Low Level Design (LLD)

WEATHER APP

Written BY	Ngamlenmang Touthang
Document Version	1.0.3
Last Revised Date	01/02/23

Document Version Control

Change Record:

Version	Date	Author	Description
1.0.1	24/01/23	Ngamlenmang touthang	Temperature and main weather added
1.0.2	25/01/23	Ngamlenmang touthang	City name, favicon and react-toastify added
1.0.3	26/01/23	Ngamlenmang touthang	Docs update

Reviews:

Version	date	Reviewer	comments

Approval Status:

version	Review Date	Reviewed By	Approved By	Comments

Contents

1. Introduction

1.1. Why this Low-Level Design Document?

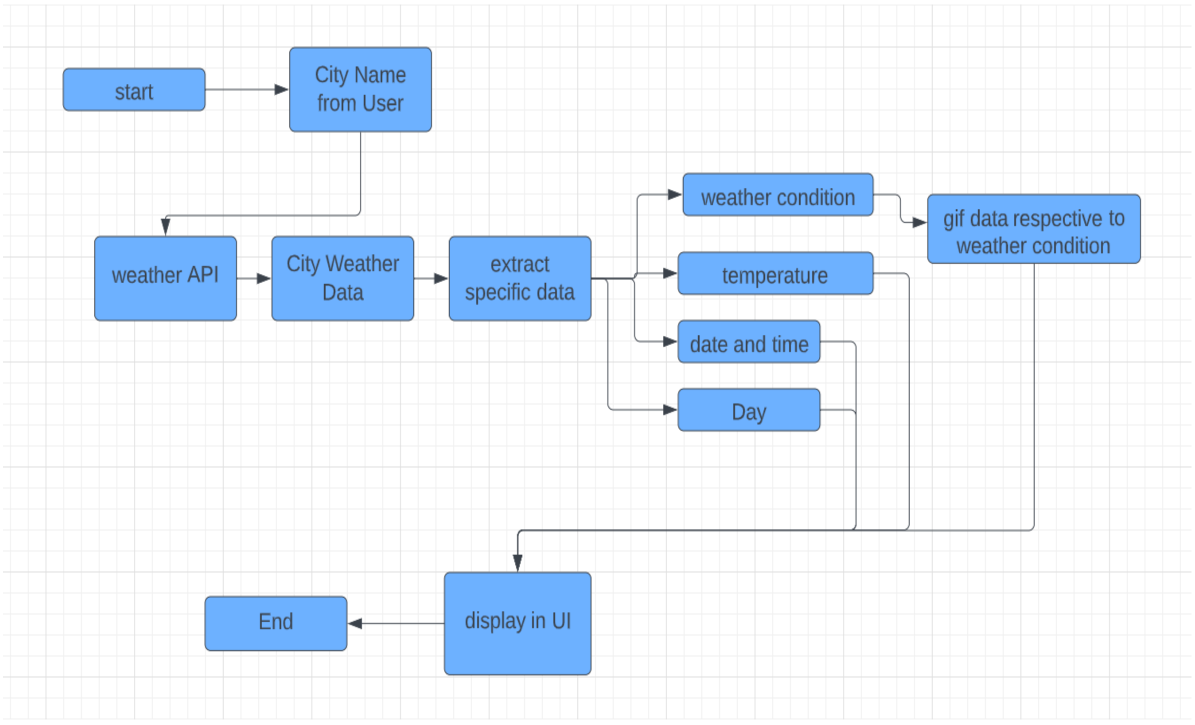
The purpose of this Low-Level Design (HLD) Document is to present a detailed description of the Weather app. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraint under which it must operate and how the system will work. This document is intended for both the stakeholders and the developers of the system and will be proposed to the higher management for its approval.

The main objective of the project is to get current weather data of a particular city. Based on the weather data provided by OpenWeatherMap api

1.2. Scope

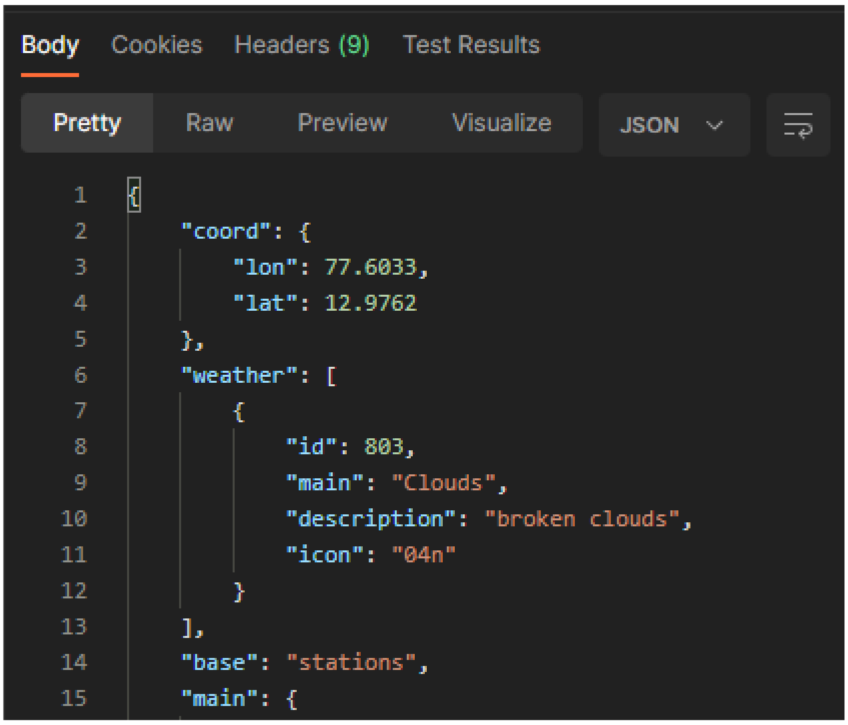
The Low level design documentation presents the structure of the system, such as data will be collected form the API, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the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

2. Architecture

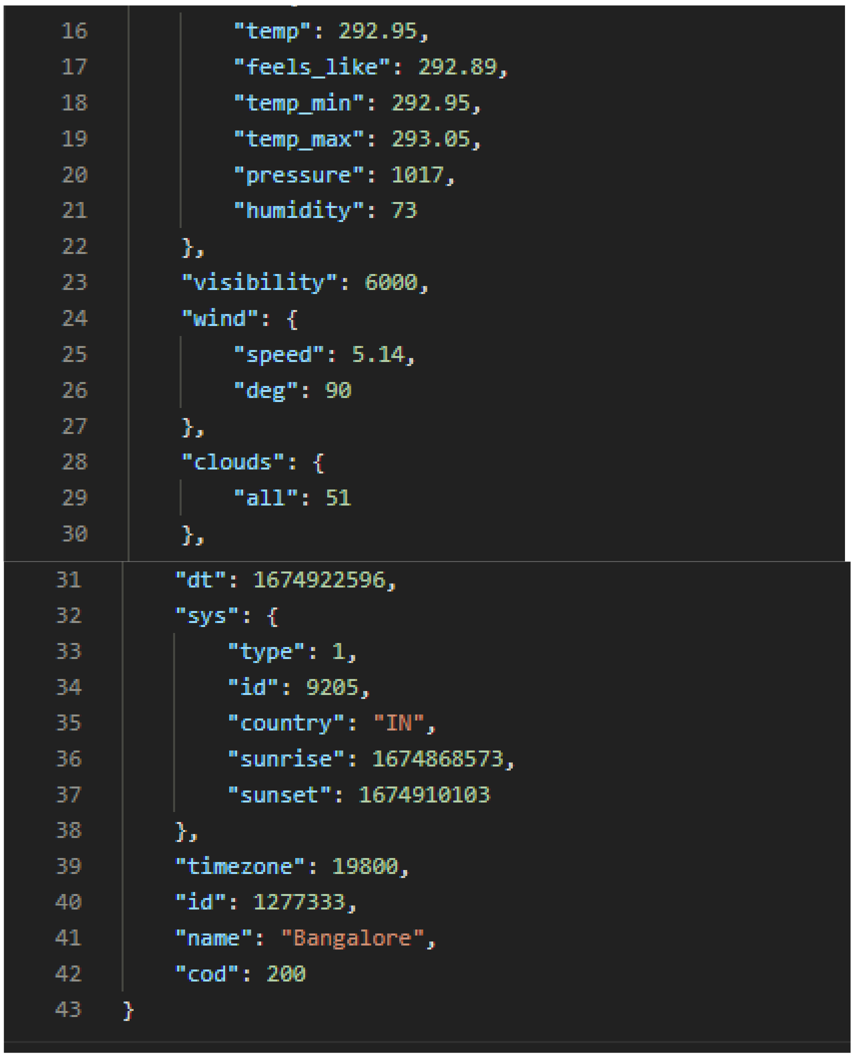


3. Architecture Description

3.1. Dataset



```
{
    "coord": {
        "lon": 77.6033,
        "lat": 12.9762
    },
    "weather": [
        {
            "id": 803,
            "main": "Clouds",
            "description": "broken clouds",
            "icon": "04n"
        }
    ],
    "base": "stations",
    "main": {
```



```
            "temp": 292.95,
            "feels_like": 292.89,
            "temp_min": 292.95,
            "temp_max": 293.05,
            "pressure": 1017,
            "humidity": 73
        },
        "visibility": 6000,
        "wind": {
            "speed": 5.14,
            "deg": 90
        },
        "clouds": {
            "all": 51
        },
        "dt": 1674922596,
        "sys": {
            "type": 1,
            "id": 9205,
            "country": "IN",
            "sunrise": 1674868573,
            "sunset": 1674910103
        },
        "timezone": 19800,
        "id": 1277333,
        "name": "Bangalore",
        "cod": 200
    }
```

3.2. Weather dataset overview

The weather data set from openweathemap API consist of objects of data namely, object of coordi, array of weathers, key value of base, object of main, key value of visibility, object of wind, object of winds, key value of dt, object of sys, key of timezone, id, name and code.

3.3. Input schema

In the input schema we will be putting any desired city name

3.4. Getting city data

Among the different data object received for a particular city the following data are taken for this system.

- Weather mains which consist of – Clouds, Thunderstorm, Drizzle, Rain, Snow, Clear, Clouds, Mist, Smoke, Haze, Dust, Fog, Sand, Ash, Squall, Tornado
- Weather main temperature which is received as kelvin is converted into celsius

3.5. Logging

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.

3.6. Database

No database is used, data is provided by an external API (OpenweatherMap API)

3.7. Deployment

The Weather app is deploy via Netlify via github.

4. Unit Test Cases

Test Case Description	Pre-Requisite	Expected Result
Verify whether the application URL is accessible to the user	1.Application URL should be defined	Application URL should be Accessible to the User
Verify whether the application loads completely for the user when the URL is accessed.	1.Applicaiton URL is accessible 2.Application is accessible	The application should load completely for the user when the URL is accessed.
Verify whether user is able to enter city name in the input field	1.application is accessible 2.User able to enter city name	User should be able to enter city name in the input field
Verify whether user gets Enter button to submit inputs	1.Application is accessible 2. Application should have submit button	User should get submit button to submit the inputs
Verify whether user is presented with weather data from openweathermap API	1. Application is accessible 2. Application should able to fetch data.	User should be presented with weather API of the particular city result on clicking submit
Verify whether user is able to put weather data in the UI	1. application is accessible	User should be see the weather details of the entered city in the UI