# Table of contents

Contents

[Table of contents 1](#_Toc13937890)

[Business Requirements 2](#_Toc13937891)

[Analysis 2](#_Toc13937892)

[Design 7](#_Toc13937893)

[Development 8](#_Toc13937894)

[Self-Review Checklist 8](#_Toc13937895)

[Unit Test Cases (High Level) 8](#_Toc13937896)

# 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**

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 Microsoft virtual earth API.
3. Calculate the actual temperature using the below formula,

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

*Structure of Reference Data:*

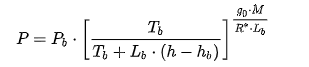
|  |  |  |
| --- | --- | --- |
| Latitude | Longitude | SST (°C) |
| -30 S | 0 | 21 |
| 75 N | 0 | 4 |
| 15 N | -120 W | 25 |

***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 Microsoft virtual earth API.
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>

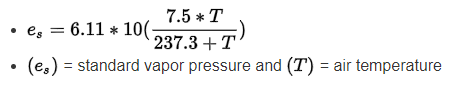
***Humidity:***

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, We use random number generator to calculate the relative humidity |
| 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



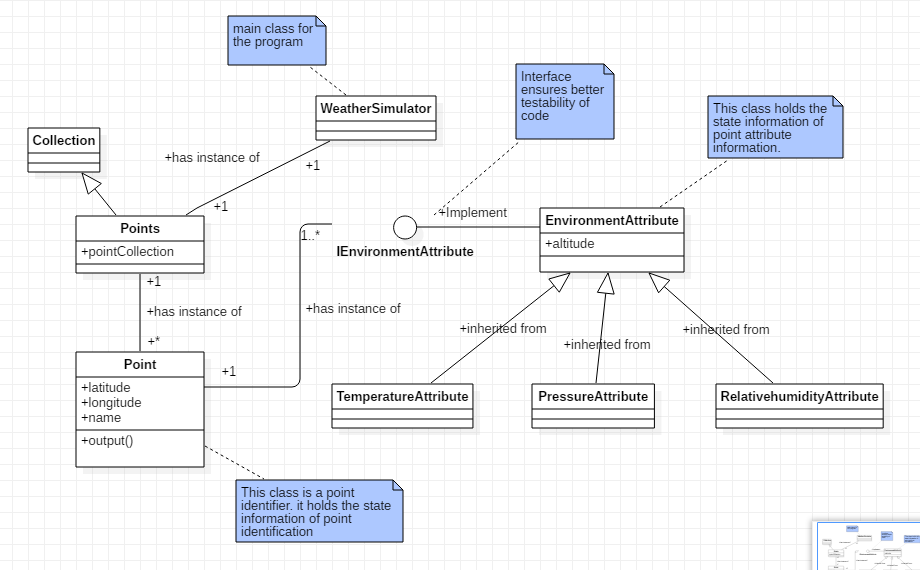
1. Relative humidity can be calculated by dividing the vapor pressure by the saturated vapor pressure.

***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.

# Development

Please refer the source code in **src** folder.

***Design pattern considered:***

1. Adaptor pattern for external communication
2. CRC methodology to derive the objects

# Self-Review Checklist

1. Constructor Initialization
2. Property get/set for a private variable
3. Python baseclass invocation using mro
4. Logical group of domain responsibilities
5. Exception scenarios
6. Input validation scenarios
7. Logic validation

# 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: ^&\* | Points.read |
| Lat: +12345  Long: +3456 | Points.read |
| Lat: 12345  Long: 3456 | Points.read |
| Point Name: @#$$ | Points.read |
| Point Name: Sydney | Points.read |
| Lat: +25.992345  Long: +25.992345 | Points.read |
| Lat: 0.000000  Long: 0.000000 | Points.read |
| Lat: +90.000000  Long: +90.000000 | Points.read |
| Lat: -90.00000  Long: -90.0000 | Points.read |
| Lat: +90.88  Long: +180.99 | Points.read |
| Lat: -90  Long: -180 | Points.read |
| Lat: +90  Long: -180 | Points.read |
| Lat: -90  Long: +180 | Points.read |
| Lat: +90.8?\*\*\*\*\*  Long: +180.8\*\*\*\* | Points.read |
| Lat: +90.@@  Long: +180.@@ | Points.read |
| Lat: 9° 55' 30.7236'' N  Long: 78° 7' 11.19'' E | Points.read |
| Lat: 180° 55' 30.7236'' N  Long: 78° 7' 11.19'' E | Points.read |
| Lat: 90° 55' 30.7236'' N  Long: 200° 7' 11.19'' E | Points.read |
| Lat: 90° 100' 30.7236'' N  Long: 180° 7' 11.19'' S | Points.read |
| Lat: 90° 100' 30.7236'' N  Long: 180° 7' 11.19'' S | Points.read |
| Different values of lat and alt | TemperatureAttribute.Calculate() |
| Different values of lat and alt | PressureAttribute.Calculate() |
| Different values of lat and alt | RelativeHumidityAttribute.Calculate() |