Water Weather Stations
Software Requirements Specification

Harrison Saliba and Daniel Shtunyuk

February 5, 2021

Version: 1.0

Ver.	Date	Who	Change
1.0	11-6-20	D.S. & H.S.	Initial Document

1 Introduction

1.1 Purpose

This document will provide the required specifications for the Weather Water Station project. It will explain the interaction between devices and the function of every device.

1.2 Document Conventions

The acronym *WWXS* stands for Water Weather Stations. The term *Buoy* will reference the hardware unit that houses the processor and sensors. The term *Device Interface* will reference to the mobile device software application used on an Android device. The term *Web Interface* will reference the Front-end website that runs on a web server.

1.3 Intended Audience and Reading Suggestions

This document is written with the intent of being reviewed by future WWXS Developers and WWXS Project Managers. **This document is not designed to be a user guide** and is only meant to be used to evaluate the completion of the software.

1.4 Project Scope

This is a continuation of the on-going project with work from a previous team. For the station, we will finish implementing the current sensors and add a new additional sensor to the buoy. The buoy firmware will be updated to have the new sensor implemented, to read the sensor, log the data, and to send it to Device Interface. The Device Interface will need to read and send the data logs, with the new sensor along with the other sensors, to the database with an encrypted connection. For the Web Interface, we will continue to finish up and complete the web app to get it up and running live, with a simple, visual representation of the data logs from the many water weather stations. For the Device Interface, we will implement the communication between the android app and the database which will be storing all the station data logs and making sure the data is transmitted reliably and securely using an encrypted connection.

1.5 References

There is documentation available for the Buoy, Device Interface, and Web Interface, on the sharepoint site:

https://wwu2.sharepoint.com/sites/WWXS

2 Overall Description

The WWXS will offer a unique way of keeping students engaged beyond the end of their camp. The buoy will allow for physical interaction with a device that creates useful data.

With a higher level of user interaction Users will be more likely to retain the skills they have learned at the Sea Discovery Camp.

2.1 Product Perspective

The User should use this device to collect data about Water Weather conditions and practice Cyber-Security skills they have been taught. Users will hopefully see the value in the information they learned at the SEA Discovery Center by interacting with the WWXS. Researchers can use the WWXS to collect data about water conditions around the Pacific Northwest. This data is free and easily accessible though the web interface.

2.2 Product Features

The WWXS product will log sensor data, like the temperature and turbidity of the water that the Buoy is placed in, and allow for the user to access this data by transferring it before hand from the Buoy to the Device Interface, then storing the data in the Web Interface. The Web Interface will allow to view the sensor data collectively and give the user the option to export the data to use it else where.

2.3 User Classes and Characteristics

Students are the students that have taken camps at the SEA Discovery Center and are the primary subject for the software as their role is critical to its effectiveness at collecting data. *Researchers* are users that will only interact with the data collected by the Students class. Their main role is the consumer of the information offered by our product.

2.4 Operating Environment

The environment that the Buoy will be used is the outdoors, in an area with access to water. The Device Interface will be used outdoors near the buoy as it receives data from the Buoy and uploads it to the web server. The campers and researchers will be accessing the data on the Web Interface through a web browser on a computer or mobile device.

2.5 Design and Implementation Constraints

The buoy must cost less than $50 to source. The device must adhere to Bluetooth Low-Energy (BLE) standards and subsequent radio transmission standards for communicating with the Device Interface and will most likely use a predefined library for this communication.

2.6 Assumptions and Dependencies

The Device Interface software is running on an Android OS device. The Android device is assumed to have basic internet access, in order to upload the data from the Buoy to the web server. The web server is running on a Unix-based operating system.

3 Features

3.1 Web Interface

3.1.1 Description

The web interface will be a Graphical User Interface accessible with any modern internet browser (i.e. Edge, Safari, Chrome, Firefox). It will display the information collected by the Buoys and allow for the downloading of that information by researchers. It will also allow for the sharing of keys to connect to the buoys from one user to another given permission.

3.1.2 Priority

This is a primary feature of the WWXS project. It is a high priority task, as the Web Interface is the portal to access the information collected by the Buoys. But is less important than being able to connect to the Buoy with the Android app and the data collection on the Buoy, as these systems can function independently of the web interface.

3.1.3 Stimulus and Response

The user visiting a URL and will be greeted with a landing page.

3.1.4 Functional Requirements

The landing page should take less than 200ms to load not including DNS Requests. The entire site should be easy to navigate with a consistent menu bar and easily identifiable icons. It should allow the user to view and sort weather data by: source Buoy, date, and user.

3.2 User Authentication

3.2.1 Description

The Web Interface will have user accounts for the use of allowing access to Buoys. Each user should be able to be assigned a key or made the owner of it so they can share access to a Buoy.

3.2.2 Priority

This is a medium-priority task as it affects the primary user base but doesn't subtract from the current state of the project.

3.2.3 Stimulus and Response

The user will input some form of identification and a secret password to authenticate a web session.

3.2.4 Functional Requirements

The user authentication process should have a reasonable about of security requirements. Such as minimum password requirements, password hashing, and encryption. Users accounts should also only be created by an authorized user.

3.3 Database

3.3.1 Description

The database will contain all the collected buoy data, user accounts, and access structure.

3.3.2 Priority

This is a high-priority task because both the Device Interface and the Web Interface require it to be functional to effectively and securely store and retrieve data.

3.3.3 Stimulus and Response

The Web Interface will send formatted requests to the database for data and the database will respond with the information in the request. The Device Interface will do the same.

3.3.4 Functional Requirements

The Database will be constructed using an ER Diagram and will be at least BCNF 5.

3.4 Exporting Data

3.4.1 Description

The Web Interface will allow the download of raw buoy data in a commonly used file format such as CSV. This should be able to be sorted, then downloaded so a user can download data from: a single Buoy, a certain User, or over a specified period of time.

3.4.2 Priority

This is a relatively low-priority feature as it does not affect our most important user base and doesn't affect the collection of data.

3.4.3 Stimulus and Response

A user specifies the data they want. Then a file is downloaded from the Web Interface.

3.4.4 Functional Requirements

The compilation of the data should either:

A. Take less than 300ms to start downloading. or B. Show a progress bar or status indicator to provide the user with feedback that the data is being compiled before download.

3.5 Multi-user access to Buoy

3.5.1 Description

The Buoy will allow for different users with Android devices to establish connection and have access to a singular Buoy and its data. The Buoy currently allows for just a single Android device to establish and maintain exclusive rights to connect and access the Buoy. No other Android device can establish connectivity with the Buoy once the first Android device has established connection with the Buoy.

3.5.2 Priority

This is a medium priority feature as it has been requested since some users may try to connect to a Buoy that is restricted to one specific Android device, yet it could be the same user that is attempting to connect.

3.5.3 Stimulus and Response

A user with specific privileges and keys accesses a specific Buoy and establishes a connection to it if they have the correct privileges. A user can set access to different Buoys through the Web Interface, which will update the user's set of accessible Buoys.

3.5.4 Functional Requirements

The user account shall have designated access given to the Buoy through the keys and group permissions set through the Web Interface. The user account on the Device Interface shall have a list of keys and access privileges for a certain set of Buoys. The Buoy shall have a specific key that is used to check the against with the keys given by the user on the Device Interface. The software shall fetch the list of Buoys the user can connect to from the Web Interface to the Device Interface.

3.6 Optimizing Power Usage of Buoy

3.6.1 Description

Measuring and testing changes with the sleep timer and the elapsed time to capture data from the sensors, and testing which values and optimizing the code can help increase the battery life. The Buoy's battery life can allow for the Buoy to capture data for long periods of time without needing to replace the batteries and missing moments when data could be captured.

3.6.2 Priority

This is a low priority task. The SoC we are currently using has features to use a low power state in between times of capturing data, so there is very little power drain from the battery.

3.6.3 Stimulus and Response

When the Buoy is powered on, when it is capturing data from the sensors, and when it is in a low power state in between capturing data, the amount of power necessary to run these tasks changes and uses more battery power.

3.6.4 Functional Requirements

The Buoy shall capture data when it wakes from the low power state to going back to low power state within 2000ms and shall transfer the logged data from the last 24 hours within 1000ms from the Buoy to the Device Interface. The Buoy should last at least 2 weeks before the battery power is depleted.

4 External Interface Requirements

4.1 User Interfaces

4.1.1 Device Interface

The WWXS will include a reasonably intuitive user interface with the purpose of allowing the user to create a secure connection to the Buoy and retrieve data. Then allow for the uploading of that Data to the database.

4.1.2 Web Interface

The Web Interface will display a visually appealing data vs. time display and options for downloading the data. It will also include a way of accessing a users account information and giving permissions to use keys for specific buoys.

4.2 Hardware Interfaces

4.2.1 Buoy

The Buoy does not currently have any hardware interface features. The unit is always-on once the battery is connected.

4.3 Software Interfaces

4.3.1 Database

The database will be a standard database interface with query language specific to its implementation. It should have the basic ability to add, remove, and update information contained within the Database.

4.3.2 Web API

The Web API will include functions that allow the Web Interface to access the database. It should only allow the features specific to the project to prevent mis-use of the database by the public.

4.4 Communications Interfaces

4.4.1 Buoy Bluetooth API

The Buoy will communicate with the Android device utilizing the Bluetooth Low-Energy (BLE) protocol. The Device Interface will send commands to the Buoy and the Buoy will process and run the command requests.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

The Buoy should process and complete requests given to it by the Device Interface within a reasonable amount of time. For example, the Buoy should return the latest data log within 2000ms from the time the data is requested. As for the Buoy battery performance and efficiency, the project currently does not have any set figures on how long the battery should last, but since the Buoy uses replaceable AA-size batteries, we will aim for a run-time of 2 weeks on a single pair of batteries before they are depleted of power. The Web Interface should load the homepage in less than 500ms under a reasonable wired network condition.

5.2 Safety Requirements

We do not expect any possible safety issues in the operation of the WWXS. Our device includes no physical movement and uses a preexisting form factor of a water bottle that can be reasonably handled by any individual.

5.3 Security Requirements

The Database will not be communicated with directly from any user interface without the use of a secure API. User information should be minimal and passwords should at least be hashed and not stored in any un-encrypted form. We recommend the use of SSL Certificates such that the Web Interface uses HTTPS but this may be beyond the scope of the project as it can be cost-prohibitive and the user data contains no personal information beyond a username and password.

5.4 Software Quality Attributes

- The Web Interface should allow users to view data in less than 2 interactions.

- The Web Interface should handle all HTTP errors with a visually descriptive error page.
- The Device Interface should display useful information on its connection status with both the Buoy and Web Interface so it is both engaging and easy to diagnose issues.
- The Device Interface should be able to receive and parse data from older versions of the Buoy code along with new versions of the Buoy code.
- The Device Interface should be able to receive and parse data from Buoys with newly added sensors along with Buoys that have less sensors.
- The Device Interface should show a visual warning message if the data received from the Buoy is incomplete or corrupted.
- The Buoy should wake from the low-power sleep state, read and log all the sensor data, and go back to the low-power sleep state under 2000 ms.

A Glossary

BLE : Bluetooth Low Energy
DNS : Domain Name Server
CSV : Comma Separated Values
BCNF : Boyce-Codd Normal Form
SoC : System-on-Chip

B Analysis Models

No models or diagrams implemented.

C Issue List

None currently.