Software Requirements Specification

for

iWeather

Version 1.0

Prepared by

Shubham Sorte
130911228
luk4shubhs@gmail.com

Instructor:	**Rajesh K**
Course:	**Software Engineering Lab**
Lab Section:	**IT B1**
Date:	**16th March , 2015**

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Shubham Sorte	Complete Requirements for the iWeather Software Project	16/03/15

1. Introduction

This document provides a Software Requirements Specification (SRS) document for a desktop based Weather app (namely iWeather) that is meant to show the weather information of any place on earth. In the following sections, we specify the purpose of this document, its intended audience, the scope of the end-product, and all sources used in the production of this document.

1.1 Document Purpose

The purpose of this document is to specify and establish the functional and non-functional requirements associated with the iWeather App version 1.0.

iWeather version 1.0 SRS will serve as the backbone for any other documents to be developed for this project in the future .Moreover, it will provide the basis for the future software verification and testing by specifying the behavior requirements of the system in the form of a Use Case Diagram and other modeling diagrams.

1.2 Product Scope

iWeather App shall be a desktop based application which will allow the user to view the present weather conditions in any city of the world. iWeather will receive the data from an online service which updates its databases in real time.

iWeather would also receive other information such as wind speed, humidity, pressure, visibility and other such conditions in that particular city. It would also tell us the sunrise and sunset time in that location. Along with this the iWeather App would also give the temperature forecast for the present day as well as the next four days.

1.3 Intended Audience and Document Overview

This SRS documentation is intended for Developers who would want to work on improving the iWeather App. The document specifies all the modules that were used to build the iWeather App. The document also specifies the installation steps for these modules on any system.

The project iWeather 1.0 source code is currently hosted on Github.com. Its under a private repository that means that the developer working on it only can view the code. Soon after the alpha testing of the software is completed, the project would be open sourced and would be available to download and used by everyone. Later the developers can contribute to the project code on Github itself.

1.4 Definitions, Acronyms and Abbreviations

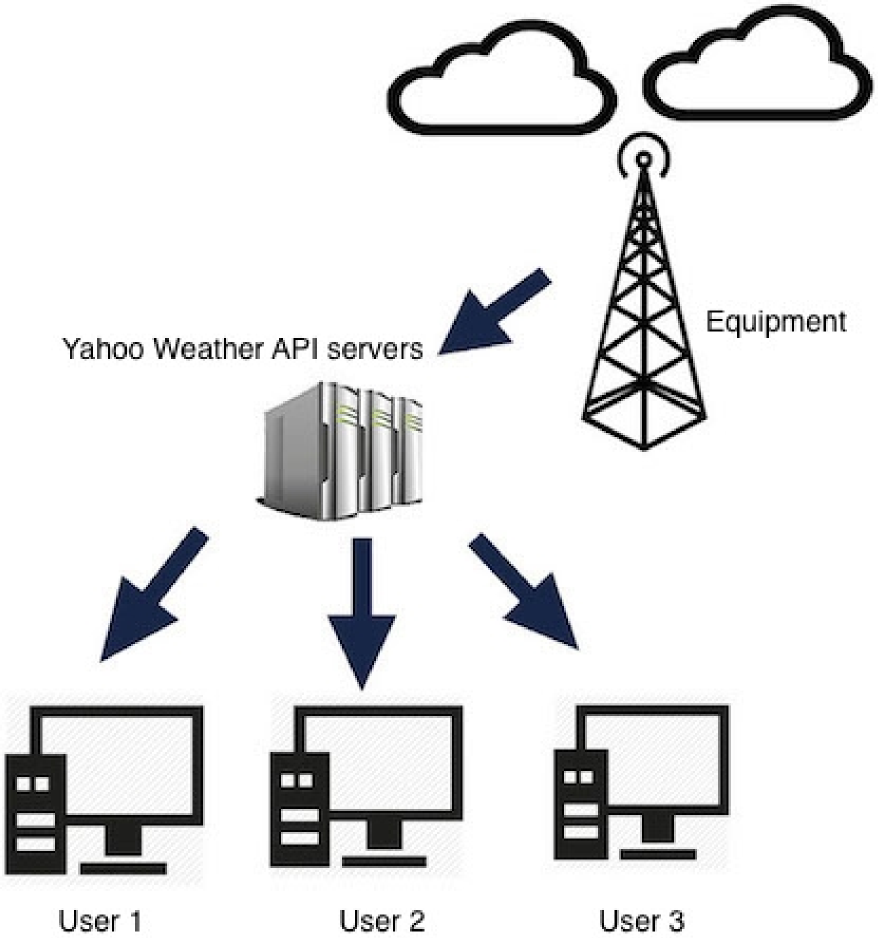
UI – User Interface
JSON – Java Script Object Notation
API – Application Programming Interface
VCS – Version Control System

2. Overall Description

The main aim of this project was to introduce a new application, which would significantly differ from the existing desktop weather applications. This application would be having a lot more features than the applications currently available.

2.1 Product Perspective

The product perspective is best illustrated in Figure given below, which describes the overall environment and the ways components of this environment interact with each other.



The weather equipment like thermometers, satellites, etc. will capture the information and the information would be sent to the Yahoo Weather API servers. This information can be accessed by any user all around the world. Since the weather measurement equipment are always active this information changes in real time and there is no lag.

~~2.2~~ Product Functionality

The product functionality are as follows:

- Works on the desktop platform like any other desktop software
- An easy to use Interface which any user can easily understand
- Weather information at your fingertips
- Weather forecast lets users be ready for the weather conditions
  (E.g. – If people see that it would be getting cold, they can be ready with more warm clothes. If people see that rains are coming, then they can be ready with their umbrellas, raincoats, etc.)
- Images used for depicting different weather conditions gives a nice and aesthetic feel to the application
- Also available other information such as
  - Wind speed
  - Wind Direction
  - Humidity
  - Atmospheric pressure
  - Visibility
  - Sunrise and sunset time
  - High and low temperature for the day
  - Real feel conditions like partly cloudy, mostly sunny, etc.

2.3 Users and Characteristics

The different types of users who would be using this application would be

1. Casual User – Users who own and use computers for daily purposes. These are people who just want to see the weather conditions without any research or study purposes. These people are the end users for this application.
   E.g. Students, teachers, employees of any company, etc.

2. Developer – Developers would be using this software to find bugs and fix them. The code being open sourced on Github , the developers would be able to contribute to the software therefore making it perfect.

3. Small Research Groups – People who do a little bit research on weather conditions in a particular region or locality. These people would have access to only the amount of information they need. These people may include local TV reporters, Newspaper Firms,etc.

2.4 Operating Environment

The software shall operate on an Intel based architecture personal computer (Pentium III or above). The computer station shall be connected to the Internet via LAN (Local Area Network) or via Wifi (if supported) and the operating system shall support this network interface.

Some operating systems that would support the application are –

- Windows XP
- Windows 7
- Windows 8
- Windows 10
- Mac OS X 10.6 and above
- Linux (Ubuntu 12.04 and above)

2.5 Design and Implementation Constraints

Upon completion of the requirements analysis process the following constraints were identified:

- For fetching the weather information the system (computer) must be connected to the internet via Wi-Fi or LAN
- The developers who want to contribute to the project must be well versed with JAVA
- The System on which we want the application to run must have Java Developer Kit (JDK) installed.
- The developer need to have an account on Github.com in order to contribute.
- The developer should have knowledge related to GIT VCS(Version Control System)

2.6 Assumptions and Dependencies

The software assumptions and dependencies are as follows –

- The Windows, Linux or Mac OS and the minimum hardware must be in place.
- The software requires a good internet connection. In absence of an internet connection the software may not work as the data fetching is done in real time from the Yahoo servers.
- Yahoo Weather is not responsible for any incorrect data and therefore 100% reliability should not be assumed on the software.
- The user must be well versed with using a Personal Computer in order to interact with the software.

3. Specific Requirements

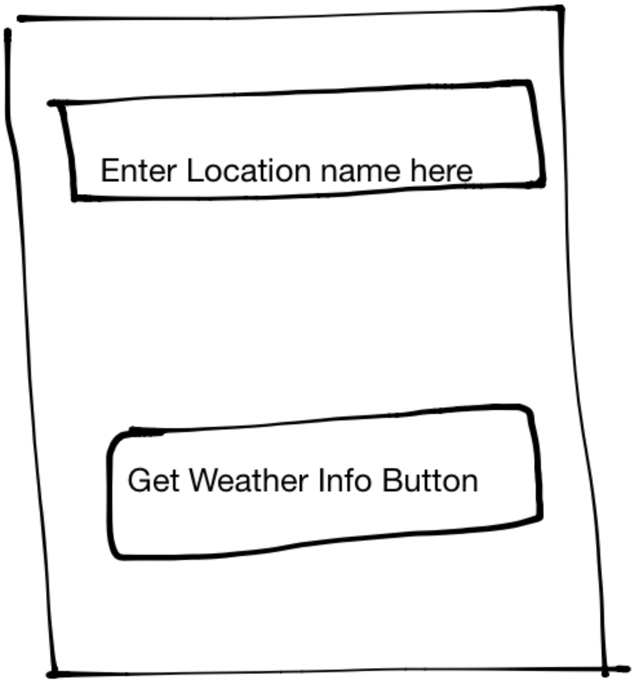
3.1 External Interface Requirements

3.1.1 User Interfaces

As we know that in today's contemporary world simplicity is not just the mere absence of clutter and ornamentation, it's about bringing order to complexity.
Such care is taken while designing the interface for this application. The software meant to give detailed info but in a simple and elegant manner.

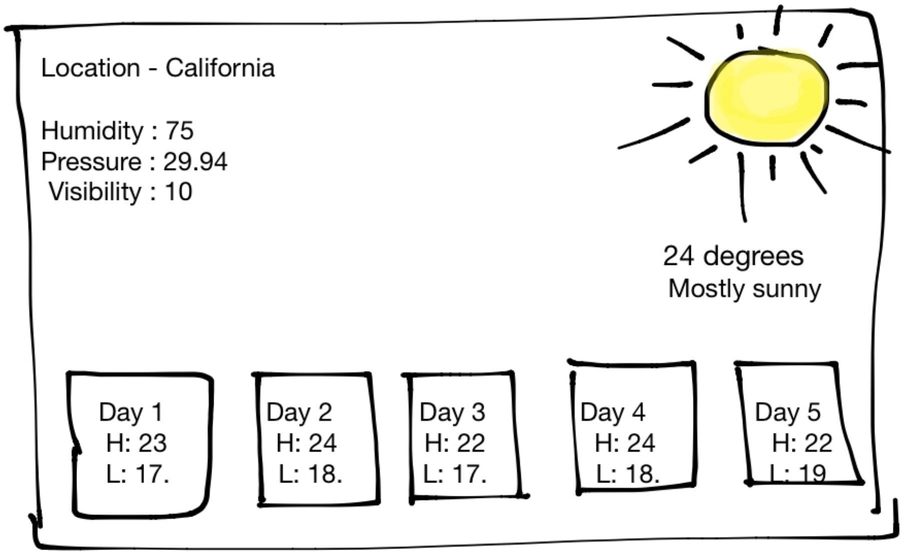
The application consists of two main screens -

1. Choose the location – Here the user can type the name of the city or location whose weather information he wants. A rough sketch of the same is given below.



2. Main Weather info screen – This screen has the weather info for the location the user has typed in the last screen. Its divided into certain sections.
   Sections are -
   - Forecast section – info of current day as well as next 4 days.
   - Atmospheric conditions section – Info about humidity,atmospheric pressure,etc.
   - Real Feel Info – This would give a brief of the weather condition.E.g. – Sunny, cloudy, etc.

   A rough sketch of the same is given below –



3.1.2 Hardware Interfaces

The user need not worry about the hardware required to measure weather conditions. Yahoo Weather Service takes care of it. The Yahoo people have placed their equipment in different places in the world which give almost accurate information about weather. It also have people who enter the gathered info into their databases which are hosted on the Yahoo Weather API servers.

3.1.3 Software Interfaces

The iWeather App works on all major Desktop OS like Windows, Linux and Mac OS X.

The app uses a 3rd party open source library (package) called **org.json**.
This library includes two files –

- json-simple-1.1.1.jar
- json-20141113.jar

The user who needs to use the app must set the classpath for these files on the Command prompt or the Terminal depending on the OS he/she is using.

Instruction related to setting the classpath are included in the README.md file of the project.

3.1.4 Communications Interfaces

The system shall communicate with the internet via HTTP.
A link would be generated according to the information requirements.
This link also called the API endpoint would then be sent via a synchronous request to the server via the internet which in turn will return the data.
Security issues would be taken care of by the Yahoo Weather Service.

3.2 Functional Requirements

3.2.1 User Interface Operations

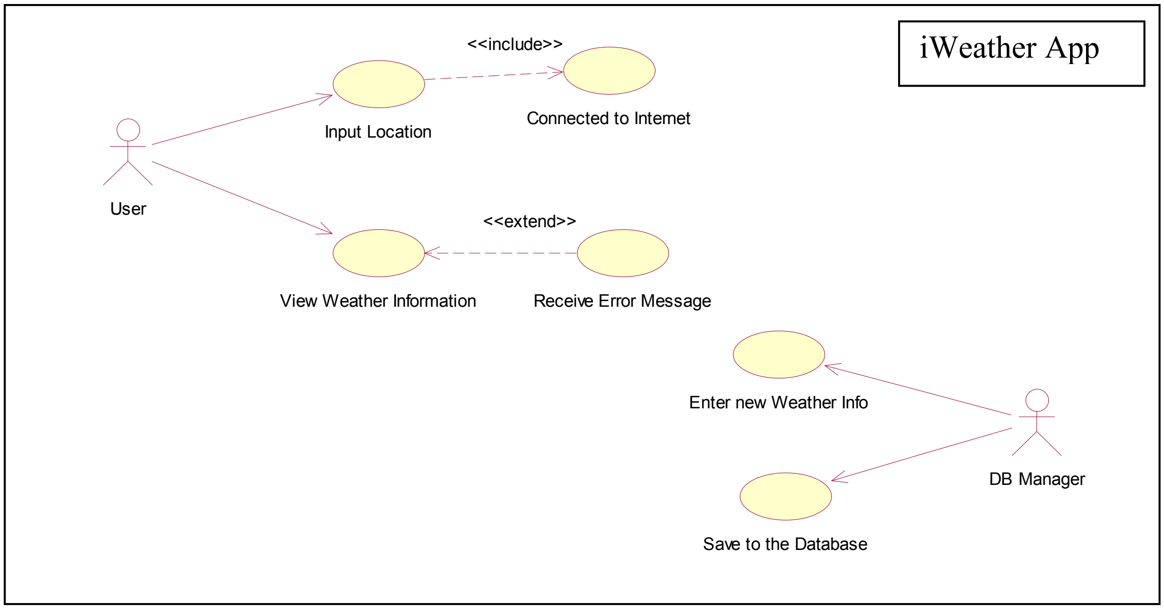
1. The user will type in the name of the location in the text field present in the first view of the application.
2. He will then press the view weather info on the same screen which will take him to another screen.
3. On the next screen he would be given weather info and all the other info that the application is intended to give.
4. If the weather info for the location that the user has typed in is not available then the next screen would display an ‘Weather info not available for this location’ message.

3.2.2 Background Activity

- As soon as the user types in the location name and presses view info button, a URL is generated with this info. The URL contains info such as API Keys, name of the location,etc.
- The generated URL is then sent using a synchronous HTTP request the Yahoo Weather Service.
- The Yahoo Weather Service then returns the data in JSON format.
- The library json.org is then used to parse this JSON data into meaningful data.
- The data (parsed) is used to get individual elements like weather condition on a particular day, real feel conditions of the location etc.

3.3 Behaviour Requirements

3.3.1 Use Case View



4. Other Non-functional Requirements

4.1 Performance Requirements

Some performance requirements for iWeather are –

1. If the user is connected to the internet, then the weather information should be display to him under 10 seconds(depending on the speed of the internet connection)
2. The JAVA Swings used in the project are completely interactive and dynamic.
3. If the information related to weather in a particular area is not available , then the error message would be displayed.
4. The software won't respond to the user if there is no internet connection.

4.2 Safety and Security Requirements

The Yahoo weather service would monitor the weather conditions and update them in real time. The client side would require fetching the data from the internet. This would lead to bandwidth consumption every time we require the weather information.

4.3 Software Quality Attributes

Listed below are some of the more important Software Quality Attributes, which have been identified as essential to this project.

4.3.1 Reliability

The system shall not crash ,or hang, other than as a result of an operating system error.
Also the absence of proper internet connection may result in crashes.

4.3.2 Availability

The software would be available to every individual. The person can use, edit or distribute the software without any license. The software in no means should be misused.

4.3.3 Correctness

Yahoo Weather service has a reputation of being 100% accurate with the data. But they don't guarantee you this fact. Therefore we shouldn't rely on the software data completely.

4.3.4 Portability

The software is available to use for desktop OS such as Windows,Linux and Mac OS X. The same files being used over different OS makes it portable enough.

4.3.5 Usability

No specific training should be required on the part of the user.
The user must be well versed with normal PC functioning and usage.

4.3.6 Maintainability

The code shall be fully documented. The documentation is available in the README.md file present with the other project files. Comments are included in the code wherever needed. Proper code factoring including indentation of code is done.
The code is available on Github and people who want to improve or work on bugs can do so by contributing to the repository.

5. Other Requirements

Some more requirement are –

1. Knowledge of swings - The interface of the application is written in Swings in Java. The developer must have knowledge related to the same.

2. GIT and terminal – The project GIT as the VCS(Version Control System). The developer need to have knowledge about the different commands used in GIT on terminal. For e.g. – git pull, git push –u origin master, etc.

3. HTTP Requests – The application sends a HTTP POST request to the server. The developer must have the knowledge about how POST requests work in the JAVA environment.

Appendix A – Data Dictionary

Here is an example data which the application receives –

```
{
 "query": {
  "count": 1,
  "created": "2015-03-15T06:55:53Z",
  "lang": "en-us",
  "results": {
   "channel": {
    "title": "Yahoo! Weather - Nome, AK",
    "link":
"http://us.rd.yahoo.com/dailynews/rss/weather/Nome__AK/*http://weather.yahoo.com/forecast/US
AK0170_f.html",
    "description": "Yahoo! Weather for Nome, AK",
    "language": "en-us",
    "lastBuildDate": "Sat, 14 Mar 2015 6:53 pm AKDT",
    "ttl": "60",
    "location": {
     "city": "Nome",
     "country": "United States",
     "region": "AK"
    },
    "units": {
     "distance": "mi",
     "pressure": "in",
     "speed": "mph",
     "temperature": "F"
    },
    "wind": {
     "chill": "-5",
     "direction": "0",
     "speed": "0"
    },
    "atmosphere": {
     "humidity": "75",
     "pressure": "29.94",
     "rising": "0",
     "visibility": "10"
    },
    "astronomy": {
     "sunrise": "9:22 am",
     "sunset": "8:59 pm"
    },
    "image": {
        "title": "Yahoo! Weather",
        "width": "142",
        "height": "18",
        "link": "http://weather.yahoo.com",
        "url": "http://l.yimg.com/a/i/brand/purplelogo//uh/us/news-wea.gif"
      },
      "item": {
        "title": "Conditions for Nome, AK at 6:53 pm AKDT",
        "lat": "64.5",
        "long": "-165.41",
        "link":
"http://us.rd.yahoo.com/dailynews/rss/weather/Nome__AK/*http://weather.yahoo.com/forecast/US
AK0170_f.html",
        "pubDate": "Sat, 14 Mar 2015 6:53 pm AKDT",
        "condition": {
          "code": "34",
          "date": "Sat, 14 Mar 2015 6:53 pm AKDT",
          "temp": "-5",
          "text": "Fair"
        },
        "description": "\n<img src=\"http://l.yimg.com/a/i/us/we/52/34.gif\"/><br />\n<b>Current
Conditions:</b><br />\nFair, -5 F<BR />\n<BR /><b>Forecast:</b><BR />\nSat - Partly Cloudy.
High: -2 Low: -18<br />\nSun - Mostly Sunny. High: 5 Low: -5<br />\nMon - AM Snow Showers.
High: 17 Low: 10<br />\nTue - PM Snow Showers. High: 26 Low: 23<br />\nWed - Cloudy. High:
33 Low: 29<br />\n<br />\n<a
href=\"http://us.rd.yahoo.com/dailynews/rss/weather/Nome__AK/*http://weather.yahoo.com/foreca
st/USAK0170_f.html\">Full Forecast at Yahoo! Weather</a><BR/><BR/>\n(provided by <a
href=\"http://www.weather.com\" >The Weather Channel</a>)<br/>\n",
        "forecast": [
          {
            "code": "29",
            "date": "14 Mar 2015",
            "day": "Sat",
            "high": "-2",
            "low": "-18",
            "text": "Partly Cloudy"
          },
          {
            "code": "34",
            "date": "15 Mar 2015",
            "day": "Sun",
            "high": "5",
            "low": "-5",
            "text": "Mostly Sunny"
          },
          {
            "code": "14",
            "date": "16 Mar 2015",
            "day": "Mon",
            "high": "17",
            "low": "10",
            "text": "AM Snow Showers"
          },
          {
```

```
        "code": "14",
        "date": "17 Mar 2015",
        "day": "Tue",
        "high": "26",
        "low": "23",
        "text": "PM Snow Showers"
       },
       {
        "code": "26",
        "date": "18 Mar 2015",
        "day": "Wed",
        "high": "33",
        "low": "29",
        "text": "Cloudy"
       }
      ],
      "guid": {
       "isPermaLink": "false",
       "content": "USAK0170_2015_03_18_7_00_AKDT"
      }
     }
    }   }
  }
}
```