WEATHER APP

Document Version Control

Date Issued	Version	Description	Author
24/01/23	1	Initial commit with temperature and weather condition name data	Ngamlenmang Touthang
25/01/23	2	React toastify added and readme update	Ngamlenmang Touthang

Contents

Document Version Control.....2

Abstract 4

1. Introduction.....5
   1.1. Why this High-Level Design Document?.....5
   1.2. Scope.....5
   1.3. Definitions.....5
2. General Description.....6
   2.1.Product Perspective.....6
   2.2.Problem statement.....6
   2.3.Proposed solution.....6
   2.4.Further improvements.....6
   2.5.Technical Requirement.....6
   2.6.Data Requirements.....7
   2.7.Tools used.....7
3. Design Details.....8
   3.1.Process Flow.....8
   3.2. Error handling.....8
   3.3. Performance.....9
   3.4. Reusability.....9
   3.5. Application Compatibility.....9
   3.6. Resource Utilization.....9
   3.7. Deployment.....9
4. Conclusion.....10

Abstract

The weather web application is designed to provide users with up-to-date and accurate weather information for a given city around the world. The application utilizes OpenWeathermap API's to collect data which is then processed and formatted for easy viewing on the user interface. The user interface is designed to be simply and user-friendly, displaying current weather condition with temperature and date. The application is designed to be responsive, adapting to different screen sizes and devices to provide optimal user experience. The application is deployed on a web server, making it easily accessible to users through a web browser.

1. Introduction

1.1. Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model of coding. The document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project.
- List and describe the non-functional attributes like
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2. Scope

The HLD documentation presents the structure of the system, such as data will be collected form the API, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system

1.3. Definitions

Term	Description
API	Application Programming Interface
APP	Application
IDE	Integrated development Environment

2. General Description

2.1. Product Perspective

The Weather application is a web application system which is based on OpenweatherMap API application that gives the user the ability to check the current weather of a particular city.

2.2. Problem Statement

Create a simple weather App that allows users to get current weather information based on the city user enter. Any external weather API is allow to be used.

2.3. Proposed Solution

The solution proposed here is a Weather app web-application can be built in order to meet the above the problem statement using an external API that is Openweathermap api along with the reactjs javascript library to create the user interface where the weather data can be displayed.

2.4. Further Improvements

The Weather app can be improve with more features like maintaining user account provided with authentication, storing history of city name enter by a particular user and improving the user interface with more data and animation.

2.5. Technical Requirements

This document addresses some of the requirements for proper functioning of the weather App

- The app must be able to handle a certain number of request per second or must be able to process data within a certain time frame.
- The app must be able to protect sensitive data or must comply with certain security standards.
- The app must be accessible to any number or users with a browser without a significant decrease in performance
- The app must be able to integrate with other systems or platform, or must support certain operating systems or hardware.

2.6. Data Requirements

Data requirement completely depend on Weather API that is OpenweatherMaps API of weather base on a particular city. This API return the require data for the Weather App

2.7. Tools used

JavaScript programming language and Reactjs which javascript library and some package like axios, react-toastify and tailwindcss which is CSS library along with OpenweatherMap API are used to build the whole system.

- Vscode is used as IDE
- For getting the Weather data OpenweatherMap API is used.
- Front end development is done using Reactjs, TailwindCss , React toastify
- Axios is used to make http/https request
- GitHub is used as version control system
- No database is used.

3. Design Details

3.1. Process Flow

For getting weather for the city name enter, we used an external API i.e Openweathermap API.

Below is the process flow diagram is as shown below.

Proposed methodology

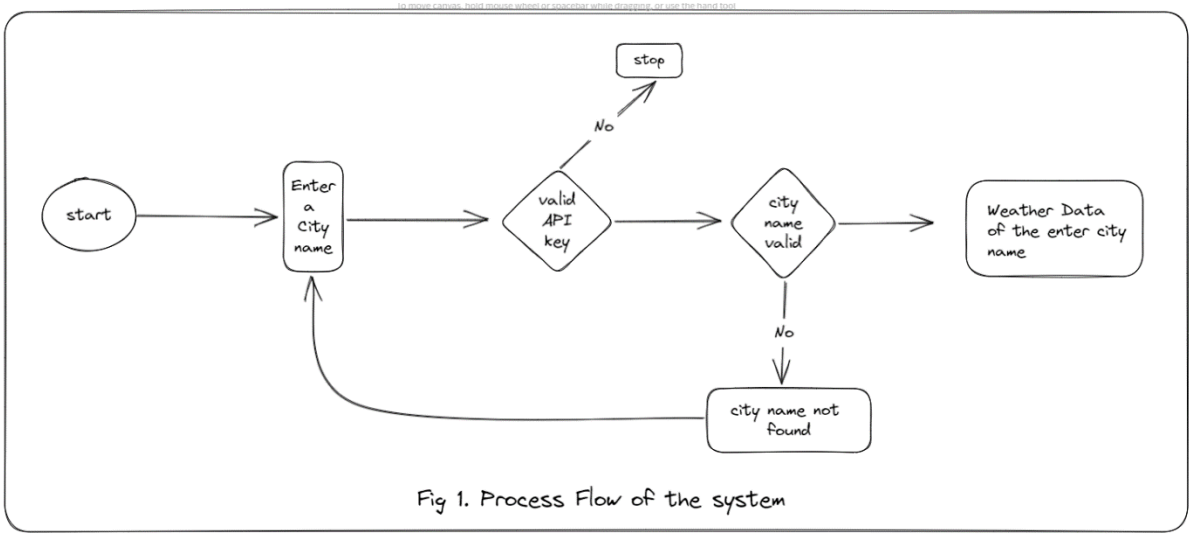

Fig 1. Process Flow of the system

1.1. Event log/Notification

The system should log/notify every event so that the user will what the app is running into.

- The system app will notify if the user enter something that is not a name of city.
- The system app will notify the user if the API key is invalid or has issue from the external API provider.

1.2. Error handling

Should errors be encounterd, an explaination will displayed as to what went wrong? An error will be defined as accordingly. React toastify is used to display all the error in the user interface.

1.3. Performance

The Weather app web application should be able to handle a number of request and performed within specific time frame and any users with Internet connection and browser should be able to use it anywhere they are.

1.4. Reusability

The code written and the components used should have the ability to be reused with no problems.

1.5. Application Compatibility

The different components for this project will be using a JavaSCript library that is Reactjs as an interface between them. Each component will have its own task to perform, and it is the job of the JavaScript to ensure proper transfer of information.

1.6. Resource Utilization

When any request is performed. It will likely use all the processing resources available until that function is finished.

1.7. Deployment

The Weather app is deploy via Netlify via github.

4. Conclusion

The developed Weather web application will give the weather details base the different city name, so that a user can identify the current weather of the city and can take necessary action according to the weather condition and the user can be well aware about the weather condition of the city where the user like to travel.