# Table of contents

Contents

[Table of contents 1](#_Toc13345472)

[Business Requirements 2](#_Toc13345473)

[Analysis 2](#_Toc13345474)

[Design 7](#_Toc13345475)

[Unit Test Cases (High Level) 8](#_Toc13345476)

# Business Requirements



# Analysis

***Overview*:**

Latitude & Longitude

Weather Simulator

Output

Point Name

Following are the input and output from the system,

Inputs to the System:

1. Latitude and Longitude pair or
2. Point Name. Point name could be city, any place etc..

Following are the Input Format(s) Supported:

**FORMAT 1**

Latitude: 9° 55' 30.7236'' N (DDDMMSS FORMAT)  
Longitude: 78° 7' 11.19'' E(DDDMMSS FORMAT)

DDD(3 digits) – Degree

MM (2 Digit) - Minutes

SS (2 Digit) – Seconds

**Note**:

1 Degree = 60 NM, 1 Minute = 1NM, 1 Seconds = 1/60=0.0167NM

1 Degree is computed based on Earth Circumference.

**CALCULATION FOR ONE DEGREE**

I.e. Earth Circumference / 360 Degree (Circle).

40,075 KM/360 = 111.319 KM

1 Nautical Mile = 1.852 KM

So 111.319/1.852 = 60 NM, Hence each degree in Lat or Long is actually 60 NM (111.31 KM) of distance

**FORMAT 2**

Latitude: +/- 9.925201 (Degrees)  
Longitude: +/- 78.119775 (Degrees)

Output from the System:

1. Output format is “**Location | Latitude, Long, Altitude | Local Time | Conditions | Temperature | Pressure | Humidity**”

Example - Sydney | -33.86, 151.21, 39 | 2015-12-23T05:02:12Z | Rain | +12.5 | 1004.3 | 97

***Temperature:***

|  |  |  |
| --- | --- | --- |
| **Factors influenced on Temperature** | **Description** | **Scope** |
| Sea Surface Temperature (SST) | Sea Surface Temperature is the reference for calculating the temperature. SST will vary based on the latitude and longitude. For this project, Reference table will be maintained for calculating the temperature. | In Scope |
| Elevation | Altitude will be calculated using third party library and the same can be used for temperature calculation along with SST. | In Scope |
| Atmospheric zones | Different atmospheric zones will contribute in the temperature. For example, There will be an increase pattern in Troposphere and decrease pattern in Stratosphere. | Image result for red cross  Not Considered as a scope |
| Time in a day (Angle of solar radiation) | Self-rotation of the earth every 24 hours. | Image result for red cross  Not Considered as a scope |
| Time considering Climate(Angle of solar radiation) | Rotation of earth around sun in an elliptical orbit. | Image result for red cross  Not Considered as a scope |
| Humidity | One of the contribution factor in temperature calculation but not critical. Water drop in the air (deviation) will be added to the existing calculation. | Image result for red cross  Not Considered as a scope |
| Pressure | One of the contribution factor in temperature calculation but not critical. | Image result for red cross  Not Considered as a scope |

*Pseudocode:*

1. Given the lat-long, select the appropriate SST which matches the input lat-long in the reference data.
2. Calculate the altitude using the **geopy** python module.
3. Calculate the actual temperature using the below formula,

“**Reference Temperature - (altitude \* temp\_lapse\_constant)**”

*Structure of Reference Data:*

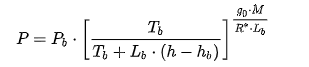
|  |  |  |
| --- | --- | --- |
| Latitude | Longitude | SST (°C) |
| -30 S | 0 | 21 |
| 75 N | 0 | 4 |
| 15 N | -120 W | 25 |

***Pressure****:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Factors influenced on Temperature** | **Description** | **Scope** | **Assumption** |
| Atmospheric zones | Different atmospheric zones will contribute in the pressure calculation.  Standard temperature, Standard pressure, gravity, lapse rate are depending on atmospheric zones (Troposphere, Stratosphere etc...) |  | This project will consider the first zone which is troposphere |
| Elevation | Altitude will be calculated using third party library and the same can be used for pressure calculation along with SST. |  |  |

*Pseudocode:*

1. Given the lat-log, select the appropriate SST which matches the input lat-long in the reference data.
2. Calculate the altitude using the **geopy** python module.
3. Calculate pressure using the below formula,



Where:

* {\displaystyle P\_{b}}Pb is static pressure (hPa)
* Tb is standard temperature (K)
* Lb is standard temperature lapse rate (K/m) in ISA
* h is height above sea level (m)
* hb is height at bottom of layer b (meters; e.g., h1 = 11 000 m)
* R\* is universal gas constant: 8.3144598 J/(mol·K)
* go is gravitational acceleration: 9.80665 m/s2
* M is molar mass of Earth's air: 0.0289644 kg/mol

***Reference:*** <https://en.wikipedia.org/wiki/Barometric_formula>

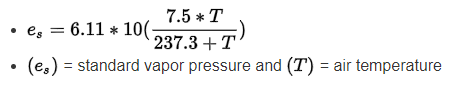
***Humidity:***

Humidity is “the amount of moisture in the air”. This can be measured using dry-bulb and wet-bulb temperature experiments. But for this project, dew point temperature will be used to calculate the relative humidity.

|  |  |  |  |
| --- | --- | --- | --- |
| **Factors influenced on Temperature** | **Description** | **Scope** | **Assumption** |
| Dew point temperature | Temperature below which water particles in the air will be condensed into water |  | For this project,  Dew point temperature = air temperature – 3 deg cel |
| Temperature | Air temperature. Please refer Temperature section for the details. |  |  |

*Pseudocode:*

1. Use the air temperature which is already calculated in the previous section.
2. Calculate the saturated vapor pressure and actual vapor pressure using the below formula



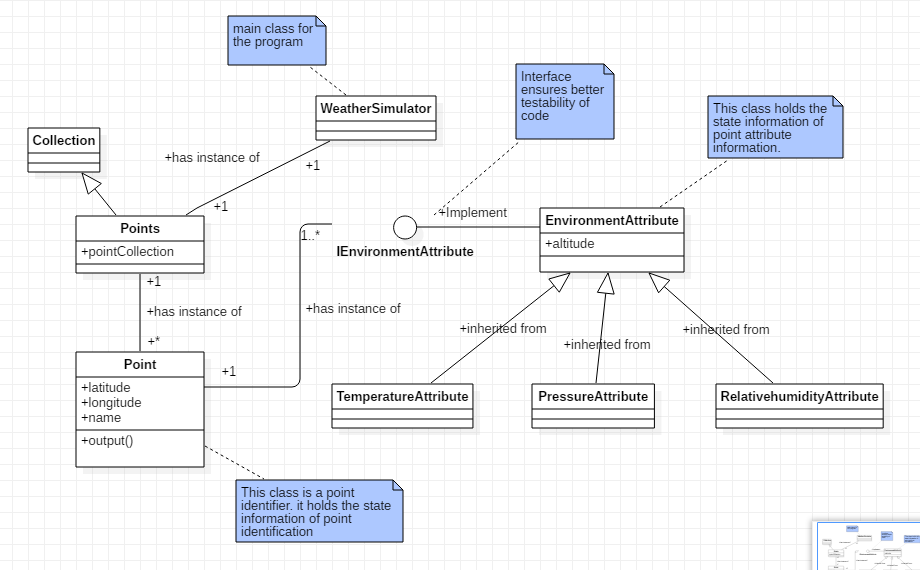
1. Relative humidity can be calculated by dividing the vapor pressure by the saturated vapor pressure.

***Conditions (Rain, snow, sunny)***

|  |  |  |  |
| --- | --- | --- | --- |
| **Factors influenced on Temperature** | **Description** | **Scope** | **Assumption** |
| Temperature | Based on these parameters range, Conditions will be derived. |  |  |
| Humidity |  |  |

# Design

***Class Level Design***



***Design Thoughts:***

1. All classes are **cohesive**.
2. Implemented Interface pattern to ensure **decoupling/loose coupling and testability**.
3. All classes are polymorphic and bound by the structural pattern
4. Ensures flexibility in creating mocks for unit testing.

# Unit Test Cases (High Level)

Intention is to cover UT for all the methods which has logic and the following verticals will be considered,

1. Boundry conditions
2. Outside permissible values
3. Not supported values
4. Invisible values – space etc
5. No values
6. Disturbing external environment during execution – Http calls, DB calls if any.

Following are the Unit test cases derived from requirements. **Complete set will be available during development.**

|  |  |
| --- | --- |
| **Test case** | **Test Target** |
| Input:  Lat: &\*^%  Long: ^&\* | Point Class.get/set method of properties |
| Lat: +12345  Long: +3456 | Point Class.get/set method of properties |
| Lat: 12345  Long: 3456 | Point Class.get/set method of properties |
| Point Name: @#$$ | Point Class.get/set method of properties |
| Point Name: Sydney | Point Class.get/set method of properties |
| Lat: +25.992345  Long: +25.992345 | Point Class.get/set method of properties |
| Lat: 0.000000  Long: 0.000000 | Point Class.get/set method of properties |
| Lat: +90.000000  Long: +90.000000 | Point Class.get/set method of properties |
| Lat: -90.00000  Long: -90.0000 | Point Class.get/set method of properties |
| Lat: +90.88  Long: +180.99 | Point Class.get/set method of properties |
| Lat: -90  Long: -180 | Point Class.get/set method of properties |
| Lat: +90  Long: -180 | Point Class.get/set method of properties |
| Lat: -90  Long: +180 | Point Class.get/set method of properties |
| Lat: +90.8?\*\*\*\*\*  Long: +180.8\*\*\*\* | Point Class.get/set method of properties |
| Lat: +90.@@  Long: +180.@@ | Point Class.get/set method of properties |
| Lat: 9° 55' 30.7236'' N  Long: 78° 7' 11.19'' E | Point Class.get/set method of properties |
| Lat: 180° 55' 30.7236'' N  Long: 78° 7' 11.19'' E | Point Class.get/set method of properties |
| Lat: 90° 55' 30.7236'' N  Long: 200° 7' 11.19'' E | Point Class.get/set method of properties |
| Lat: 90° 100' 30.7236'' N  Long: 180° 7' 11.19'' S | Point Class.get/set method of properties |
| Lat: 90° 100' 30.7236'' N  Long: 180° 7' 11.19'' S | Point Class.get/set method of properties |
| To be filled in | TemperatureAttribute.Calculate() |
| To be filled in | PressureAttribute.Calculate() |
| To be filled in | RelativeHumidityAttribute.Calculate() |
| To be filled in -- Disconnecting network calls, etc | Over All Environment |