System Requirements Document 1.0
Client: Daintree (Group 7)
Project: Ocean Data Aggregation

Horizon Software (Group 2)

Braiden Cutforth, Daelyn Jones, Graham Grayson, Joel Kerfoot, Julian Rocha,
Rafay Chaudhry, Sarah Thomasgard, Slater Gordon

February 5th 2019

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Initial Document	Jan. 30 2019	Transcribing RFP and elicitation notes to a structured document.	0.7
Formatted Sections	Feb. 1 2019	Formatting notes into sections and filling out more info.	0.8
First Complete Draft	Feb. 4 2019	All sections complete, ready for team review.	0.9
Final Copy	Feb. 5 2019	All sections complete and reviewed, ready for client reading.	1.0

1 Introduction

1.1 Purpose

This document describes the requirements needed for the Daintree Ocean Data Aggregation Project. The purpose of the project is to improve the accessibility of the data collected by Daintree. Currently, the lack of a unified solution for data retrieval and visualization hinders user access to Daintree's wealth of ocean data. This project will result in the creation of a new data access system for Daintree's users.

1.2 Project Scope

The Daintree Ocean Data Aggregation Project has two main objectives. The first objective is to visualize the raw data collected by Daintree and made available through their API. Currently when a customer asks for new visualized data, Daintree must develop a unique visualization solution. A centralized system to generate visualizations would benefit Daintree by reducing cost and time spent developing unique solutions for their customers. The second objective is to make these visualizations easily accessible to all users through a single interface. Making these visualizations accessible would help more people make use of Daintree data and would align with Daintree's social business goals of supporting ocean education and public awareness.

The project will look at the current system and the way that it allows users to visualize raw data. The project does not need to change the way the raw data is retrieved from sensors, how it is stored in the database, or how it is accessed from the database.

1.3 Glossary of Terms

Application Programming Interface (API)	A set of functions that programmatically access the data stored in Daintree's database.
Data	Raw data that has been processed by the system into a visualization.
Database	A store of raw data that is accessed through the Daintree API.
Free User	A user who accesses Daintree's data free of charge.
Horizon Software	A team of dedicated systems analysts and developers in the Software Engineering and Computer Science programs at the University of Victoria.
Sensor	A data collection device that feeds a stream of raw data into Daintree's database.
To Be Determined (TBD)	Awaiting feedback from Daintree.
Paid User	A user who pays for increased access to Daintree's data.
Raw Data	Data received directly from the API.
Visualization	A presentation of raw data in some manner.

1.4 References

[1] Daintree Request for Proposal, [Online]
Available: https://sites.google.com/view/daintree-company/documents

[2] Daintree Ocean Division, [Online]
Available: https://sites.google.com/view/daintree-company/home

[3] Daintree Elicitation notes, [Online]
Available: https://kerfootj.github.io/seng321_designer/docs/Daintree%20Elicitation%20Notes.pdf

[4] Material Design, [Online]
Available: https://material.io/

[5] API Documentation [Online]
Available: https://wiki.oceannetworks.ca/display/O2A/Oceans+2.0+API+Home

1.5 Overview

This document contains six sections and an appendix. The second section includes an overview of the current system. Main product features are described along with project stakeholders, user classes and assumptions about Daintree's API. The third section describes the required system features for our proposed solution. The fourth section describes any interfaces that must exist for the user to effectively interact with the system. The fifth section outlines non-functional requirements for the system. The final section includes additional requirements.

2 Overall Description

2.1 Product Perspective

Daintree maintains a network of sensors that measure various ocean conditions such as salinity, acidity and temperature. The measurements are collected, processed and stored as raw data in Daintree's database. The raw data stored in Daintree's database is accessible through the API. The existing system for creating visualizations involves making a custom application for each customer which is inefficient and costly for Daintree.

The system currently being developed by Horizon Software is a replacement for Daintree's existing system for data visualization. The new system will provide a single interface where Daintree raw data can be processed and viewed. With the new system, no more custom applications will need to be developed.

The primary stakeholders for the Daintree Ocean Data Aggregation Project include Daintree, the users of the new system and the Horizon Software development team. Other potential stakeholders include research & educational organizations, ocean sensor suppliers, and environmental agencies.

2.2 Product Features

The major features of the new system being developed by Horizon Software can be summarized as follows:

- All users must be able to view visualizations of raw data.
- All users must be able to search visualizations by location, sensor type, and time range.
- All users must be able to filter visualizations by location, sensor type, and time range.
- All users must be able to save the visualizations.

2.3 User Classes and Characteristics

There are two user classes for the new system: Free User, and Paid User. The user classes are outlined below.

2.3.1 Free User

Free users are not paying for access to Daintree's data. These users typically consist of anyone who is curious about Daintree's data. Free users have limited access to Daintree's data. Free users may wish to save visualizations for future reference. It is assumed that the free user group lacks technical software knowledge.

2.3.2 Paid User

Paid users consist of users who pay to use the system. Unlike free users, paid users have access to all Daintree data. Paid users will be able to view annotations made by Daintree on videos and images. The paid user group could include the general public, students, researchers, and businesses. Paid users require the same functionality as free users to save visualizations.

2.4 Operating Environment

The system will request raw data from Daintree's database through the Daintree API. The new system must therefore conform to the specifications for the Daintree API requests.

The users of the system are expected to access the interface through their personal devices. The system interface will therefore need to be accessible on mobile and desktop environments, supporting the major operating systems: Windows, Mac OS, Linux, iOS, and Android.

2.5 Design and Implementation Constraints

Daintree was very flexible regarding the design and implementation of the final system. The only constraint on the system is that it must be built on top of Daintree's existing API:

C-1: System must request raw data from Daintree's existing API.

2.6 Assumptions and dependencies

The system will be based on the following assumptions and dependencies as outlined below.

2.6.1 Assumptions

In order to generate reliable visualizations, the raw data sent to the system from Daintree's API must be reliable. If the raw data contains inconsistencies, the visualization being generated will be inconsistent. If the raw data contains errors, the visualizations will contain errors. It is not the responsibility of the new system to notify Daintree of errors or correct these errors.

A-1: The raw data provided by the Daintree API is free of inconsistencies and errors.

2.6.2 Dependencies

The system needs to access Daintree's data through the API. It is critical that the API is always available, and the API calls must not change often. Should a change to the API specification occur, the new system must be manually changed to match the new specification.

D-1: Daintree's API is robust and consistent.

The system will expect data returned from Daintree's API to be in specific formats. The system must be built to accept data formats currently provided by the API. Any new formats provided by the API will need to be added to the system manually.

D-2: The system depends on Daintree's API responding with consistent data formats.

3 System Features

The system will include the following features: data visualization, to present the raw data collected by Daintree in a meaningful way; filtering, to allow a user to refine the data; searching, to allow a user to search for the data and visualizations; and save history, to allow a user to save and return to specific visualizations.

3.1 Data Visualization

3.1.1 Description and Priority

The visualization of raw data is how a user can view and comprehend Daintree's collection of data. Without an effective way to visualize the raw data, the user will be easily confused. The user must be able to visually compare different sets of raw data. Viewing multiple sets of raw data overlaid on the same visualization can help users recognize relationships between data. These relationships and insights promote education and awareness.

Priority: High

3.1.2 Functional Requirements

- REQ-VI-1: The system must produce and display visualizations of Daintree's raw data.
- REQ-VI-2: The system must play the videos collected by Daintree.
- REQ-VI-3: The system must display the images collected by Daintree.
- REQ-VI-4: The system must display annotations on videos and images when available.
- REQ-VI-5: The system must provide the user with the option to include up to four sets of raw data overlaid in the same visualization.
- REQ-VI-6: The system must provide the user with the option to view multiple visualizations side by side on the same page.

3.2 Filter

3.2.1 Description and Priority

Filtering data is an important part of the way that data is utilized. Filtering is how users can select data they would like to view. Users must have the option to select as many or as few data sets as they would like to view at one time. Filtering data is imperative to helping users comprehend and view data relevant to their research or studies. Without filtering, there would be an immense amount of data shown at one time and users would be unable to distinguish between different data.

Priority: TBD

3.2.2 Functional Requirements

- REQ-FI-1: All users must be able to filter data by time.
- REQ-FI-2: All users must be able to filter data by sensor type(s).
- REQ-FI-3: All users must be able to filter data by location(s).
- REQ-FI-4: All users must be able to filter data by any combination of time, location, or sensor type simultaneously.

3.3 Search

3.3.1 Description and Priority

Searching for data from a certain sensor, location or time is essential for users who need to find specific pieces of data. Since the new system needs to be based around users utilizing data for their research and studies, it is critical that users can find the data that they need.

Priority: TBD

3.3.2 Functional Requirements

- REQ-SE-1: All users must be able to search for data collected from specified location(s).
- REQ-SE-2: All users must be able to search for data collected from specified sensor(s).
- REQ-SE-3: All users must be able to search for data collected from a specified timeframe.

3.4 Save

3.4.1 Description and Priority

Free users and paid users must be able to save visualizations. This is important because if a free user finds a visualization of interest, that user desires the ability to easily reference it at a later time.

Priority: TBD

3.4.2 Functional Requirements

- REQ-UA-1: All users must be able to save specific visualizations.
- REQ-UA-2: Free and paid users must be able to display their saved visualizations.

4 External Interface Requirements

4.1 User Interfaces

The following sections describe the characteristics of user interfaces for free and paid users.

4.1.1 Free User Interface

The free user interface is accessed by free users. The free user interface must meet the requirements of the system listed in Section 3. The free user interface must only allow access to the limited Daintree data that is provided to a free user. No user interface requirements or guidelines were provided but a recommended guideline to follow is Google's Material Design.

4.1.2 Paid User Interface

The paid user interface is accessed by paid users. The paid user interface must have the same functionality as the free user interface. In addition, paid users must have access to all Daintree data and must be able to keep track of their payments, including billing information. No user interface requirements or guidelines were provided but a recommended guideline to follow is Google's Material Design.

4.2 Communications Interfaces

The Daintree API provides a means of programmatically accessing the data stored in Daintree's database. The system will access the discovery service to find sensors or locations from which to pull data. The system will access the delivery service to retrieve data and data products.

Figure 1: API Data Flow [5]

Communication with the API occurs over Hypertext Transfer Protocol (HTTP). Because of the large amount of data that can be transferred, data transfer rates are dependent upon the speed of the Daintree API.

5 Other Non-Functional Requirements

5.1 Performance Requirements

The following section specifies the efficiency requirements that the system must meet.

5.1.1 Efficiency

The system must be capable of handling a potentially large amount of data requested from the API.

- REQ-EF-1: The system must be capable of displaying images and video within 3 seconds.
- REQ-EF-2: The system must be capable of visualizing data within 3 seconds.

5.2 Security Requirements

5.2.1 User Data Security

User data security is important for both user classes. The free user group must have access to only a limited subset of data provided by Daintree. No raw data can be accessed by any user. Payment transactions must be secure so that paid users are comfortable using the system.

- REQ-AS-1: Free users can only access a limited subset of data.
- REQ-AS-2: No user can access the raw data.
- REQ-AS-3: Payment transactions must be secure between the paid user and the system.

5.3 Software Quality Attributes

The following sections outline the quality attributes the system must meet including: adaptability, portability, usability, and reliability.

5.3.1 Adaptability

If new sensors are deployed, the system must adapt to accommodate the new data from the sensors. It is essential that users have access to the new data so that it can be used immediately. It is important that new sensors added by Daintree do not disrupt the system.

- REQ-AD-1: The system must be able to accommodate new sensors added to the system.
- REQ-AD-2: The system must require few to no changes when new sensors are added.
- REQ-AD-3: A user must be able to use data collected by new sensors as soon as it available from the Daintree API.

5.3.2 Portability

In order to support a wider range of user devices the system must run on desktop, tablet, and mobile devices.

- REQ-PO-1: The system must work on desktop and mobile platforms.

5.3.3 Usability

The system needs to provide a level of usability such that users with limited technical knowledge are able to use the system.

- REQ-US-1: New users to the site must be able to find the data they are interested in within 3 minutes.

5.3.4 Reliability

- REQ-RE-1: The system must not modify the data when creating visualizations.
- REQ-RE-2: The system must present the data accurately such that visualizations are repeatable given the same raw data.

6 Other Requirements

6.1 Internationalization

REQ-IN-1: The system must support internationalization, so users can view the system interface in different languages.

Appendix: Issues List

There are currently no issues.