Weather Data Simulation Design Document

## Purpose

The purpose of this document is to provide Weather Data Simulation project design guidelines.

We provide a prototype of a program which artificially simulates the weather and outputs weather data in a standard format for our game to read.

The objective of this project is as follows:

1. Toy simulation of the environment emits measurements location, position, localtime, conditions, temperature, position, humidity
2. Toy simulation of the environment takes measurements temperature, sea level pressure, conditions and humidity.
3. We need to capture city coordinators by lookup world cities coordinator list.

## Project Summary

### Project Overview

Weather Data Simulation project provide a prototype of a program which artificially simulates the weather and outputs weather data in a standard format for our game to read.

### Scope

The scope of this project is as follows:

1. Toy simulation of the environment emits measurements location, position, localtime, conditions, temperature, position, humidity
2. Toy simulation of the environment takes measurements temperature, sea level pressure, conditions and humidity.
3. We need to capture city coordinators by lookup world cities coordinator list.
4. It must be simple to use and understand the project

### Acceptance Criteria

The Acceptance Criteria is as follows:

1. Toy simulation of the environment takes measurements temperature, sea level pressure, conditions and humidity.
2. We need to capture city coordinators by lookup world cities coordinator list.
3. Weather Data Predictor predicts weather data by using two techniques

\* Longitude and Latitude Adjustments

\* Year wise Predictor here slightly using sliding window algorithm

1. It generates output as location, position, localtime, conditions, temperature, position, humidity
2. It accepts parameters as city, country, datewithtime, worldcitiescoordinatorslistpath, weatherhistoricalyearwisedataPath

### Assumptions and Constraints

|  |  |
| --- | --- |
| **Assumptions and Constraints** | **Impact to Plan if not True** |
| Worldcitiescoordinatorslist file contains id, country, city, latitude, longitude, elevation separated with ‘|’ Delimiter. | There will be re-work required. |
| We should have 365 days weather historical information with respect to latitude longitude and elevation. Same thing we are taking as weatherhistoricalyearwisedataPath | There will be re-work required. |
| We should get input as city, country, datewithtime, worldcitiescoordinatorslistpath, weatherhistoricalyearwisedataPath  e.g1., Sydney Australia 2015-12-23T16:02:13Z < worldcitiescoordinatorslistpath > < weatherhistoricalyearwisedataPath >  e.g2. Sydney Australia 2015-12-23 < worldcitiescoordinatorslistpath > < weatherhistoricalyearwisedataPath > | There will be re-work required. |
| This Project can generate weather report from 2000 to 2025 by based on year for the given input. Here we are using 365 cycle data to generate the report. We are checking below conditions to predict data   1. Checking for previous day for specified input year 2. Checking for previous year till 2000 for specified input year 3. Checking for next year till 2025 for specified input year | There will be re-work required. |
| We are adjusting latitude and longitude values to get the historical year data for any position. | There will be re-work required. |
| We are loading weather historical data for 3 places for 365 days   1. Sydney 2. Hyderabad 3. Los Angeles | We need to load more position 365 days information to get accurate weather data |

### Prerequisite Environment

1. Java
2. Maven
3. GITHub

### Reference

**Project GitHub Location:**

[**https://github.com/srawanthi/weather-simulator-jar/**](https://github.com/srawanthi/weather-simulator-jar/)

[**https://github.com/srawanthi/weather-simulator**](https://github.com/srawanthi/weather-simulator)

## Design Architecture

Weather Simulator Program Weather data in two files:

1. World\_Cities\_Coordinators file

World\_Cities\_Coordinators.txtfile contains

location|position|localtimeutc|conditions|temparature|position|humidity fields as below

1|Afghanistan|Kabul|34.5166667|69.1833344|1808

2|Afghanistan|Kandahar|31.61|65.6999969|1015

3|Afghanistan|Mazar-e Sharif|36.7069444|67.1122208|369

4|Afghanistan|Herat|34.34|62.1899986|927

Constructs HashMap<String, CityPositionDto> object by parsing World\_Cities\_Coordinators

1. Historical Weather data file

Weather-1year-historicaldata.txt file contains

position|city|timezone fields as below

-33.86|151.21|39|2016|7|1|15|10|6|76|59|38|1031|1029|1028|Sunny

-33.86|151.21|39|2016|7|2|17|11|4|87|65|35|1029|1026|1023|Sunny

-33.86|151.21|39|2016|7|3|17|11|5|93|70|46|1023|1018|1013|Rain

-33.86|151.21|39|2016|7|4|23|16|9|94|54|16|1014|1012|1009|Sunny

-33.86|151.21|39|2016|7|5|21|17|12|58|43|21|1015|1013|1009|Sunny

-33.86|151.21|39|2016|7|6|19|13|8|71|53|26|1017|1015|1014|Sunny

-33.86|151.21|39|2016|7|7|20|13|6|81|52|17|1015|1012|1009|Sunny

Constructs HashMap<String, HashMap<String, HashMap<String, HashMap<String, WeatherHistoryDataDto>>>> as HashMap<longlat,HashMap<year,HashMap<month,HashMap<day, WeatherHistoryDataDto>>>>

1. Input as city, country as city, country, datewithtime, worldcitiescoordinatorslistpath, weatherhistoricalyearwisedataPath

Ex1: Sydney Australia 2015-12-23T16:02:13Z < worldcitiescoordinatorslistpath > < weatherhistoricalyearwisedataPath >

Ex2: Sydney Australia 2015-12-23 < worldcitiescoordinatorslistpath > < weatherhistoricalyearwisedataPath >

Sydney 2015-12-23 16:02:13

1. It can generate weather report from 2000 to 2025 by based on year for the given input. Here we are using 365 cycle data to generate the report. We are checking below conditions to predict data
   * Checking for previous day for specified input year
   * Checking for previous year till 2000 for specified input year
   * Checking for next year till 2025 for specified input year
2. We are adjusting latitude and longitude values to get the historical year data for any position.
3. Create WeatherDataDto as output

temp 19.5 humidity 19.5 pressure 19.5 conditions Rain city Melbourne Locationtimeutc 2015-09-14T10:51:10 position -37.8139966;#:144.9633179;#:58

### Class Diagram

|  |
| --- |
| WeatherGameSimulator |
| city |
| country |
| datewithtime |
| worldcitiescoordinatorslistpath |
| watherhistoricalyearwisedataPath |
| main() |
| setupProcess() |
| validateArgs() |
| simulateWeatherData () |

|  |
| --- |
| WeatherDataUtils |
| getFileContentinBuffer() |
| parseLineToWeatherHistoryObject() |
| classifyHistoryYearWiseData() |

|  |
| --- |
| WeatherDataPredictor |
| predictYearWiseWeatherData () |
| mapWeatherData() |
| predictlonglat() |
| lookupinHistoryYearData() |
| lookupinFutureYearData() |

|  |
| --- |
| WeatherDataDto |
| -location |
| -position |
| -localtimeutc |
| -conditions |
| -temparature |
| -pressure |
| -humidity |
| all setter and getter methods |

|  |
| --- |
| WeatherHistoryDataClassifier |
| classifyWorldCitiesInfoCoordinators() |
| classifyWeatherHistoryInfo() |

|  |
| --- |
| WeatherHistoryDataDto |
| Latitude  longitude  elevation  year  month  day  tempMax  tempMin  tempAvg  humMax  humAvg  humMin  seaPressureMax  seaPressureAvg  seaPressureMin  conditions |
| all setter and getter methods |

|  |
| --- |
| WeatherConstants |
| -ROW\_SEPERATOR ="|";  -ROW\_SPLIT\_SEPERATOR ="\\|";  -TIME\_yyyy\_MM\_dd\_T\_HH\_mm\_ss\_Z= "yyyy-MM-dd'T'HH:mm:ss'Z'";  -TIME\_UTC\_TIMEZONE = "UTC";  -DATE\_FIELD\_SEPERATOR = "-";  -TIME\_FIELD\_SEPERATOR = ":"; |
|  |
|  |

|  |
| --- |
| CityPositionDto |
| -city |
| -country |
| -latlongelevation |
| -latlong |
| all setter and getter methods |

### Output Data

**Weather Data Output Data:**

• Location is an optional label describing one or more positions,

• Position is a comma-separated triple containing latitude, longitude, and elevation in metrics above sea

Level.

• Local time is given input time,

• Conditions is either Snow, Rain, Sunny,Cloudy

• Temperature is in °C,

• Pressure is in hPa, and

• Relative humidity is a %.

## Enhancement Plans

1. Here we can pass all input parameters in file get the weather report in another file.
2. We can implement with Spark and scala application
3. We can create partition folders while loading weather input data into HDFS