Software Requirements Specification

For

**Weather Forecast and Disaster Alarm in Hilly Areas**

Prepared by

|  |  |  |
| --- | --- | --- |
| **Specialization** | **SAP ID** | **Name** |
| Btech. CSE (CCVT) | 500075265 | Anirudh Kushwah |
| Btech. CSE (CCVT) | 500076356 | Gaurav Porwal |
| Btech. CSE (CCVT) | 500077245 | Garvit Pant |
| Btech. CSE (CCVT) | 500077079 | Kumar Baibhav |

Under the guidance of:

**Dr. Jagdish Chandra Patni**

**Associate Professor**



**School of Computer Science (SOCS) UNIVERSITY OF PETROLEUM & ENERGY STUDIES, DEHRADUN- 248007. Uttarakhand**

Table of Contents

|  |  |  |
| --- | --- | --- |
| **Topic** | | **Page No** |
| Table of Content | | 2 |
| 1 | Introduction | 3 |
|  | 1.1 Purpose of the Project | 3 |
|  | 1.2 Target Beneficiary | 3 |
|  | 1.3 Project Scope | 3 |
|  | 1.4 References | 4 |
| 2 | Project Description | 4 |
|  | 2.1 Reference Algorithm | 4 |
|  | 2.2 SWOT Analysis | 7 |
|  | 2.3 Project Features | 8 |
|  | 2.4 Design diagrams | 9 |
| 3 | System Requirements |  |
|  | 3.1 User Interface | 10 |
|  | 3.2 Software Interface | 10 |
|  | 3.4 Protocols | 10 |
| 4 | Non-functional Requirements |  |
|  | 4.1 Performance requirements | 11 |
|  | 4.2 Security requirements | 11 |
|  | 4.3 Software Quality Attributes | 11 |
|  | 4.4 Conclusion | 12 |
| 5 | Output | 13-14 |

**INTRODUCTION**

**Purpose of the Project**

Weather forecast and disaster alarm is to display the weather and warn about disasters. This project is the study of data of disaster-prone areas and then alarm people according to the weather. There are many levels of disasters like earthquake, flood, tsunami and other storms, landslides, cloudburst, forest fire. In this project, we focus on reviewing the live weather data and history of data on disasters collected so far and analyze and predict disasters.

**For Ex: Weather: Heavy rain., Warning: It is not safe, Danger of floods.**

**This tells people that it is not safe to travel and to stay as there is heavy rain and danger of floods**.

Like this example our project first will forecast the weather and analyses disaster readings of the place and then according to the data It will analyze if it’s safe or not or there can be floods.

**Target Beneficiary**

* People living in hilly areas.
* Travelers and explorers.

**Project Scope**

We aim to perform weather forecast and disaster prediction in hills. This project will help peoples to avoid potential disasters and travel safely in hilly areas. It will help residential people to be safe and secure avoiding potential disasters. We will analyze the weather and disaster data generated by reputed database and classify the weather as being rainy, heavy rain, sunny and clear as for disaster as danger of floods.

**References**

* Flood Disaster in Mountain Environment: A Study of Himachal Pradesh, India Vishwa B. S. Chandel, Simrit Kahlon, Karanjot Kaur Brar: Managing Our Resources Perspectives and Planning

(accessed on 2 Dec 2021)

:<https://www.researchgate.net/publication/303436834_Flood_Disaster_in_Mountain_Environment_A_Study_of_Himachal_Pradesh_India>

* Managing the Risks of Extreme Events and Disasters to Advance Climate Change Adaptation Special Report of the Intergovernmental Panel on Climate Change:

(accessed on 15 Nov 2021)

<https://www.ipcc.ch/site/assets/uploads/2018/03/SREX_Full_Report-1.pdf>.

* Geological investigations in Rudraprayag district with special reference to mass instability

(accessed on 15 Nov 2021)

:<https://dmmc.uk.gov.in/files/pdf/Rudraprayag_final.pdf>.

* Exploring the nature and various forms of the impact of climate change on flora, fauna and local communalities in Uttarkashi, Pithoragarh and Bageswhar districts of Uttarakhand

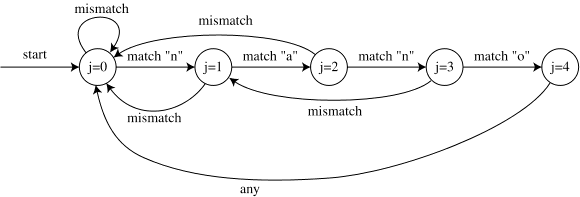
(accessed on 20 Nov 2021)

:<https://dmmc.uk.gov.in/files/Impacts_of_Climate_Change.pdf>.

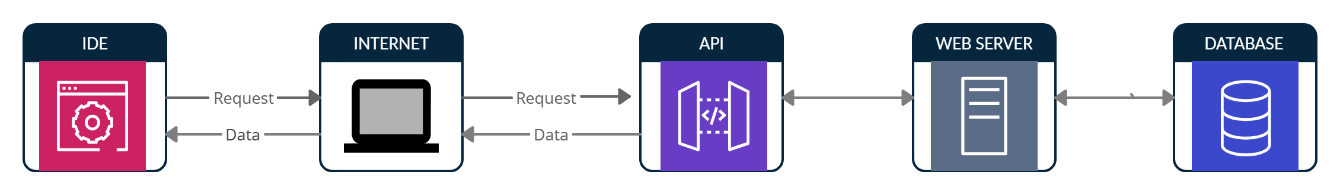
**PROJECT DESCRIPTION**

**Reference Algorithm**

KMP Algorithm is one of the most popular patterns matching algorithms. KMP stands for Knuth Morris Pratt. KMP algorithm. KMP algorithm is used to find a "Pattern" in a "Text". This algorithm campares character by character from left to right. But whenever a mismatch occurs, it uses a preprocessed table called "Prefix Table" to skip characters comparison while matching. Some times prefix table is also known as LPS Table. Here LPS stands for "Longest proper Prefix which is also Suffix".



**Figure 1: Finite State Automata Diagram of KMP Algorithm.**



**Figure 2: Establishing connection through API call**

**Stages for API call**

1. Request to Api gateway.

2. Fetch information from data server.

**Stages for weather forecast and disaster alarm:**

1. Display the weather data.

2. Sort the data in the table.

3. Analyze the data with the researched disaster data.

4. Alarm about the disaster.

****

**Figure 3: Basic Workflow of the program**

**Characteristic of data**

We can send data in text format to the user with the help of our application and inform the people through the alarm, since it is a real time data and could even be modified at any moment.

**SWOT Analysis**

* **Strengths**

1. Accessibility: It is easily accessible from anywhere.
2. Make use of existing facilities, easy to manage.
3. Early warning to those at risk.
4. Cost effectiveness.

* **Weakness**

1. Since it is real time interaction the significant weakness would be the requirement of high-speed internet and efficient data.
2. Congestion in Network: Too many requests from the user may lead to congestion. Overload can lead to breaking-down of servers. Also, if the number of users increases more than the limit their requests would get queued which may take extra time to get response from the server.

* **Opportunities**

Provides opportunity for all section of people to connect in a more interactive environment where everyone can share their ideas etc. and can give their thoughts different shapes and even make designs to increase the understanding.

Collaboration in developing the more effective warning system.

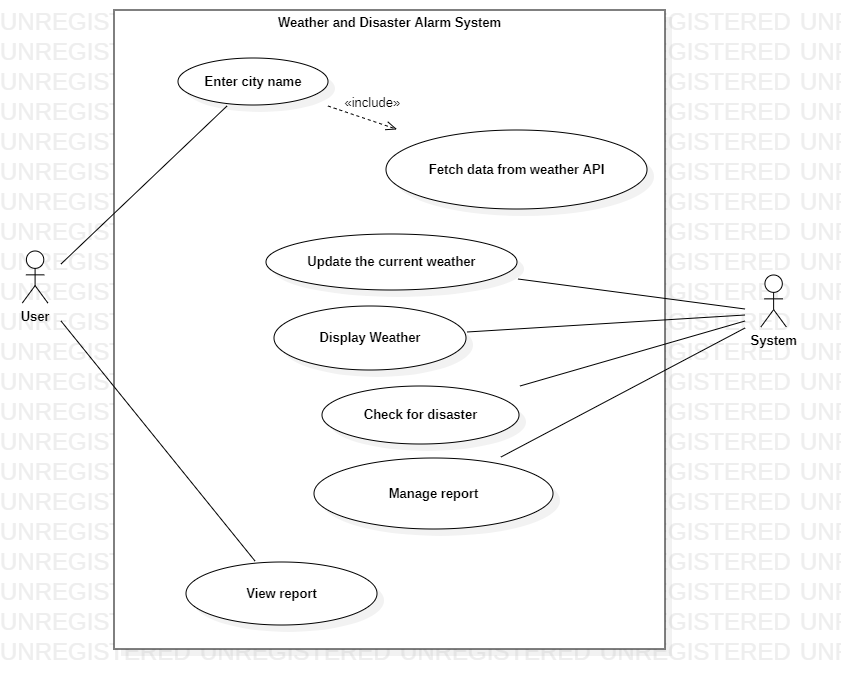
* **Threats**

For this system to reach its maximum capabilities it needs a good computing server that can respond to the requests in no time. Ineffective weather Forecasting and Warning system and delayed evacuation response.

**Project Features**

* + - It also helps in understanding the nature of floods, by this we can take proper measures as to be safe from the floods.
    - It can lead to more accurate tools for extracting disaster and weather behaviors and provides means for rescue operations before disasters hits hence assuring no human lives are lost.
    - To analyze the weather readings.
    - To determine the weather conditions by using the analyzed readings.
    - To display and read the weather conditions and confirm it with the existing data on disasters.
    - To classify the data into danger of floods and safe

**Design Diagram**

****

**Figure 4 : Use-Case Diagram of Weather and Disaster Alarm System.**

* Request to Api gateway.
* Fetch information from data server.
* Display the weather data.
* Analyze the data with the researched disaster data.
* Alarm about the disaster.

**System Requirements**

**Hardware Interface**

* 64 bits processor architecture supported by any Operating System.
* Minimum RAM requirement for proper functioning is 512 MB.

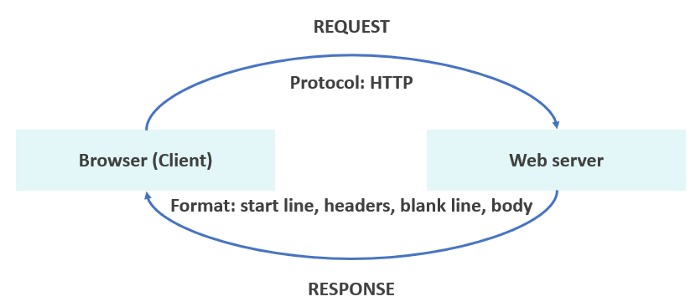
**Software Interface**

* Operating System: Windows, Linux
* (Integrated development Environment)
* Openweather API.

**Protocols**

When we are creating the connection between the client and the server using HTTP (Hypertext transfer Protocol) which is the protocol used by web browsers refer to as client for communicating with web servers. The data between the two parties is not in plain files like in FTP (File transfer Protocol), but hypertext written in HTML format which are transferred over the internet here ‘s how this process looks:

1. The request in this case is sent by the browser and includes the request method, the URL and the protocol version followed by other information.
2. Server receives the request, and runs a program to process it.
3. The server returns the HTTP response to the browser.

 **Figure 5: Client Server Diagram of Api.**

**Non-Functional Requirements**

**Performance requirements**

* In this project, user initiates communications by entering data.
* User issues request to a server.
* Server receives the request and runs the program to process it.
* If number of users are more, increase the capacity of the server to respond to the requests.

**Security requirements**

Broken, exposed or hacked APIs are behind major data breaches. They exposed sensitive medical, financial, and personal data for public consumption. In our case a hacker can hack and misuse our API key to access the data and mislead the user about disaster which can further leads to loss of human lives which could have been saved.

**Software Quality Attributes**

1. **Availability**

The project can consistently be accessible 24/7. If number of users are more, the capacity of the server is increased to respond to the requests.

1. **Maintainability**

Maintainability is very easy as we are not using any physical location and database.

1. **Flexibility**

This project is highly flexible depending on the user requests. If the user requests may increase, increase the capacity of the server to respond to the requests.

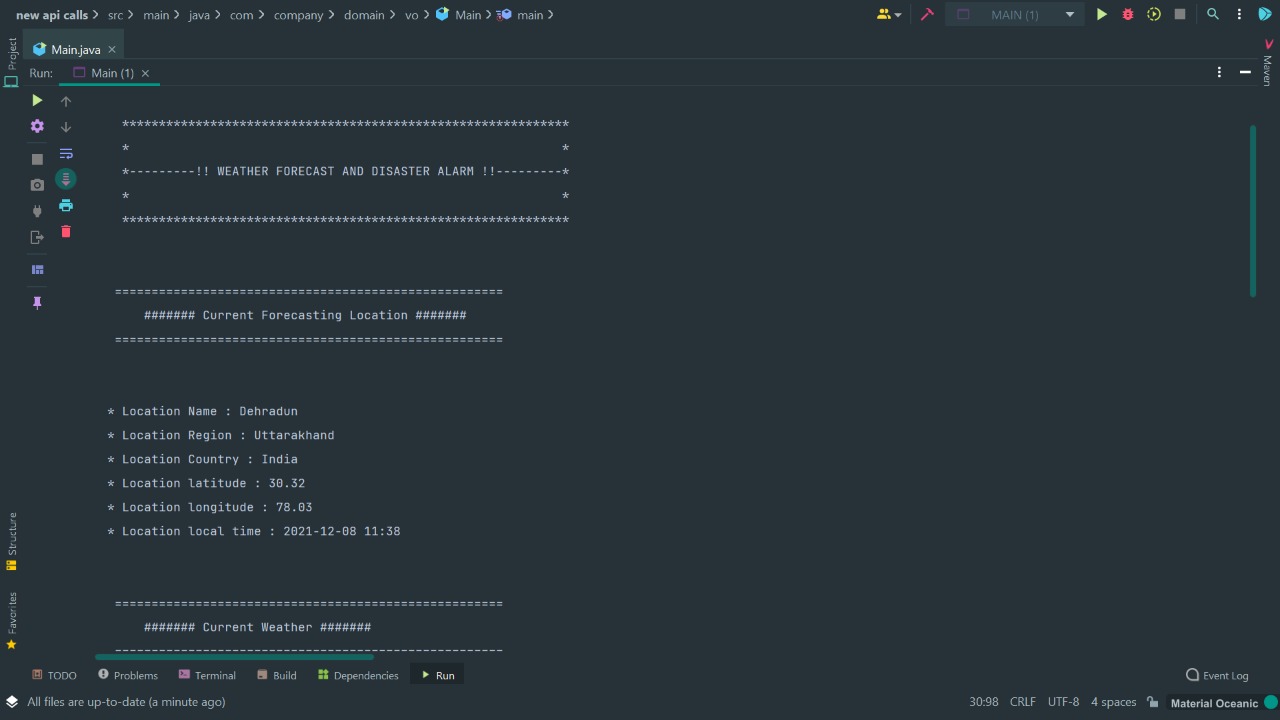
1. **Usability**

This specific project is ease to use.

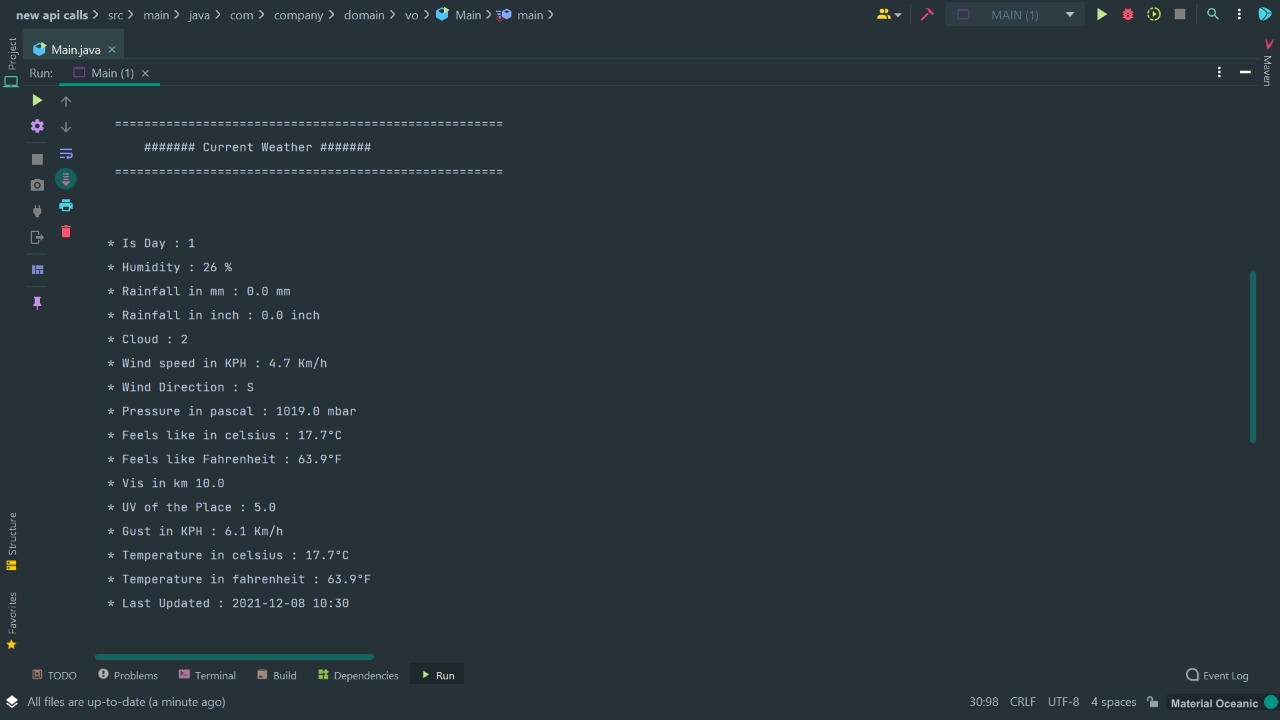
**Conclusion**

* Weather forecasting is a complex and challenging science that depends on the efficient interplay of weather observation, data analysis by meteorologists and computers, and rapid communication systems.
* Meteorologists have achieved a very respectable level of skill for shortrange weather forecasting.
* Further improvement is expected with denser surface and upper air observational networks, more precise numerical models of the atmosphere, larger and faster computers and more are to be realized.
* However, continued international cooperation is essential, for the atmosphere is a continuous fluid that knows no political boundaries.

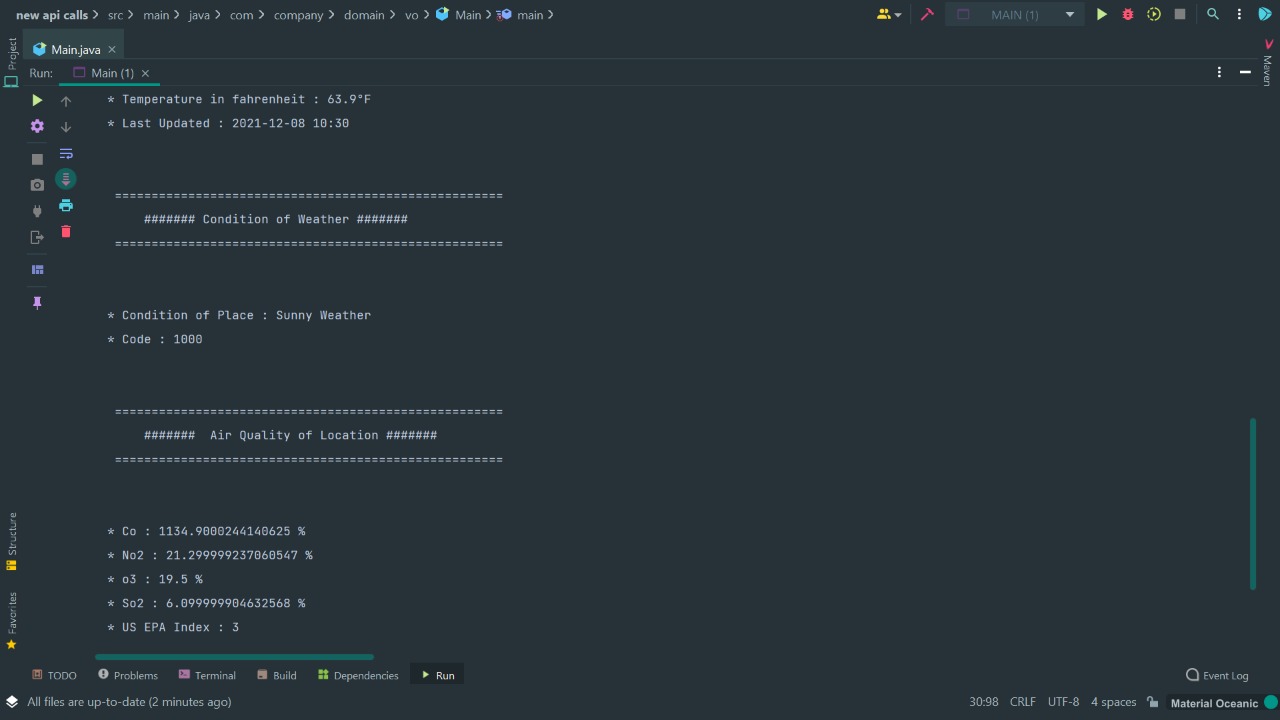
**Output**

****

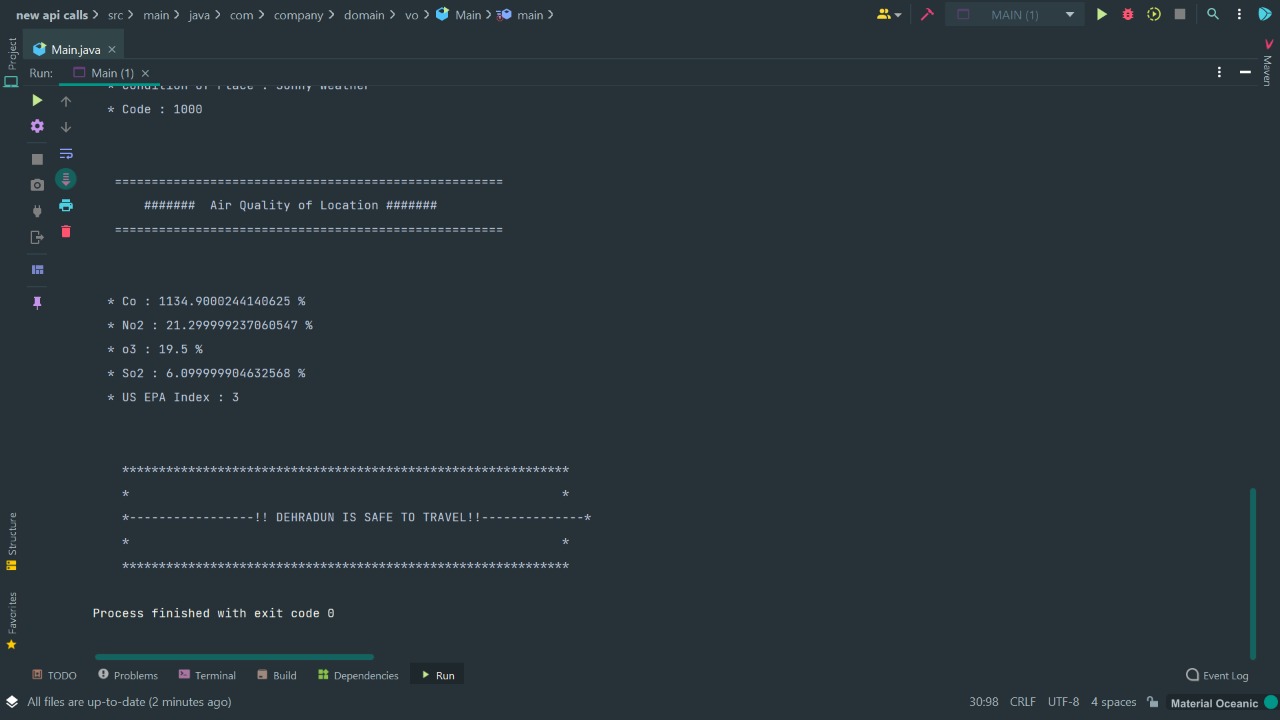
**Screen:1**

****

**Screen:2**

****

**Screen:3**

****

**Screen:4**