Weather Station System

The Name #3

Ziwei Ye, Hau Khual and Phu Dinh

**Software Requirements Specification**

**Document**

**Version: 1.0 Date: 10/31/2017**

**Table of Contents**

[1. Introduction 4](#_Toc497251616)

[1.1 Purpose 4](#_Toc497251617)

[1.2 Scope 4](#_Toc497251618)

[1.3 Definitions, Acronyms, and Abbreviations. 5](#_Toc497251619)

[1.4 References 6](#_Toc497251620)

[1.5 Overview 6](#_Toc497251621)

[2. The Overall Description 7](#_Toc497251622)

[2.1 Product Perspective 7](#_Toc497251623)

[2.1.1 System Interfaces 8](#_Toc497251624)

[2.1.2 Interfaces 8](#_Toc497251625)

[2.1.3 Hardware Interfaces 8](#_Toc497251626)

[2.1.4 Software Interfaces 9](#_Toc497251627)

[2.1.5 Communications Interfaces 9](#_Toc497251628)

[2.1.6 Memory Constraints 9](#_Toc497251629)

[2.1.7 Operations 9](#_Toc497251630)

[2.1.8 Site Adaptation Requirements 9](#_Toc497251631)

[2.2 Product Functions 10](#_Toc497251632)

[2.3 User Characteristics 11](#_Toc497251633)

[2.4 Constraints 11](#_Toc497251634)

[2.5 Assumptions and Dependencies 11](#_Toc497251635)

[2.6 Apportioning of Requirements 11](#_Toc497251636)

[3. Specific Requirements 12](#_Toc497251637)

[3.1 External Interfaces 12](#_Toc497251638)

[3.2 Functions 12](#_Toc497251639)

[3.3 Performance Requirements 13](#_Toc497251640)

[3.4 Logical Database Requirements 13](#_Toc497251641)

[3.5 Design Constraints 14](#_Toc497251642)

[3.5.1 Standards Compliance 14](#_Toc497251643)

[3.6 Software System Attributes 15](#_Toc497251644)

[3.6.1 Reliability 15](#_Toc497251645)

[3.6.2 Availability 15](#_Toc497251646)

[3.6.3 Security 16](#_Toc497251647)

[3.6.4 Maintainability 16](#_Toc497251648)

[3.6.5 Portability 16](#_Toc497251649)

[3.7 Organizing the Specific Requirements 17](#_Toc497251650)

[3.7.1 System Mode 17](#_Toc497251651)

[3.7.2 User Class 17](#_Toc497251652)

[3.7.3 Objects 18](#_Toc497251653)

[3.7.4 Feature 18](#_Toc497251654)

[3.7.5 Stimulus 18](#_Toc497251655)

[3.7.6 Response 19](#_Toc497251656)

[3.7.7 Functional Hierarchy 19](#_Toc497251657)

[3.8 Additional Comments 21](#_Toc497251658)

[4. Change Management Process 21](#_Toc497251659)

[5. Document Approvals 21](#_Toc497251660)

# 1. Introduction

* A national weather service wishes to collect weather information from remote areas to help with weather forecasting, forecast accuracy assessment and climate change modeling.
* Currently, limited collections are made manually by people visiting remote stations every day.
* The intention therefore is to develop remote automatic collection systems that are connected to a broader weather information system.

## 1.1 Purpose

* The purpose of the Software Requirements Specification document is to clearly define the system under development, namely Wilderness Weather Station System.
* To help monitor climate change and to improve the accuracy of weather forecasts in remote areas, the government of a country with large areas of wilderness decides to deploy several hundred weather stations in remote areas.
* These weather stations collect data from a set of instruments that measure temperature and pressure, sunshine, rainfall, wind speed and wind direction.
* Therefore, the audience will understand how weather station will give them all information about weather.
* This SRS will be entirely design independent, focusing only the end requirements that the final software must achieve.

## 1.2 Scope

To help monitor climate change and to improve the accuracy of weather forecasts in remote areas, the government of a country with large areas of wilderness decides to deploy several hundred weather stations in remote areas.

1. **Monitor** the instruments, power, and communication hardware and report faults to the Management system.
2. **Manage** the system power, ensuring that batteries are charged whenever the environment conditions permit but also that generators are shut down in potentially damaging weather conditions, such as high wind.
3. Allows for dynamic reconfiguration where parts of the software are replaced with new versions and where backup instruments are switched into the system in the event of system failure.

The weather station system is responsible for collecting weather data, carrying

out some initial data processing and transmitting it to the data management system.

The data management and archiving system collect the data from all of

the wilderness weather stations, carries out data processing and analysis and

archives the data in a form that can be retrieved by other systems, such as weather

forecasting systems.

*SatComms: sends message to weather Station via satellite link to create a summary of the collected weather data*

The station maintenance system can communicate by satellite with all

wilderness weather stations to monitor the health of these systems and provide

reports of problems. It can update the embedded software in these systems. In the

event of system problems, this system can also be used to remotely control a

wilderness weather system.

*Commslink is calls the summarize method in the object weatherdata and waits for a reply*

## 1.3 Definitions, Acronyms, and Abbreviations.

**Definitions**

|  |  |
| --- | --- |
| **TERM** | **DEFINITION** |
| Actors | An actor is any outside entity that interacts with the system |
| System | A group of related hardware units or programs or both |
| Use Case | A use case is a list of actions or event steps |
| User | Any person who uses the program |
| Administrator | A person who has administrative access to Weather station advanced settings system |
| Interface | An equipment or program for communication or interaction |
|  |  |

**Acronyms/Abbreviations**

|  |  |
| --- | --- |
| **Term/Acronym/Abbreviation** | **Expansion/Description** |
| HTTP | Hypertext Transfer Protocol |
| SRS | Software Requirements Specification |
| GUI | Graphical User Interface |
| HTTPS | Hypertext Transfer Protocol Secure |
| TCP | Transmission Control Protocol |
| IP | Internet Protocol |
| SFTP | Secure File Transfer Protocol |
| IDE | Integrated Development Environment |
|  |  |

## 1.4 References

Case Study - Wilderness Weather Station

<https://canvas.northseattle.edu/courses/1487576/pages/case-study-wilderness-weather-station?module_item_id=25095328>

Lists of network protocols

<https://en.wikipedia.org/wiki/Lists_of_network_protocols>

Satellite Data Access by Satellite and Instrument

<https://www.ncdc.noaa.gov/data-access/satellite-data/satellite-data-access-sorted-satellite-instrument>

## 1.5 Overview

The Overall Description (section 2) describes the background and general factors about this weather station system for users; Specific Requirements (section 3) contains all detailed software requirements for developers; Change Management Process (section 4) shows the policy of managing changes for developers; Document Approvals (section 5) logs the approvers of the SRS document.

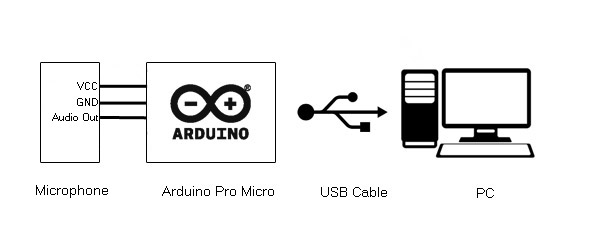
# 2. The Overall Description

This system which is a part of a larger system (weather information system), is responsible for collecting weather data, carrying out some initial data processing and transmitting it to the data management system.

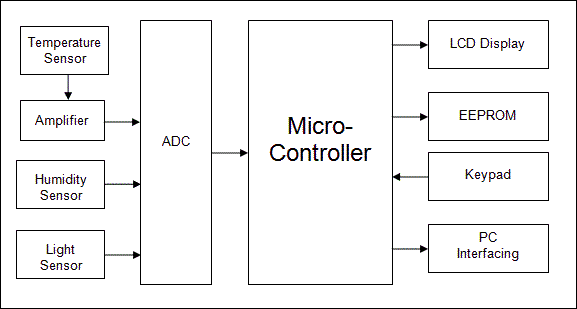
## 2.1 Product Perspective

For personal use, users can use mobile phone and apps to access this system to get the weather forecast information. For professional use, it can generate detailed reports to support analysts’ and scientists’ research.

The related research is how software and hardware talk together and how they can be configured from software and can be controlled by interface.



VS



### 2.1.1 System Interfaces

**Sensor Modules:** It measures temperature, pressure, sunshine, rainfall, wind speed, and wind direction.

**Monitoring System (Software System):** It take parameter readings periodically and manages the data collected from Sensors.

**Solar Charging Systems:** Each weather station is battery-powered therefore, solar supplies the power needed to charge the battery in wilderness Weather Station System.

**Control Panel:** Allows the User to control wilderness Weather Station System via Satellites.

### 2.1.2 Interfaces

**Built-in Panel:** A GUI interface, allows technicians to control and troubleshoot the station on site.

**Remote Control:** A GUI interface, where Admins can manage, configure, monitor the station online.

### 2.1.3 Hardware Interfaces

The system has below hardware interface requirements:

Anemometer – wind speed measurement

²Barometer – air pressure measurement

²Ground and air thermometers

²Rain/precipitation gauge

²Sunshine gauge

²Visibility gauge

When we build weather station, we will worry much about collecting weather

information from instruments at regular interval and transmitting this information, on

request, to the weather information system over the satellite link. The software will take

control and configure with features we want interface hardware to do. (e.g. I want wind

speed measurement control 24/7 and send the info and show in software.)

### 2.1.4 Software Interfaces

The Software Interfaces that will be used are Eclipse and SQL Server. The Java

would be primary Language used to design the Software Interface. Using SQL Server

to collect and store weather station system data.

### 2.1.5 Communications Interfaces

TCP/IP, SFTP, HTTP, HTTPs.

### 2.1.6 Memory Constraints

The Wilderness Weather Station System will need 128MB of RAM or higher.

### 2.1.7 Operations

**The various modes of operation of this software would be:**

1. Collecting weather observations at frequent intervals-for example temperatures are measured every minute.
2. Because of the bandwidth to the satellite is relatively narrow therefore, the weather station carries out some local processing and aggregation of the data.
3. It then transmits this aggregated data when requested by the data collection system.
4. If connection cannot be established, then the weather station maintains the data locally unit communication can be resumed.

### 2.1.8 Site Adaptation Requirements

It must be exposed outside, installed on a solid surface.

## 2.2 Product Functions

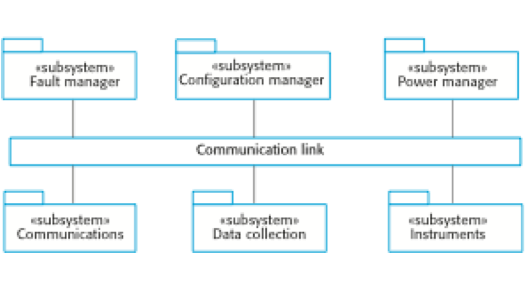
1. *The major function that weather station will use is*

*fault manager - configuration manager --- power manager*

*communication link*

*communications --- data collection ----- instruments*

*(2)*

**

*data collection:*

*transmitter + Receiver*

*weatherdata*

Once interactions between the system and its environment have been understood, this info is used in designing the system architecture.

*fault manager -- will connect to co to talk communication link*

*communications - broadening messages on a common infrastructure receives a control command such as shutdown the command picked up by each of other subsystem which then shut themselves down in correct way*

*data collection -- will collection what it has been communication between subsystem*

*power manager --- will control the power*

*configuration manager will set all function will let them do that way*

## 2.3 User Characteristics

The intended users of the product presumably will have basic knowledge of weather station system.

The users of the program however will not be expected to have technical expertise with software in order to be comfortable and productive with the product.

**Local Database:** It stores wilderness weather data that could be retrieved via Satellite. This reduced the workload needed to retrieve weather data source from the physical device.

## 2.4 Constraints

Permission to access related systems’ data and functions;

Accessibility to the satellite;

Environmental conditions.

## 2.5 Assumptions and Dependencies

* Staffing: We do not have a release/delivery manager and those duties are performed by project manager, which is a staffing constraint.
* Additional open source contributors are welcome, but are not assumed.
* Budget: Personally financed
* Schedule: Development efforts are constrained by "day-jobs".

## 2.6 Apportioning of Requirements

The basic functions like Sensors, Data Management, archiving system, and wilderness Weather Station should be implemented first in order to construct the general structure of the Wilderness Weather System.

The assumed minimal goals of the project will be to have a Weather Station System install and running which can interact with an emulated client that will be running on a computer.

# 3. Specific Requirements

## 3.1 External Interfaces

* Wing measure
* to measure of how fast wing can go
* input to weather info system via wire or wireless
* local 20 mile
* mph
* 24/7 real-time update
* Relationships to weather info system only
* format via number
* we will use software to get data
* use signal to send message to system
* temperature measure
* to measure of how hot or cold in local
* input to weather info system via wire or wireless
* local 20 mile
* F or C
* 24/7 real-time update
* Relationships to weather info system only
* format via number
* we will use software to get data
* use signal to send message to system

## 3.2 Functions

**Weather Station System**

Description: The system shall collect weather data, carrying out some initial data processing, and transmitting it to the data management system.

**Sensors**

Description: These sensors shall measure temperature, pressure, sunshine, rainfall, wind speed, and wind direction in the wilderness Environment.

**Control and Monitoring**

Description: The environmental sensors shall be remotely controlled by station maintenance system via Satellite connection. It shall also provide operational status of sensor equipment to the maintenance system.

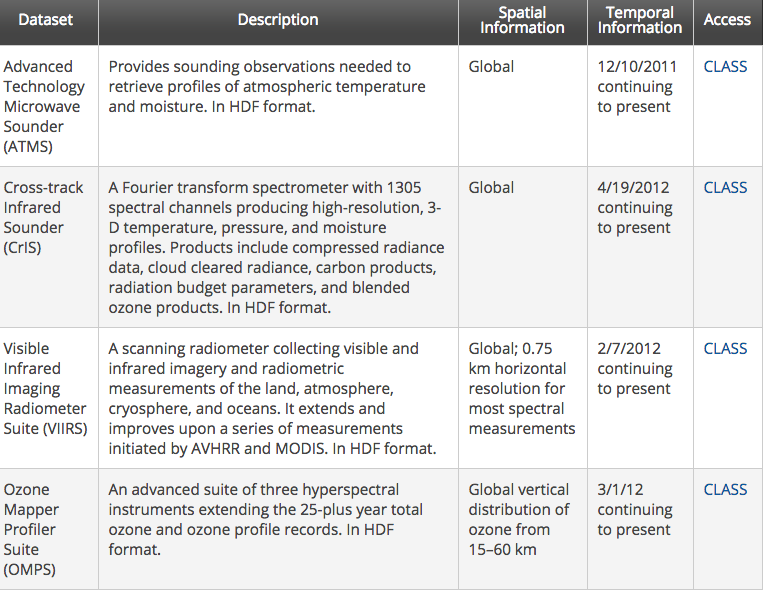
## 3.3 Performance Requirements

* 100TB of primitive weather data can be stored locally.
* 80% of data uploads should be finished in 5 minutes.
* 90% of remote control delay should be less than 1 second.
* Connection should be checked every hour.

## 3.4 Logical Database Requirements

The information that will be put into the database are the photos (if the system has progressed that far) and the database would be accessed daily if the photos are stored.

* Types of information used by type of info we want report like ATMS / CRLS like picture bellow
* frequency of 434 MHz
* Accessing capabilities via MySQL database
* Data entities and their relationships like text / number they will combine together when interface send data and this will save as text or number
* System they use
* Data retention requirements is Number / text / picture / graphic color line

**

## 3.5 Design Constraints

Hardware Limitations: the system has limited RAM to process the given data.

Computations cannot be completed before new data arrive.

Limit of the language features used in the design; we will use Java language for the design.

### 3.5.1 Standards Compliance

* Report format must be the same one used in its parent system, Weather Information System.
* Data naming must be in accordance with its parent system, Weather Information System.
* Financial information has to be recorded in a encrypted file with expected and actual values.

## 3.6 Software System Attributes

Product characteristic Description Maintainability Software should be written in such a way so that it is easy to change, to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment. Dependability Software dependability includes a range of characteristics including reliability, security and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system. Efficiency Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilization, etc. Acceptability Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable and compatible with other systems that they use.

### 3.6.1 Reliability

Factors required to establish the required Reliability of the Software System.

* Assess customer’s situation
* develop goals
* write program plan
* execute program plan

It must be possible for the system to reconfigure itself in order to continue observing, in a different mode if required, given the failure of a single non-critical subsystem.

### 3.6.2 Availability

This system which runs 24/7, allows users to restart or manage versions after Admin granted the permission.

### 3.6.3 Security

|  |  |
| --- | --- |
| Dependability and Security | Software dependability includes a range of characteristics including reliability, security and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system. |

### 3.6.4 Maintainability

This system can communicate by satellite with all wilderness stations to monitor the health of these systems and provide reports of problems. It can update the embedded software in these systems. in the event of system problems, this system can also be used to remotely control a wilderness weather system.

### 3.6.5 Portability

Around 20% of components with host-dependent code;

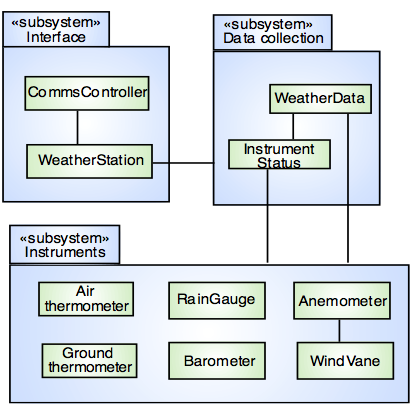
Around 35% of code that is host dependent;

Programing language: Java

IDE: Eclipse

Operating system: Windows 10

## 3.7 Organizing the Specific Requirements

**

### 3.7.1 System Mode

Custom controller and custom extension always run in System Mode while ignoring user’s permission.

### 3.7.2 User Class

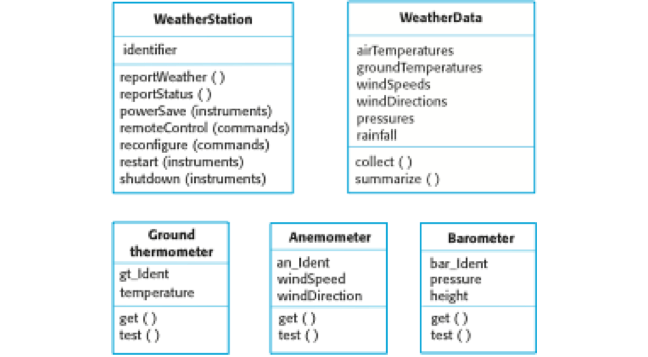
**Admin**: View Data, Monitor, Manage, Configure, Permission Control

**Technician**: Manage, Monitor, Configure

**Viewer**: View Data, Manage

### 

### 3.7.3 Objects



### 3.7.4 Feature

**Input Controls**: buttons, text field, checkboxes, radio buttons, dropdown lists, list boxes, date field

**Navigational Components**: breadcrumb, search field, tags, icons

**Informational Components**: tooltips, progress bar, notifications, boxes

**Containers**: accordion (Graphical Control element)

### 3.7.5 Stimulus

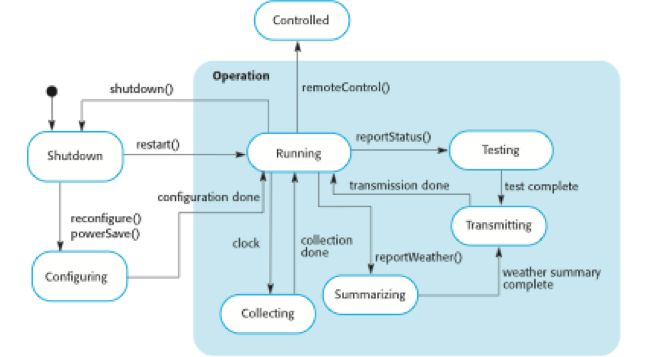
**View Data:** Specific time range and target data types

**Manage:** Specific item and corresponding action

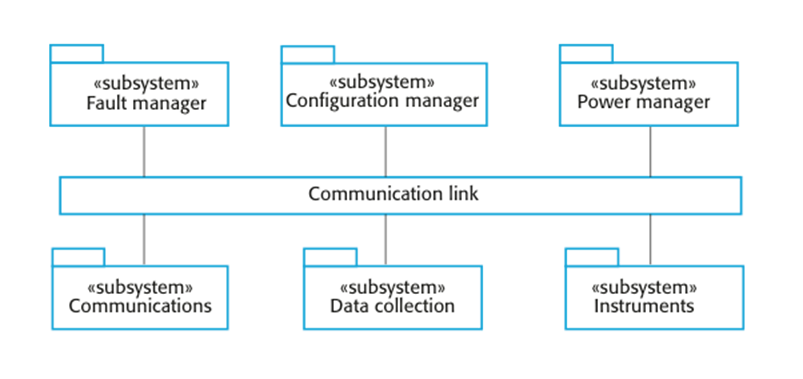
**Configure:** Specific part of the system and version ID

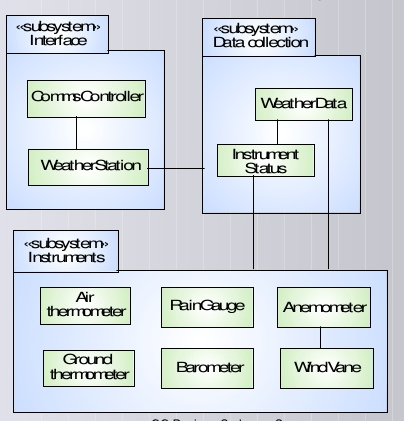
**Permission Control:** User ID and specific permission

### 3.7.6 Response

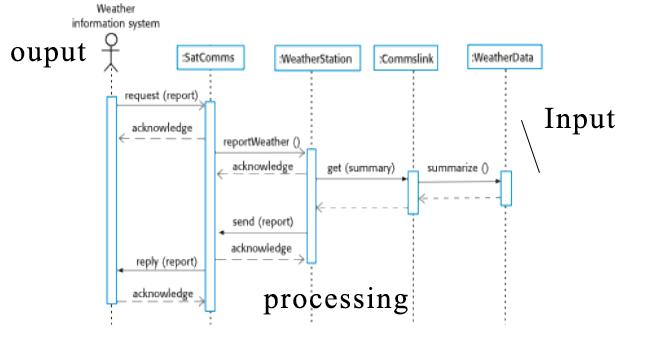
**

### 3.7.7 Functional Hierarchy

****

****

## 3.8 Additional Comments



# Change Management Process

* Changes to the requirements has to get a team consensus
* Customer asks for something new that needs to communicate with the team lead
* Formally in email

# Document Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Phu Dinh | Lead Software Eng. | 10/31/2017 |
|  | John Steger | Instructor, AD300 |  |
|  |  |  |  |