the project requirements. This includes the functional and non-functional requirements. It also includes the software requirements for the end user. Chapter four is about the system design. This chapter has three UML diagrams that explain the design of the system in detail. These diagrams were used as references when developing the application. Finally, chapter five is about the system implementations. This provides a description of the technologies that we are using and the challenges that we have faced so far in the implementation phase of the project.

Chapter 2 Background and Related Work (Contributor: Jay Solanki)

2.1 Background and Used Technologies

The world sees several wildfires each year. These wildfires are extremely dangerous and cause a lot of harm. For people who live in states such as California, wildfires are a major source of anxiety every wildfire season. As students of San Jose State, we live close to the devastation caused by these wildfires and we see people who have to suffer their consequences every year. Hence, we decided to build a machine learning web application that can predict wildfires. The process of building this web application can be divided into several parts such as documentation, dataset searching, data cleaning, data analyzing, and programming. The documentation for this project included deciding on the project architecture and design. Our project architecture includes four different layers which are the presentation, business, service, and data layers. The processes of dataset searching, data cleaning, and data analyzing has already been done. The programming is being done asynchronously where we work on the UI and machine learning model simultaneously. We will be using an object oriented programming model and the programming language used will be Python. This project will make use of several different machine learning approaches. The courses we have taken that we applied to this project include: Data Structure and Algorithms (CS 146), Object-Oriented Design (CS 151), Software Engineering I and II (CMPE 131 and CMPE 133), Machine Learning for Big Data (CMPE 188), and Software Quality Engineering (CMPE 187).

This project makes use of several different technologies and resources which can be divided into different types based on their functionality. Jira ticketing system, which is used for tracking issues and project management is used to divide the work among the group members and keep a track of who's working on what issue. Websites such as Google Datasets were used to find the appropriate datasets. The technologies that are being used to build the machine learning web application include Jupyter Notebook, Google Maps JavaScript API, Django Framework, REACT, and Bootstrap. Jupyter Notebook is being used on the Service Layer of our application in order to create the machine learning algorithm whereas Django Framework is being used on the Business Layer to develop API. The UI mockups were made using Adobe XD whereas tools such as REACT, Bootstrap, Google Maps JavaScript API are being used to develop the Presentation Layer in order to display the results to the end users.

2.2 Literature Search

Literature Search

Every year there are several wildfires that wreak havoc across the globe. Wildfires are a major threat to life, property, and the environment. In most cases wildfires end up becoming out of control because they are not prevented on time. The purpose of this project is to use machine learning to predict the possibility of wildfires at certain places. This literature search discusses the various factors affecting wildfires and the ways in which Machine Learning can be used to prevent them. Most of the texts that are used in this search are peer reviewed articles from reputable sources such as the Fire Safety Journal. Additionally, a conference proceeding available on the IEEE website is also used. These texts were found using the SJSU Library website and Google Scholar. Several filters and search tips were applied on both the websites to narrow down the relevant texts. The decision to include the relevant texts in this literature search was made after reading the abstracts of the selected texts.

There are various Machine Learning approaches that are used in wildfire science where each approach has its advantages and limitations (Jain et al., 2020, p. 478). These approaches and their advantages and limitations are discussed in detail in this literature search. Additionally, there are several factors that affect wildfires and depending on which factors the machine learning models take into account, the results obtained from the models can differ. These factors are based on vegetation, climate and location features (Lall et al., 2016, p. 1).

Impacts of Wildfires

Wildfires have several negative impacts on the natural environment, economy and health of human beings (Sayad et al., 2019, p. 130). They lead to the destruction of thousands of square kilometers of forest every year (Sayad et al., 2019, p. 130). Wildfires cause air and water pollution which harms the wildlife of the area. Furthermore, they cause property damage and many people are displaced because of them. The vegetation of an area is impacted by forest fires which is not ideal for the forest ecosystem since forests play a key role in protecting against natural disasters such as landslides (Ghorbanzadeh et al., 2019, p. 43).

Factors Affecting Wildfires

There are several factors that can be studied to better understand the origins and development of wildfires. Such factors include abiotic, biotic, and human factors (Cardille et al., 2001, p. 111). The roles played by abiotic, biotic, and human factors in determining the spatial patterns of the origins of wildfires over a region were first studied by conducting research on more than 18,000 cases of wildfires in forested regions of Minnesota, Wisconsin, and Michigan (Cardille et al., 2001, p. 111). These states were selected for this study because they have diverse land cover, soil type, etc. (Cardille et al., 2001, p. 111). The factors that affect a wildfire can be found for a different region using the methods stated in this study (Cardille et al., 2001, p. 124-125). This study provided a platform for other useful studies done in the future. Some of those studies will be used by us in this project. Current Land Cover which is a biotic factor talks about how land cover influences wildfires by influencing fire development and spread through variations in fuel abundance and connectivity (Cardille et al., 2001, p. 114). Abiotic factors that might affect wildfires include Available Water Capacity (AWC), wind conditions etc. (Cardille et al., 2001, p. 114-116). Similarly, human factors include Rail Density, Road Density, Population Density etc. (Cardille et al., 2001, p. 116-117). Even though there are several factors that affect wildfires, it is not possible to use all of them to create a Machine Learning model. Therefore, we only use specific factors when creating a Machine Learning model.

Machine Learning and Wildfires

Over the years the use of machine learning in wildfire science has increased. As the field of machine learning continues to expand, its applications in wildfire science continue to expand as well (Jain et al., 2020, p. 478). There are six major problem domains that can be studied by machine learning in wildfire science (Jain et al., 2020, p. 478). These problem domains include fire detection and mapping, fire weather and climate change, fire susceptibility and risk, fire behavior prediction, fire effects, and fire management (Jain et al., 2020, p. 478). One of the ways in which spatial prediction of wildfire susceptibility can be done is by applying various machine learning models on certain datasets (Ghorbanzadeh et al., 2019, p. 43). The accuracy and results of these models depends on the datasets that they are applied to. Additionally, the accuracy of each model is different. Therefore, it is essential to find the appropriate dataset and model when using this approach to predict the susceptibility and risk of fires.

2.3 State-of-the-art Summary

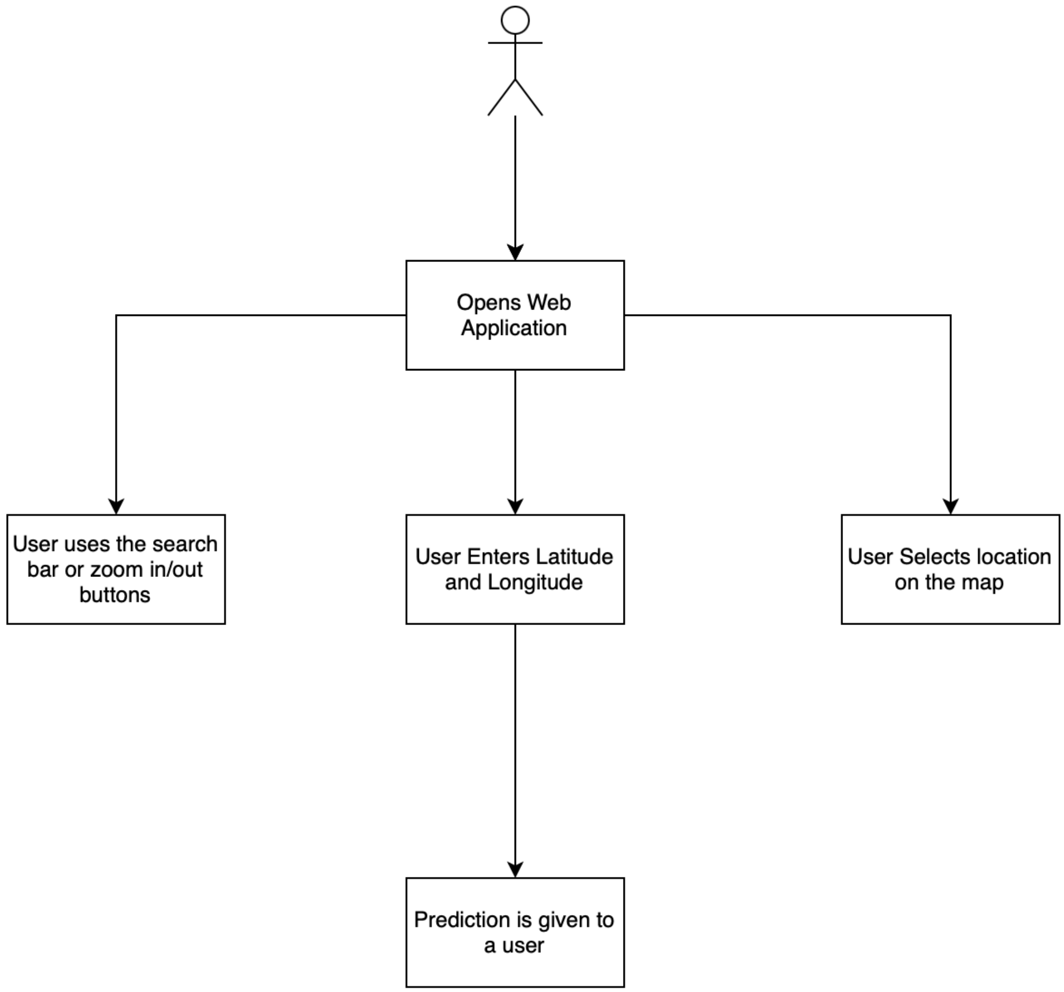
Wildfires are hard to predict, hard to extinguish and cause a lot of damage [12]. One of the reasons that wildfires cause a lot of damage each year is because they are not stopped on time and become out of control. Additionally, people are not evacuated on time which leads to messy evacuations that pose a threat to lives. All of this happens because the conventional methods used to predict wildfires are not accurate. Therefore, it becomes extremely difficult for the authorities to act in advance to prevent the wildfires. Recently, leading-edge Artificial Intelligence techniques like Big Data and Machine Learning are being used to predict and prevent them [12]. Big Data refers to a collection of large and continuously growing datasets that are not structured [12]. Machine Learning involves detecting patterns in data and using them to predict future outcomes [6]. This is done by applying various machine learning approaches. Some of the machine learning approaches that can be applied are artificial neural networks (ANN), support vector machines (SVM), and random forest (RF) [4]. While ANN is a slightly older machine learning approach to predict wildfires, RF is the most recent and popular machine learning approach [6]. While ANN uses a set of interconnected nodes to imitate the human brain, RF merges numerous decision trees for classification of the input datasets [4]. Additionally, some machine learning algorithms that are being applied in this project include logistic regression and decision tree.

The information that is generally available regarding wildfires is in the form of maps that show the areas where a wildfire is ablaze. These maps are not as useful because they do not predict wildfires. On the other hand, the information using the above mentioned leading-edge techniques will be very beneficial to help create the web application that predicts wildfire. The dataset for the creation of this application was found using websites such as Google Datasets. Datasets such as those found on Google Datasets are created using data from different sources and instruments. Moderate Resolution Imaging Spectroradiometer (MODIS) is one such instrument that is aboard the Terra and Aqua satellites [12].

Chapter 3 Project Requirements: (Contributors: Nimay, Deep, Jay)

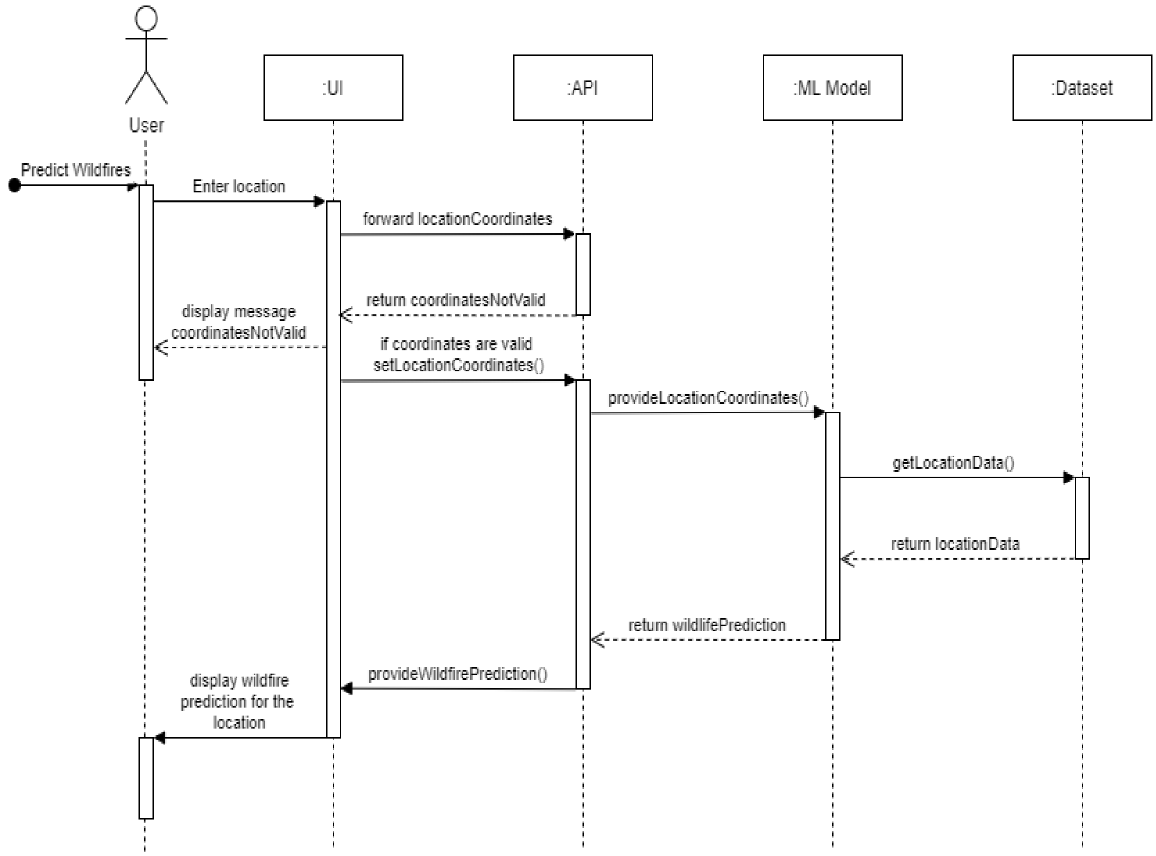
3.1 Domain and Business Requirements

Process Decomposition Diagram:



In the process Decomposition diagram above, on opening the Inferno alert web application, the user will get 3 options to predict fire. (1) First is where the User uses the search bar or zoom in/out buttons, (2) Second is when the user Enters Latitude and Longitude directly by inputting the values, (3) And finally, User can select location on the map by adjusting the pin directly on the map.

Process Summary Diagram:



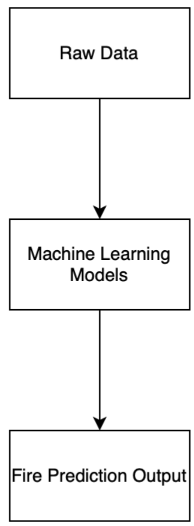
Process Summary Diagram

The process summary diagram shows the interactions between different objects to carry out a given scenario. This diagram helps identify the order in which certain tasks are carried out and the time taken to complete those tasks. The four objects in this sequence diagram are UI, API, Machine Learning Model, and Dataset. The scenario that needs to be carried out is the prediction of wildfires. The vertically oriented rectangles are called activation boxes and represent the time taken to complete a task. The solid arrow is the synchronous message symbol whereas the dotted arrow is the asynchronous return message symbol.

The scenario of predicting wildfires depicted in this diagram is carried out as follows:

- The user enters the coordinates of the location where the fire is to be predicted into the UI.

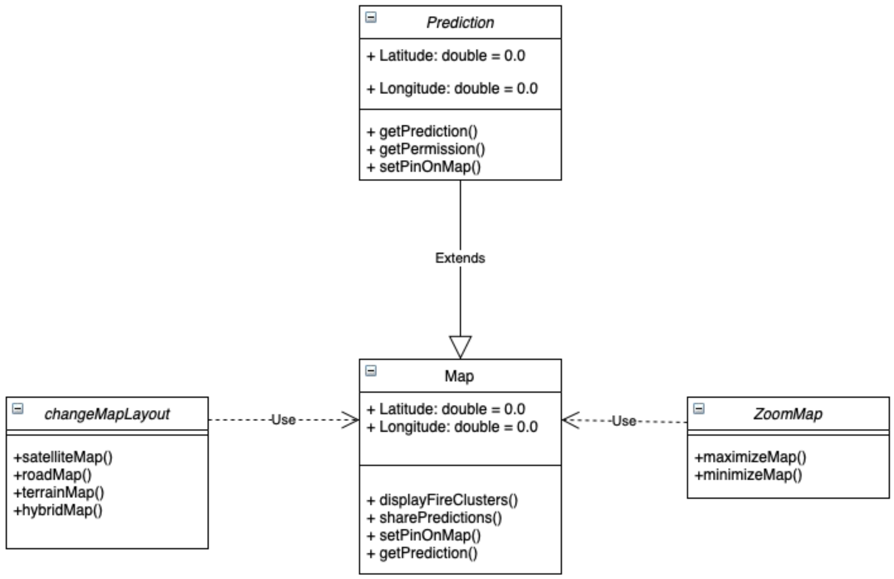
- These coordinates are forwarded to the API where the validity of the coordinates is checked. If the coordinates are valid they are forwarded to the ML model. If the coordinates are invalid, a message is displayed to the user on the UI.
- The ML model predicts the chances of wildfire at the given coordinates using the data accessed from the dataset.
- The wildfire prediction is forwarded to be displayed on the UI using the API.



Process Diagram from Raw Data to Final Output

In the process diagram above, the raw data is inputted in the machine learning code, where the data cleaning, data extraction, adjusting missing data, etc. is handled. We will use the accurate machine learning models in order to find the nearest possibilities, and if that matches the percentage of fire prediction is shown.

Domain Class Diagram:



Domain class diagram is a type of flowchart that illustrates the relationships between major components. We are creating a web application for Inferno Alert: Wildfire Alert using a machine learning model for this project. (1) Prediction, (2) Map, (3) changeMapLayout, and (4) zoomMap are the four domain classes we use. Prediction is the main class from which a user may enter their latitude and longitude, pin their location on the UIMap, or let the app use geolocationAPI to confirm their current location.

State Machine Diagram:

For this group project, we're developing a machine learning algorithm that makes predictions on chances of wildfire. Instead of having a state machine in a traditional sense, usually the dataset that consists of a combination of all data points in a single massive table is imported into a database format. An advantage of this is that it provides flexibility in terms of data modification. But a huge disadvantage of this approach is that it is notoriously slow and exponentially increases the execution time of the entire system thus degrading its performance. However, there is a better option available that could be implemented for this which is highly optimized and more efficient for this type of project. Therefore we, the project group 12, with consent from our advisor Prof Ahmed Banafa, have decided to not have this diagram for this project all together as this diagram seems to be a bit out of context with our project.

3.2 System (or Component) Functional Requirements

No.	Functional Requirements	Type
01	Enter Location - A Homepage should let the user input the values for latitude and longitude of a specific location.	Essential
02	View Output on Home Page - Once the latitude and longitudes are entered, the system should display a popup box that provides the user with a percentage value for the selected location. This percentage value should show the chances of occurrence of a wildfire.	Essential
03	Redirect Option - The user should have a button on the home page. Clicking this button should redirect to a Map UI.	Essential
04	View Output on Map - Once a location is selected the system provides the user with a percentage value for the selected location. This percentage value should show the chances of occurrence of a wildfire and is displayed as a pop up on the map.	Essential
05	Zoom In/ Zoom Out - The user shall have two buttons that shall allow the user to zoom in and out on the map.	Desirable
06	Search Bar - The user shall enter the names of certain locations in a search bar on the Map UI. Doing this shall narrow down the map area by zooming in on the location entered.	Desirable
07	Share this application - The user shall share the results of the wildfire using the application to other people in order to alert them beforehand.	Optional
08	Select Map Type - The system shall provide the user with an option to select the type of map.	Optional

3.3 Non-functional Requirements

No.	Non-Functional Requirements	Type
01	Performance - The system should run smoothly without any lag. The user can be redirected to the map UI within 3 seconds of clicking the redirect button 80 percent of the time as measured end-end	Essential
02	Security - Climate related data should be available at the service layer. It will not be viewable on the other layers.	Essential
03	Usability - The web application should be simple to use for the user. Additionally, the user interface will be intuitive, organized, and pleasant to look at.	Essential
04	Portability and Compatibility - This web application shall be able to run on all web browsers such as Google Chrome, Microsoft Edge, and Mozilla Firefox. It shall be accessible using these web browsers across all operating systems such as Windows, macOS, etc.	Desirable
05	Availability - The web application shall be available for the user to access 24 hours a day.	Desirable
06	Localization - The map UI shall be viewable in several different languages.	Optional

3.4 Context and Interface Requirements

The context environments supporting the development, testing and deployment of our project is Jupyter Labs. The deployment environment is on a UI made on ReactJS. Initially we will import our dataset on Jupyter Notebook and apply machine learning models to it. After that we will display the results of these models on the user screen.

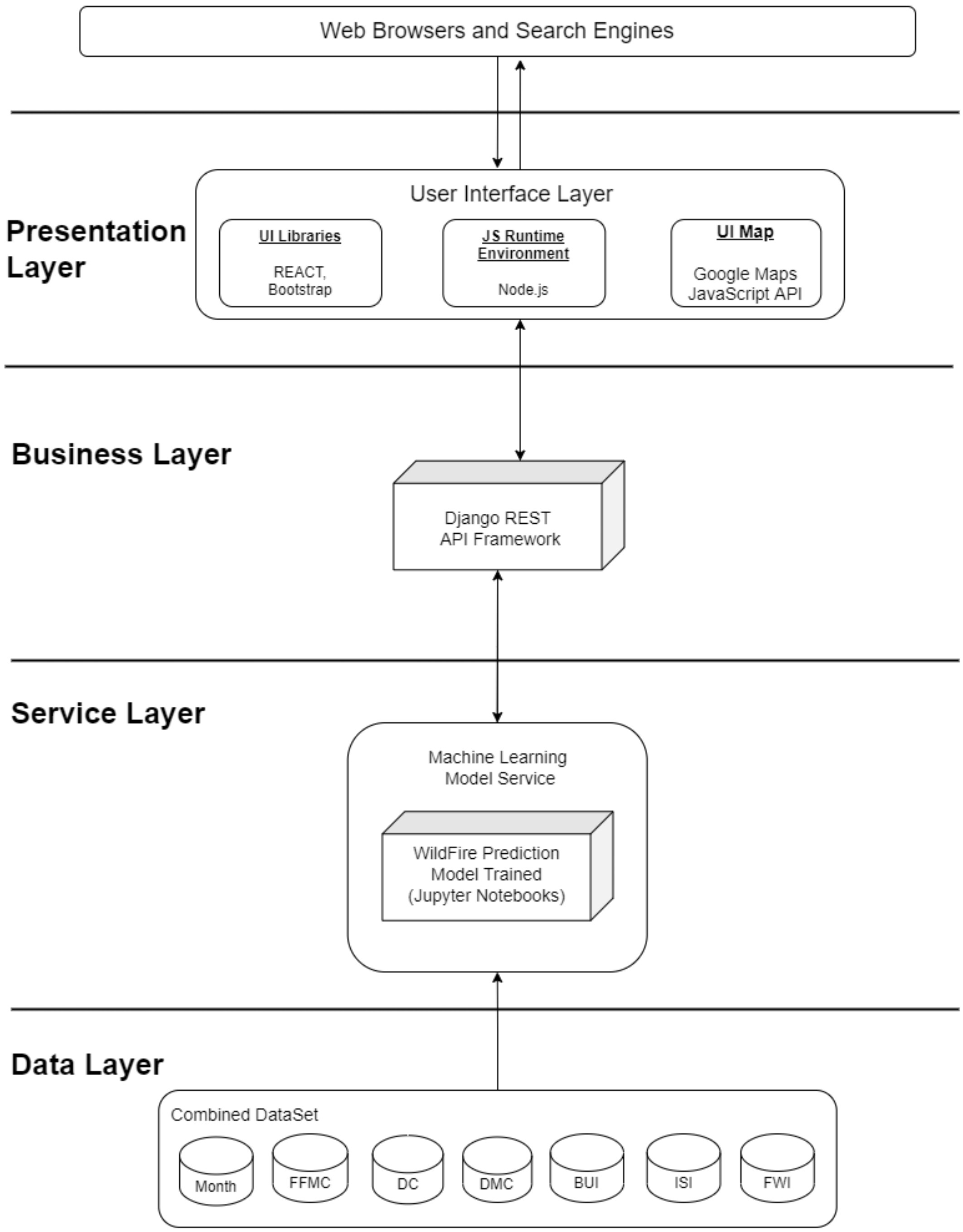
Our interface requirement is that our algorithm works on the Numpy, Scikit library and NLTK toolkit. These libraries help us in predicting the results we will display to users.

3.5 Technology and Resource Requirements

Software Requirements
Stable Internet Connection
A stable version of browser (Chrome 62.0 or above, Edge 40.0 or above, Safari 11 or above)
A stable O.S. (Android 10 or above, Windows 10 or above, MacOS 11 or above, iOS 13.7 or above)
Touch screen capability for interacting with dynamic map

Chapter 4 System Design: - (Contributor: Nimay Patel)

4.1 Architecture Design



Architecture Description:

For our web application architecture, we have decided to follow a Framework Layered Architecture model. In this model, the web application is basically divided into multiple layers depending on what type of tools will be utilized to develop each and every layer of the web application.

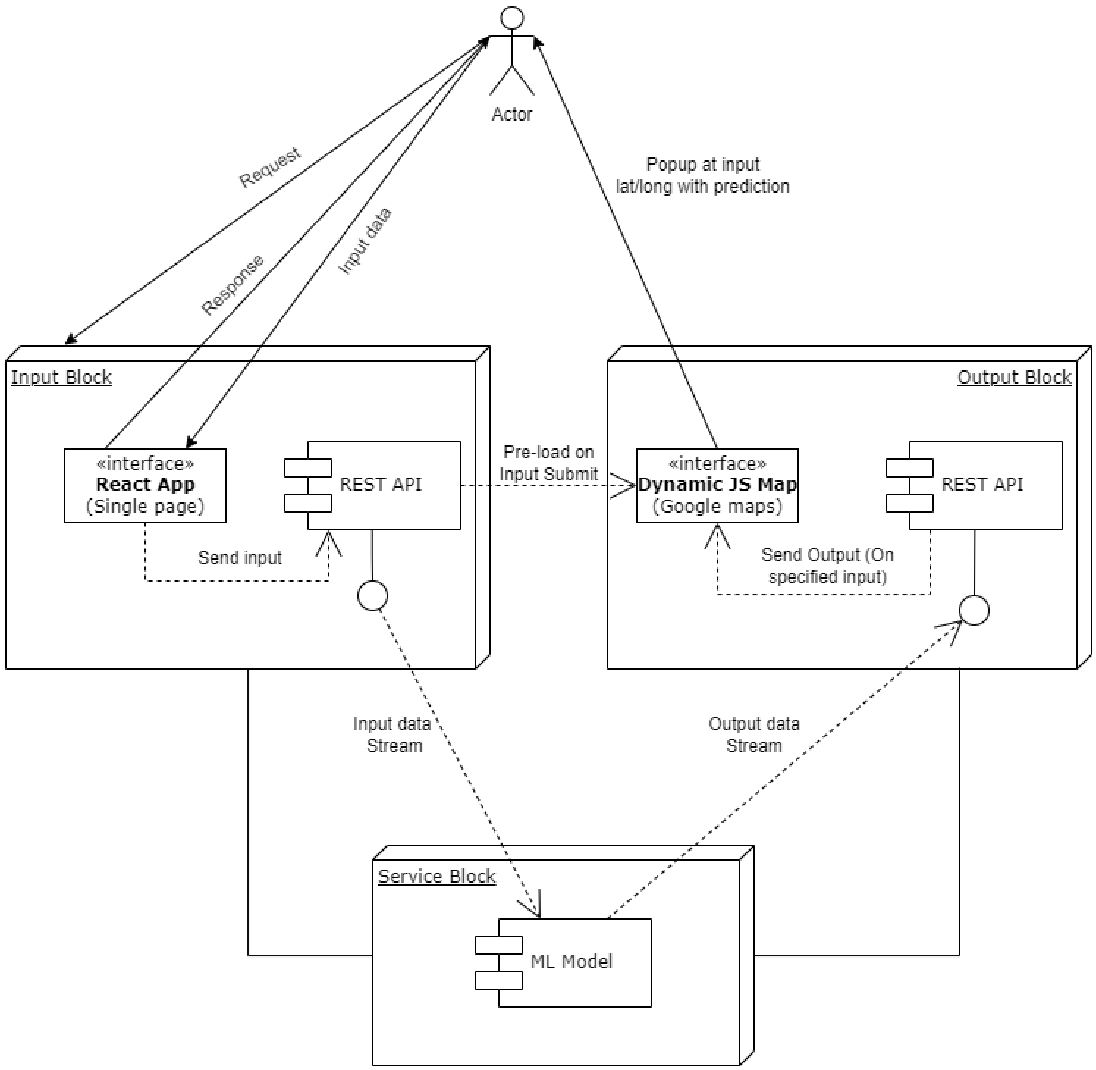
First, we have the User Interface layer at the top. This is the layer with which the user will interact and provide input for the application via web. This input includes the values of latitude and longitude. It is an essential requirement of our web application to ensure that this UI is compatible with web browsers on PC, mobile and ios. In our web application, the input can be made by user using one of the two methods:

(1) By typing in the values of latitude and longitude and other parameters using the text boxes that we provide using UI libraries such as REACT and Bootstrap.
(2) By clicking on a button, the prediction results shall be visible on the particular location on an interactive map that we provide using Google Map JavaScript API.

After this, the values are sent to the business layer of our web application via HTTP Get requests. Our business layer hosts a Web API developed using Django REST Framework and our service layer will house a machine learning model trained using Jupyter Notebook. This model will be trained to take in the values from the UI layer and make a prediction on what are the chances of wildfire happening on that location. We are testing multiple machine learning algorithms such as random forest classifier, decision tree, logistic regression and checking all of them for their precision and accuracy before selecting which one we will be training for the final version of our web application.

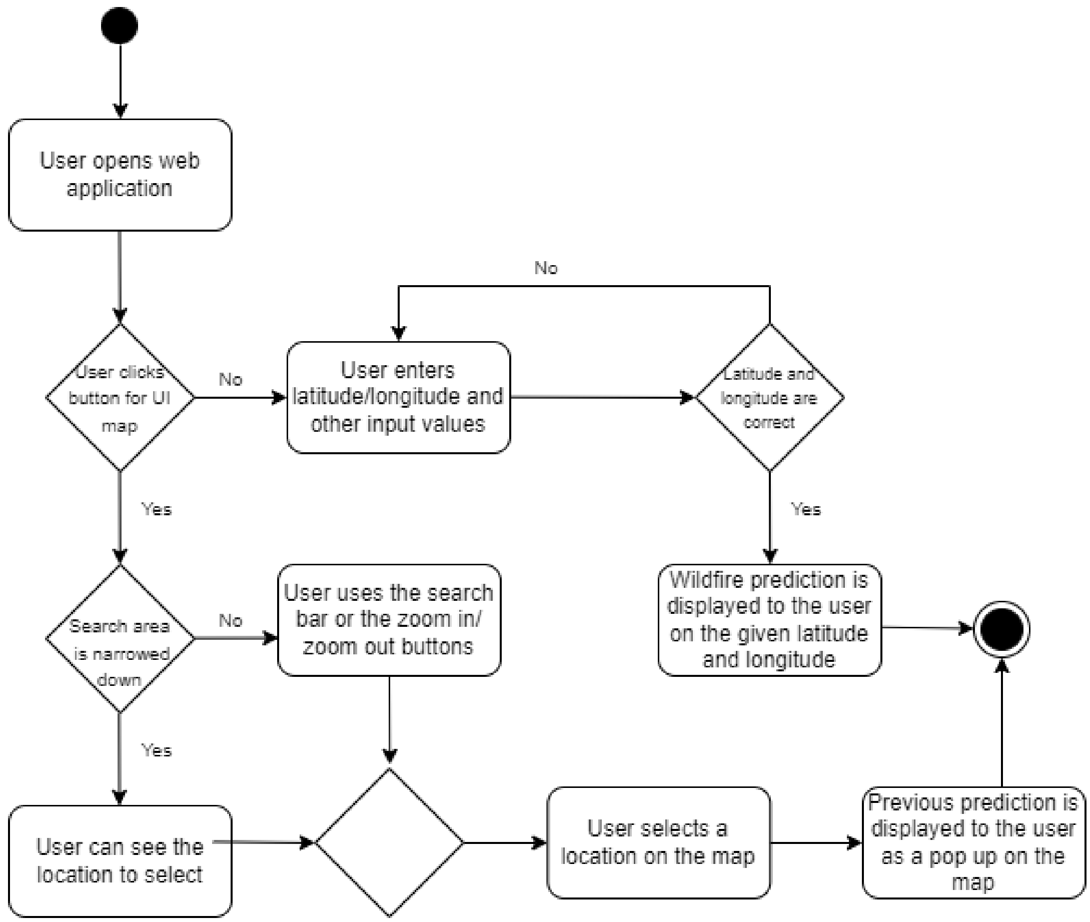
The values of environmental factors mentioned above will be imported by the machine learning model for training and testing purposes from the combined dataset that lies on the data layer of our application. After the system is deployed, it will use the input parameters to make the prediction based on what it learned training and the dataset. No user accessing the web application will either have access to this dataset or have access to any of the environment variables that this combined dataset contains. These values will only be used for training and testing purposes and will be obtained from a combined dataset which will contain columns such as wind, rain, humidity, temperature, FFMC,DMC, DC, BUI, ISI and FWI along with the month of the year in which this data was recorded and 'fire/not fire' classification.

4.2 Interface and Component Design



There are 3 main components of this system: Input, Output and Service blocks. When user requests to access web applications, the Input block is what responds by loading an interface to interact with. User uses this to input the information such as the latitude /longitude that they wish to make a prediction for along with some parameter values such as Buildup Index and Initial Spread Index. This input is forwarded to the service block along with redirecting users to the dynamic map where output will be displayed. Inside service block, the ML model will make a prediction and send it to output block where it will be displayed on the mentioned location by a popup box sent to user as final prediction result.

4.3 Structure and Logic Design



This diagram shows the structure and logic flow of our web application from a user standpoint. It shows the order in which activities should take place and the impact of choices made by the user. The solid dot is the start symbol whereas the ring represents the end. The activities are displayed in the rounded rectangles whereas the decisions are written in the diamonds. The empty diamond is used to merge the two different scenarios. The activities and decisions are connected using the arrows. Additionally, the choices made by the user are depicted by a Yes or No. The purpose of this diagram is to explain the structure of our system and the choice of logical paths that an end user can take. This diagram visually represents a series of actions or the flow of the control in a system. In a way, this is very similar to the flow chart or data flow diagram. The flow that the user follows in the system is decided based on what actions the user takes and what decision the user makes.

4.4 Design Constraints, Problems, Trade-offs, and Solutions

4.4.1 Design Constraints and Challenges

The most difficult aspect of designing our online application was gaining access to past fire datasets. The FWI Calculator that we used to calculate values of some of the parameters is only publicly available in Beta version as the main product continues to be classified as sensitive information by the various governments. The biggest challenge we face is to accurately forecast wildfires in any region. Because the dataset only contains 792 data points among which, a disproportionate amount of them are with fire classification, it might cause problems in predicting no fire instances which can lead to a waste of resources, time, and energy.

4.4.2 Design Solutions and Trade-offs

The trade-off is that if a wildfire is started by a person, machine learning algorithms may not be able to predict it in advance, resulting in an ineffective prediction. Also, the size of the dataset used for prediction is a source of concern for us. When it came down to selecting parameters required for making accurate predictions, basically, we had 2 options: (1) to go with a lot of parameters recorded and made available by weather reporting agencies or (2) to go with a few but more accurate parameters collected by organizations who specialize in research on wildfires. We chose option two because it gave us data that was small but had disproportionately high data points of active fire classification that that of no fire classification. Trade-offs were between choosing a large dataset of thousands of data points representing no fire throughout the year with only a handful of fire instances and choosing a small dataset with a disproportionate number of fire instances compared to no fire instances from agencies that specialize in the study of wildfires. Since, using a huge dataset, the possibilities of no fire situation skewing our data become too significant. The idea was, since wildfires only happen in certain climates and certain times of the year, the second option will be more accurate in training and testing out machine learning model.

Chapter 5 System Implementation

5.1 Implementation Overview

For this group project, we are developing a machine learning algorithm that makes predictions on chances of wildfire. Tools we use in this project include:

1. React.js (FrontEnd)
2. BootStrap library (FrontEnd)
3. Google Maps Javascript API (UI Maps)
4. Django REST API Framework (API Development)
5. Jupyter Notebooks (ML Algorithm)
6. Google Data Studio (Data Analytics)
7. Jira (Issue Tracking and Ticketing)
8. Selenium (Testing)
9. AWS (For deployment)
10. Whatsapp Group Calling (Team Meetings)
11. Zoom (Project Advisor Meetings)
12. Diagrams.net (UML Diagrams)
13. Lucid Chart (UI Mockups)
14. Google Drive (Documentation)
15. Github (Team Collaboration on coding)

The main coding language will be Python. We will utilize Jupyter notebook to run our code. Our project is heavily dependent on the dataset as only by accuracy in the data can we make accurate predictions.

5.2 Implementation of Developed Solutions

We are devising a machine learning algorithm that predicts the chance of wildfire. The prediction is based on a trained dataset which takes in Initial Spread Index(ISI), Buildup Index(BUI), Fire Weather Index(FWI) as its features. We will train 80 percent of our dataset and test on the rest 20 percent to get accurate predictions. The machine learning models we will implement on this dataset are SVM(support vector machines), Linear Regression. The next step for us is to compare the accuracy of these models. The application receives a request from UI(user-Interface) then internally predicts and displays back for FIRE or NO FIRE on the user's screen.

5.3 Implementation Problems, Challenges, and Lesson Learned

The challenges we faced were first finding a database which we could access and which included almost all the factors needed for fire prediction. Next hurdle we had to overcome was figuring out a way to split data as the dataset is skewed and an uncareful split can result in inaccurate prediction.

Our biggest concern is that if the reason behind wildfire is not natural i.e. what if the cause of fire was man made then the machine learning algorithm may not be able to anticipate the wildfire ahead of time, resulting in an ineffective prediction.

We learned a lot of lessons which included making roadmaps to complete tasks, implementing machine learning models on our dataset and documenting our work in an organized manner.

Chapter 6 Tools and Standards

6.1. Tools Used

Here is the list of all tools we are using for this project:

1) React (FrontEnd): We created a web application UI for manually putting values
2) Node.js (FrontEnd): Javascript runtime environment for executing JS code
3) Google Maps Javascript API (UI Maps): Integrating a dynamic map for user input
4) OpenWeatherMap API(UI MAps): Fetching weather data for the selected location
5) npm PackageRunner (FrontEnd): Package manager for Node.js code execution
6) VSCode (FrontEnd): Source code editor for Front end web app development
7) Google Colab(ML Model): Used as cloud-based Jupyter Notebook environment
8) Flask (API Development): Develop custom API for FWI calculation
9) Jupyter Notebooks (ML Algorithm): Create and share documents with live code
10) Google Data Studio (Dataset): Create data visualization from custom dataset
11) RStudio (Data Analytics): Creating custom dataset by merging subset data
12) Jira (Issue Tracking and Ticketing): Keep track of progress and responsibilities
13) Selenium (Testing): Used to automate web app testing activity once it is made
14) TestRail (Testing): To write test case, design test strategies and test management
15) Whatsapp Group Calling (Team Meetings): For team coordination in project work
16) Zoom (Project Advisor Meetings): For updating advisor with our progress
17) Diagrams.net (UML Diagrams): Design and develop software architecture
18) Lucid Chart (UI Mockups): Design and develop software architecture
19) Google Drive (Documentation): Document sharing and documentation purposes
20) Github (Software hosting): Sharing dataset and coding files and hosting web app

6.2. Standards

Here is the list of all ISO standards we are following for this project:

(1) ISO/IEC/IEEE 12207:2008 – Systems and software engineering – Software life cycle processes

(2) IEEE Std. 730-2014 (IEEE, 2014) presents requirements that cover all aspects of software quality assurance

(3) ISO/IEC 15445:2000 (ISO-HTML) International Standard for the core of the HyperText Markup Language.

(4) ISO/CD TS 32005 DOCUMENT MANAGEMENT — PORTABLE DOCUMENT FORMAT — PDF 1.7 AND 2.0 STRUCTURE

(5) ISO/TR 24156:2008 GUIDELINES FOR USING UML NOTATION IN TERMINOLOGY WORK

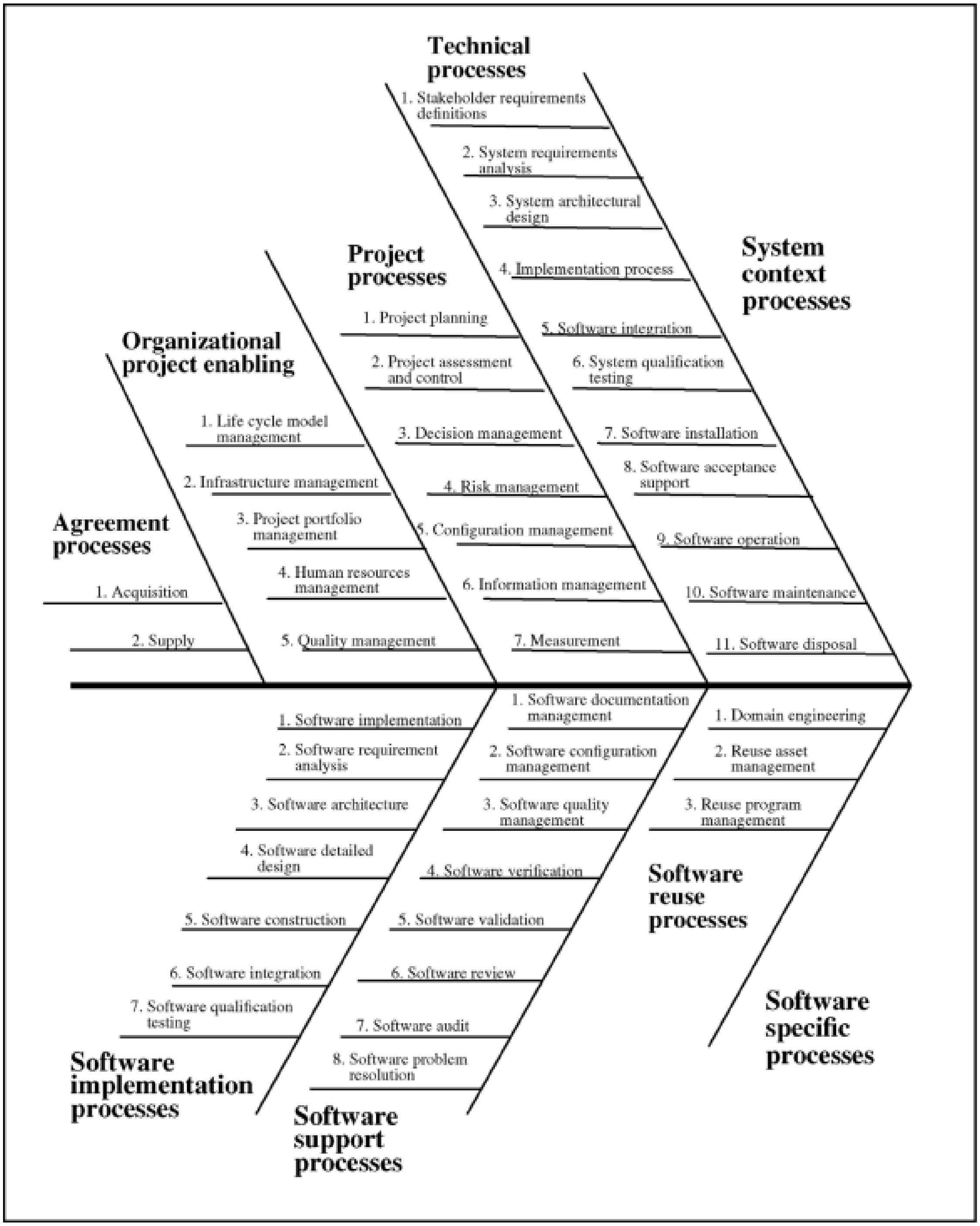
(6) ISO/IEC 23009: 2017 Series of ISO standards for HTTP and HTTPS

(7) ISO/IEC 29179:2012 INFORMATION TECHNOLOGY — MOBILE ITEM IDENTIFICATION AND MANAGEMENT — MOBILE AIDC APPLICATION PROGRAMMING INTERFACE

(8) ISO/IEC DTS 4213.2 INFORMATION TECHNOLOGY — ARTIFICIAL INTELLIGENCE — ASSESSMENT OF MACHINE LEARNING CLASSIFICATION PERFORMANCE

(9) ISO/IEC AWI 5259-3 ARTIFICIAL INTELLIGENCE — DATA QUALITY FOR ANALYTICS AND MACHINE LEARNING (ML) — PART 3: DATA QUALITY MANAGEMENT REQUIREMENTS AND GUIDELINES

(10) ISO/IEC AWI TS 12791 INFORMATION TECHNOLOGY — ARTIFICIAL INTELLIGENCE — TREATMENT OF UNWANTED BIAS IN CLASSIFICATION AND REGRESSION MACHINE LEARNING TASKS



Fishbone Diagram of entire Software Development Process

Chapter 7 Testing and Experiment

7.1 Testing and Experiment Scope

The main goal is to apply different Black-box and White-box Testing Methodologies for testing the functionality of the system. The overview of the test process is as follows:

- Use both manual and map input to test the prediction function
- Manual Input Testing
  (1) Input latitude and longitude through text box and make a call to Open Weather Map API.
  (2) Analyze weather observation data and calculate Fire Behavior Indices
  (3) Output a fire/no fire prediction using the fire Behavior Indices
- Map Input Testing
  (1) Click any location on map to provide system with latitude and longitude and an automatic call to Open Weather Map API will be made
  (2) Analyze weather observation data and calculate Fire behavior indices.
  (3) Output the fire/no fire prediction on the map in the form of popup box

Test Criteria for Unit Testing and Integration Testing:

We require that every team member will document the tests he or she will run, and have peers review those tests to ensure appropriate coverage. Also important is the establishment of group standards on what should be included in the unit tests. Following are the criteria that we came up with for our testing procedure:

(1) Functionality:
Each module must be tested to ensure it satisfies its design and actually does what it should do correctly. Questions like: What inputs must it handle? What things must it do? What services will it provide? What outputs should it produce? What data must it manage, and what must it do with that data? should be answered while conducting the tests. We must also ensure that the module actually does what it was intended to do.

(2) Negative Testing:
The module must handle all error conditions correctly before it is considered to be complete. This includes answering questions like: Does the module do the right thing when things go wrong? What happens when it is presented with invalid inputs? What if they are poorly formed or out of sequence? How about non-numeric data when numbers are expected?

(3) Coverage:
It's reasonable to require that every line of code be executed during unit tests. One step beyond mere code coverage, testing of every path in the code is reasonable as they are critical to assuring that the program will handle the situation as it was supposed to.

7.2 Testing and Experiment Approach

Test Strategy:

Test Strategy is also known as test approach defines how testing would be carried out. Test approach has two techniques that could have been used in our project:

- Proactive - An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.

- Reactive - An approach in which the testing is not started until after design and coding are completed. This is the approach that we decided to pursue in our project. The main reasoning is that since the concept we were working on is brand new and we didn't have any historic project to look for references, we were unsure of how the development would progress and what trade offs we might have to weigh in. As such, we found it to be beneficial if we start testing after we are done with the design and development phase. This way, we can better adjust our testing methods to our development.

Test Case Flow and Design:

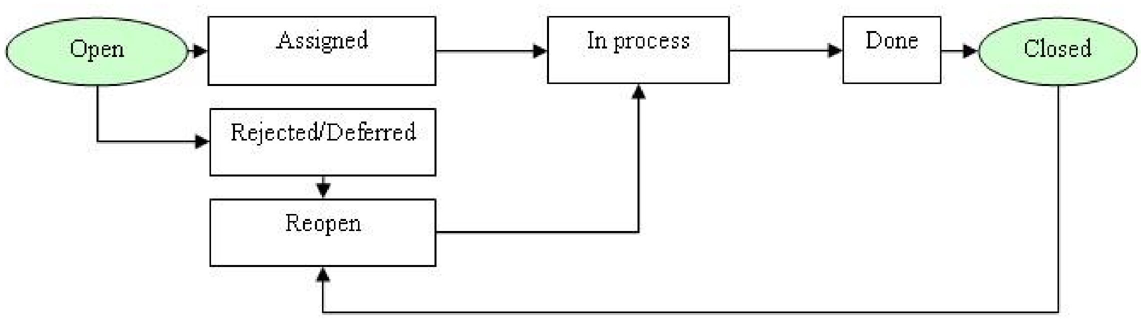


Figure: Lifecycle and flow of every test case

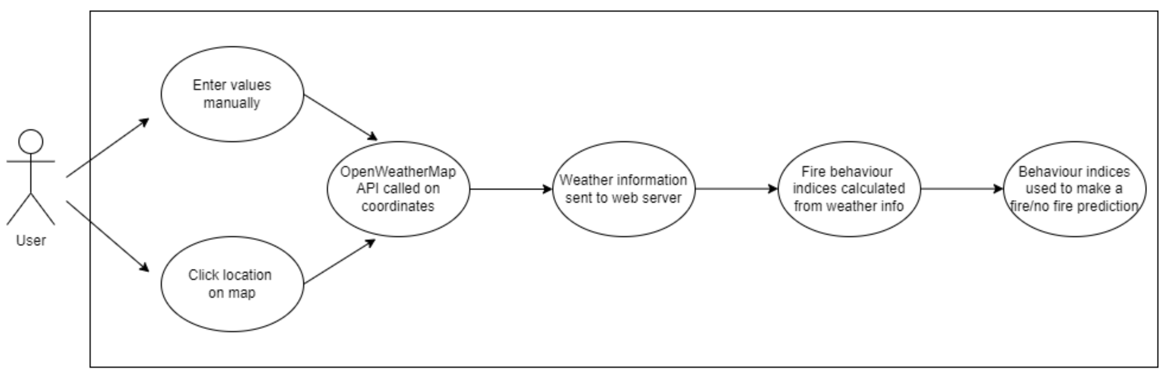
Every test case that is created will go through this lifecycle and it will continue to be reopened until all issues are resolved and all test cases are closed. That is when the testing process will finally end and the deployment process will begin. As of now, we are working on designing test cases for this purpose and testing will begin soon.

Test ID	**Sample_Specimen**
Test Description	**Sample_Specimen**
Test Case App	**Sample_Specimen**
Test Input	**Sample_Specimen**
Execution Date	**Sample_Specimen**
Expected Result	**Sample_Specimen**
Actual Result	**Sample_Specimen**
Test Output	**Sample_Specimen**

Table: Sample Test Case Design

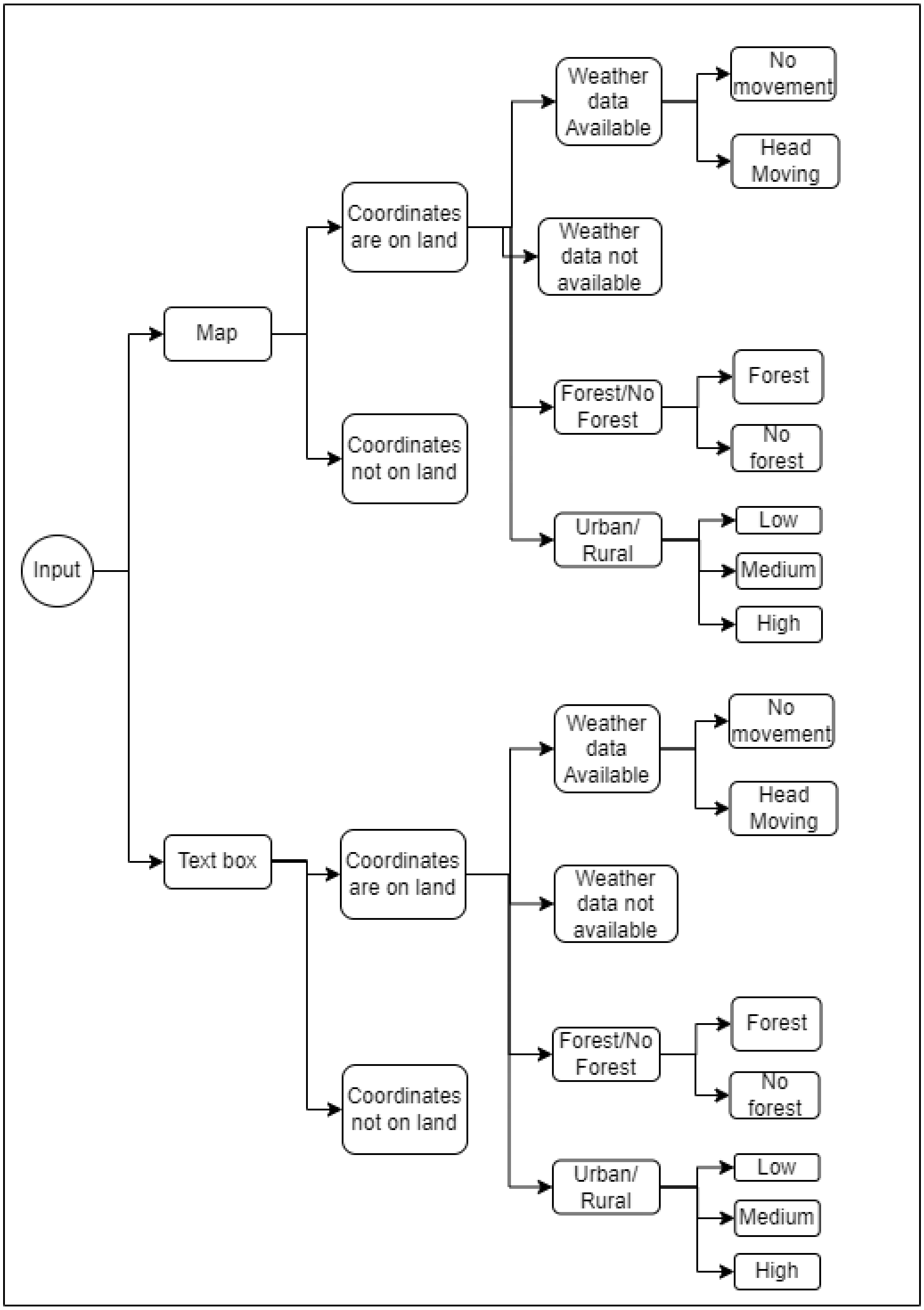
This table specifies the design and structure of individual test cases. All test cases will follow this design and implementation and will be managed using a software called TestRail and executed using a software called Selenium.

Scenario Diagram for testing:



The scenario diagram shows the stages and scenarios that we will be testing as a part of testing process. Every scenario has multiple test case to check for different pathways that a user can take.

Test Implementation Method and Technique:



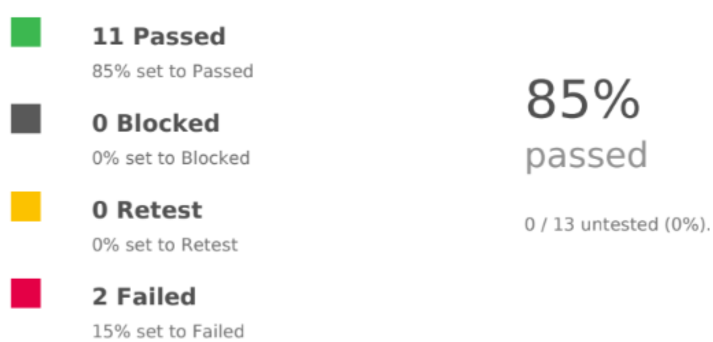
7.3 Testing and Experiment Results and Analysis

Cost of doing tests:

Function	Cost
Discussion	250 minutes
Test methods	200 minutes
Test case	150 minutes
Test Data	40 minutes
Test Analysis and Summary	50 minutes

Currently, we are in the test case design phase and as such, we haven't actually started the testing process as per our gantt chart. Our testing process is expected to start from April 15th and as such, we would be able to mention the test result summary and analysis along with a detailed bug analysis in the next iteration of report submission.

Test Result Summary:



Performance Test Results Analysis:

Following is the summary of our testing. Test cases were written using TestRail and testing was conducted using Selenium IDE WebDriver and BrowserStack Automate. Each of 13 cases are made based on 13 requirements set at the beginning. We are happy to report that all requirements marked critical and high priority have been fulfilled and 2 failed test cases were for low priority requirements. No critical component failed testing.

Test Execution:

For testing this web application,we first created 3 sections depending on our criteria in TestRail and wrote test cases (i.e. specifying preconditions, steps to follow and expected results). Based on these, we used Selenium IDE to write test scripts for all test cases. But execution of these scripts was a little complicated as our application is running on localhost, so we integrated our Selenium IDE project with BrowserStack Automate for this compatibility purpose. Then, we created TestRun in TestRail to execute these scripts and reported the results back into TestRail and generated a report for the final testrun.

Test Coverage:

The AI function test quality assessment is a way for developers to get feedback on their functions from an impartial party. Every combination of input and context category partition diagrams from the last section will be considered over here. Hence, each and every partition from that diagram which represents each and every test case in this document will be also tested at least once because the main method of testing that we chose was category partition testing and in this method we basically divide all the possible input and context from the domain of all existing values in this universe as known to mankind into N number of partitions and select one value from each input and one value from each and every context domain to create a world class quality of test case manufactured by the best engineers in the entire university.

Bug Analysis:

View Report

Component testing (5)

ID	Title	Final Test-Run
C2378	HomePage	Passed
C2379	Predict page	Passed
C2380	User data privacy	Passed
C2388	Map interaction	Passed
C2389	Dynamic Default values depending on location	Failed

Module Integration and Flexibility (5)

ID	Title	Final Test-Run
C2381	Connectivity among web pages	Passed
C2382	Test calls to OpenWeatherMap API	Passed
C2386	FireBehaviorIndices API	Passed
C2387	Multi platform accessibility	Passed
C2390	Application performance under different map types	Passed

Edge Case testing (3)

ID	Title	Final Test-Run
C2383	Running prediction on blank coordinate text boxes and no selections on map	Passed
C2384	Running prediction on invalid coordinates (out of range, coordinates on water bodies)	Failed
C2385	Clicking refresh multiple times	Passed

Generated with TestRail test management software – 7.5.1.7013

As visible from the above image, the 2 test cases that failed are:

(1) Dynamic Default values depending on location

It is my understanding that this failed due to a conflict in requirement specification. One of our requirements says that no user data will be collected for accessing this system. But unfortunately, the feature of dynamically setting default values based on user location required us to code in a mechanism that would popup the user requesting them to allow us to access their location. Since, user data privacy is much more valuable, that requirement is critical for our system and hence, we resolved the conflict by sacrificing the low priority requirement. Hence, the working system does not change default values depending on user location and instead stays zoomed in the south San Francisco Bay Area, precisely in San Jose and surrounding area because that is where this is developed. Users are, however, free to scroll through the global map and zoom in/out on any location of their choice.

(2) Running prediction on invalid coordinates

It is my understanding that this failed because coding in the mechanism to determine if coordinates are on land or sea proved much more complex than previously thought of. Also a major factor is the realization that dawned upon us because of the ongoing war in Ukraine is that, unlike geography, wars can happen anywhere and any time. This makes it virtually impossible for us to code anything in this matter. However, due to 95% or above accuracy of our algorithm, instead of displaying that prediction is not possible, it simply shows that fire is not possible which is technically true for the water bodies such as oceans, but would be inaccurate for war zones and areas where no data is available. Fortunately, this does not impact the overall working of the system as the actual requirement for this edge case was initially deemed as of medium priority.

Chapter 8 Conclusion and Future Work

8.1 Conclusion

With the increase in global warming, natural disasters like wildfire will no longer be a rare problem contained within one season. Instead, by looking at the current data, it is safe to say that worse days are yet to come. In such circumstances, it is difficult for civilians to know which areas to avoid while planning any trip. So we decided to make a web application that lets all customers enter any location and make a prediction on whether the risk of fire is there or not using the current weather conditions in the location. Our main focus, building this application is to help concerned authorities who try their best in minimizing loss of life and property to contain the wildfires by informing citizens to stay away from vulnerable areas. This application is built to provide services, not just in the USA but also for the entire world as customers can choose any land based location on the map. In pure geographic terms, instances of wildfires are limited to areas with dense forest covers in Latin America and Africa along with areas bordering the Ring of Fire which borders the notoriously unstable Pacific tectonic plate and goes from Australia upto Japan and Alaska on one side then coming down to Canada, USA, Mexico and latin America on the other side. This region is and has been extremely prone to volcano eruptions, earthquakes, tsunamis and wildfires and we believe that people living in these regions will be the largest benefactors from this project.

8.2 Future Work

Since this application is starting as a pilot project for such a concept, we believe that there is a lot of room for innovation and future upgrades. From our perspective, having a system where the data clustering is available on map would be very great for users. This way, the system automatically generates routine OpenWeatherMap API calls and stores the prediction results in a database which is displayed on the map in the form of clusters. That would be a proactive way of serving customers as they can find out the predictions, on map without entering any values and waiting for results. Also, currently, the system can only accept one pair of latitude and longitude at a time and display one result with becomes tedious when users need to check for a large area on map especially during hikes. Also, in a distant future, we believe that there is room for analyzing historic weather data precisely for these prediction clusters over a period of years and determine which locations are not suitable for human inhabitation.

REFERENCES: -

[1] A. Srinivasan, "Leveraging Machine Learning to predict wildfires: Contributing to the United Nations Sustainable…," *Medium*, Dec. 15, 2020. https://towardsdatascience.com/leveraging-machine-learning-to-predict-wildfires-contributing-to-the-united-nations-sustainable-a10c5044dcae?gi=8bb471cae7bb (accessed Nov. 11, 2021).

[2] Berwyn, B. (2018, August 23). *How wildfires can affect climate change (and vice versa)*. Inside Climate News. Retrieved October 13, 2021, from https://insideclimatenews.org/news/23082018/extreme-wildfires-climate-change-global-warming-air-pollution-fire-management-black-carbon-co2/?gclid=CjwKCAjwh5qLBhALEiwAioodswvm5t4B9zB9Y1DyXOL95K-c1jGadth4zRsfAOhmaadDA8eE0K-hpRoC6DcQAvD_BwE.

[3] Cardille, J. A., Ventura, S. J., & Turner, M. G. (2001). Environmental and Social Factors Influencing Wildfires in the Upper Midwest, United States. Ecological Applications, 11(1), 111–127. https://doi.org/10.1890/1051-0761(2001)011[0111:EASFIW]2.0.CO;2

[4] Ghorbanzadeh, O., Valizadeh Kamran, K., Blaschke, T., Aryal, J., Naboureh, A., Einali, J., & Bian, J. (2019). Spatial Prediction of Wildfire Susceptibility Using Field Survey GPS Data and Machine Learning Approaches. Fire (Basel, Switzerland), 2(3), 43–. https://doi.org/10.3390/fire2030043

[5] "Hybrid artificial intelligence models based on a neuro-fuzzy system and metaheuristic optimization algorithms for spatial prediction of wildfire probability," *Agricultural and Forest Meteorology*, vol. 266–267, pp. 198–207, Mar. 2019, doi: 10.1016/j.agrformet.2018.12.015.

[6] Jain, P., Coogan, S. C. ., Subramanian, S. G., Crowley, M., Taylor, S., & Flanigan, M. D. (2020). A review of machine learning application in wildfire science and management. Environmental Reviews, 28(4), 478–505. https://doi.org/10.1139/er-2020-0019

[7] J. Brownlee, "How to Connect Model Input Data With Predictions for Machine Learning," *Machine Learning Mastery*, Nov. 14, 2019.https://machinelearningmastery.com/how-to-connect-model-input-data-with-predictions-for-machine-learning/ (accessed Nov. 11, 2021).

[8] Kaustumbh Jaiswal, "Deploying a Machine Learning Model Using Django: Part-1," *Medium*, Jul. 03, 2019. https://medium.com/saarthi-ai/deploying-a-machine-learning-model-using-django-part-1-6c7de05c8d7.

[9] K. Nighania, "Various ways to evaluate a machine learning models performance," *Medium*, Jan. 30, 2019. https://towardsdatascience.com/various-ways-to-evaluate-a-machine-learning-models-performance-230449055f15.
[10] Lall, S., & Mathibela, B. (2016). The application of artificial neural networks for wildfire risk prediction. 2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA), 1–6. https://doi.org/10.1109/RAHA.2016.7931880

[11] *Lawson, B.D.; Armitage, O.B. 2008.* ***Weather Guide for the Canadian Forest Fire Danger Rating System****. Natural Resources Canada, Canadian Forest Service, Northern Forestry Center, Edmonton, AB. 84 p.*

[12] "Overview | Maps JavaScript API," *Google Developers*. https://developers.google.com/maps/documentation/javascript/overview.

[13] Sayad, Y. O., Mousannif, H., & Al Moatassime, H. (2019). Predictive modeling of wildfires: A new dataset and machine learning approach. Fire Safety Journal, 104, 130–146. https://doi.org/10.1016/j.firesaf.2019.01.006

[14] S. E. - Artemis.bm, "California wildfires in 2021 have destroyed or damaged 3,500 structures - Artemis.bm," *Artemis.bm - The Catastrophe Bond, Insurance Linked Securities & Investment, Reinsurance Capital, Alternative Risk Transfer and Weather Risk Management site*, Oct. 01, 2021.https://www.artemis.bm/news/california-wildfires-in-2021-have-destroyed-or-damaged-3500-structures/.

[15] *The economic impacts of large wildfires | ecosystem* . (n.d.). Retrieved October 13, 2021, from https://ewp.uoregon.edu/largefires/content.

[16] *Turner, J.A.; Lawson, B.D. 1978.* ***Weather in the Canadian Forest Fire Danger Rating System. A user guide to national standards and practices****. Environment Canada, Pacific Forest Research Centre, Victoria, BC. BC-X-177.*

[17] *Van Wagner, C.E.; Pickett, T.L. 1985.* ***Equations and FORTRAN program for the Canadian Forest Fire Weather Index System****. Canadian Forest Service, Ottawa, ON. Forestry Technical Report 33.*