This document is subject to change. The latest version may be found at https://github.com/ashawnbandy/cecs491/tree/master/docs

10/9/2012

CSULB Marine Biology Department Software Project

*Requirements Analysis (Preliminary rev 3)*

Contents

[Introduction 2](#_Toc339462748)

[Purpose of the System 2](#_Toc339462749)

[Objectives 2](#_Toc339462750)

[Definitions 2](#_Toc339462751)

[Functional Requirements 2](#_Toc339462752)

[Usability 2](#_Toc339462753)

[Reliability 3](#_Toc339462754)

[Safety 3](#_Toc339462755)

[Security 3](#_Toc339462756)

[Performance 3](#_Toc339462757)

[Supportability 3](#_Toc339462758)

[Implementation 3](#_Toc339462759)

[Interface 3](#_Toc339462760)

[Packaging 4](#_Toc339462761)

[Use Cases 5](#_Toc339462762)

[Connect to VR2C Receiver (Receiver Life-Cycle) 5](#_Toc339462763)

[Setup email alert(s) 5](#_Toc339462764)

[Start/stop Receiver Hardware 6](#_Toc339462765)

[Open and use a text console for direct interaction with receiver hardware 6](#_Toc339462766)

[Access/query recorded “real time” data 6](#_Toc339462767)

[Store recorded data 7](#_Toc339462768)

[Appendix 1 – System Overview Diagrams (updated 10/09/2012) 8](#_Toc339462769)

[Appendix 2 – Data Flow Diagrams (Added 10/09/2012) 10](#_Toc339462772)

[“Data Browser” Motif 10](#_Toc339462773)

# Introduction

## Purpose of the System

The CSULB Marine Biology department collects data from marine life that has been tagged with an acoustic transmitter. Data will be collected by a receiver located off of Manhattan Beach Pier (MBP), which will record the ID number of tags, associated sensor data, date, and time of detection. The receiver in turn will be connected to a computer through a serial port connection. The purpose of the system will be to interface with this computer remotely in order to control the receiver and receive data from it.

## Objectives

1. Ability to connect remotely to the computer managing the receiver
2. Ability to control the receiver remotely through that connection.
3. Ability to stream real-time data from the receiver.
4. Send email alerts when a detection meeting user-defined criteria is observed.
5. Archive recorded data and recording metadata.

## Definitions

1. “The system” will refer to the software being created by this project, and not the firmware on the receiver equipment.
2. Software connecting to the remote sites (e.g. Manhattan Beach Pier) will be referred to as the “server” or “backend”.
3. Software running on or transferred from csulbsharklab.com will be referred to as the “client” or “front-end”
4. The roles of “User” and “Administrator” is defined by the csulbsharklab.com software and documentation.
5. Real-Time-Mode (RTM) refers to the presentation of data as it comes into the server from the receiver and may or not utilize Real Time Mode 0 on the receiver hardware.

# Functional Requirements

1. Connect to receivers located remotely (e.g. MBP)
2. Start and stop recording data from receiver to server
3. Provide connection to receiver through console
4. Parse and aggregate data sent from receiver
5. Translate data received to SQL commands
6. Send generated SQL commands to csulbsharklab.com automatically for later retrieval by end-users.
7. Email alerts sent out when the system detects user defined parameter

# Non-Functional Requirements

## Usability

* The application will minimize network configuration.
* Additional receiver nodes with up to *N* receivers will also require minimal configuration by a remote operator.
* The front-end user interface will follow design practices in the csulbsharkalb.com. Any server user interface will follow familiar design practices.

## Reliability

* The server should be continuously available. To this end, the system should detect critical faults and reset without end-user administration.
* Non-critical faults will be either logged or reported to a console.

## Safety

There are no known safety requirements.

## Security

* Access to the client will be handled by the existing csulbsharklab.com user validation system.
* Data identified as sensitive will be encrypted when transmitted over open networks (e.g. the internet, the CSULB network, etc.)

## Performance

* Commands to receivers and their effects should be sent and received in near-real-time.

## Supportability

* Sufficient documentation will be provided to the customer to allow for future bug fixes by a third-party.
* The design will be modular and make use of simple declarative language control files (i.e. json) to allow for future changes or additions to the system.

## Implementation

* The server software will be written in C# for deployment on Microsoft Windows machines.
* The server software that interfaces with the hardware will be written as part of this project.
* The server module connecting to the MySQL database will be a standard library from Microsoft and/or Oracle (publisher of MySQL).
* Configuration files will be implemented using a simple declarative language such as JSON (javascript object notation).
* Network communication between remote sites and the server will be handled by serial-over-ethernet hardware devices.

## Interface

* The system will interface with the firmware (current at time of implementation) on each receiver.
* The system will generate data consistent with existing output formats (SQL).

## Packaging

* + Server software will be installed at a site on the California State University of Long Beach campus by a member of the team.
  + All software will also be packaged in a manner that facilitates additional installations.

# Use Cases



|  |  |
| --- | --- |
| M | |
| Name | Connect to VR2C Receiver (Receiver Life-Cycle) |
| Actor(s) | Serial Port (.NET hardware abstraction) |
| Pre-conditions | A running service manager. |
| Flow-of-Control | 1. New serial port is discovered. 2. VR2C receiver is configured 3. Receiver software sends commands and receives messages from the hardware until directed to stop. |
| Post-conditions | Receiver software disconnects from the hardware and is removed from the system. |

|  |  |
| --- | --- |
| Y | |
| Name | Setup email alert(s) |
| Actor(s) | Administrator |
| Pre-conditions | 1. Network connection to server. 2. Administrator has logged with valid credentials |
| Flow-of-Control | 1. The user selects Settings 🡪 Email Alert from menu. 2. The user enters an email address on the text box and clicks add button to enter the email on the email list. 3. The user setup filter for the email alert(s) based on location/receiver/specific tag/time etc. 4. The user clicks save to save the email alert setup. 5. The user clicks red X button on the right side upper corner to close the email setting window. |
| Post-conditions | The server machine sends out email alert(s) to the email addresses on the email list when the trigger condition occurs. |

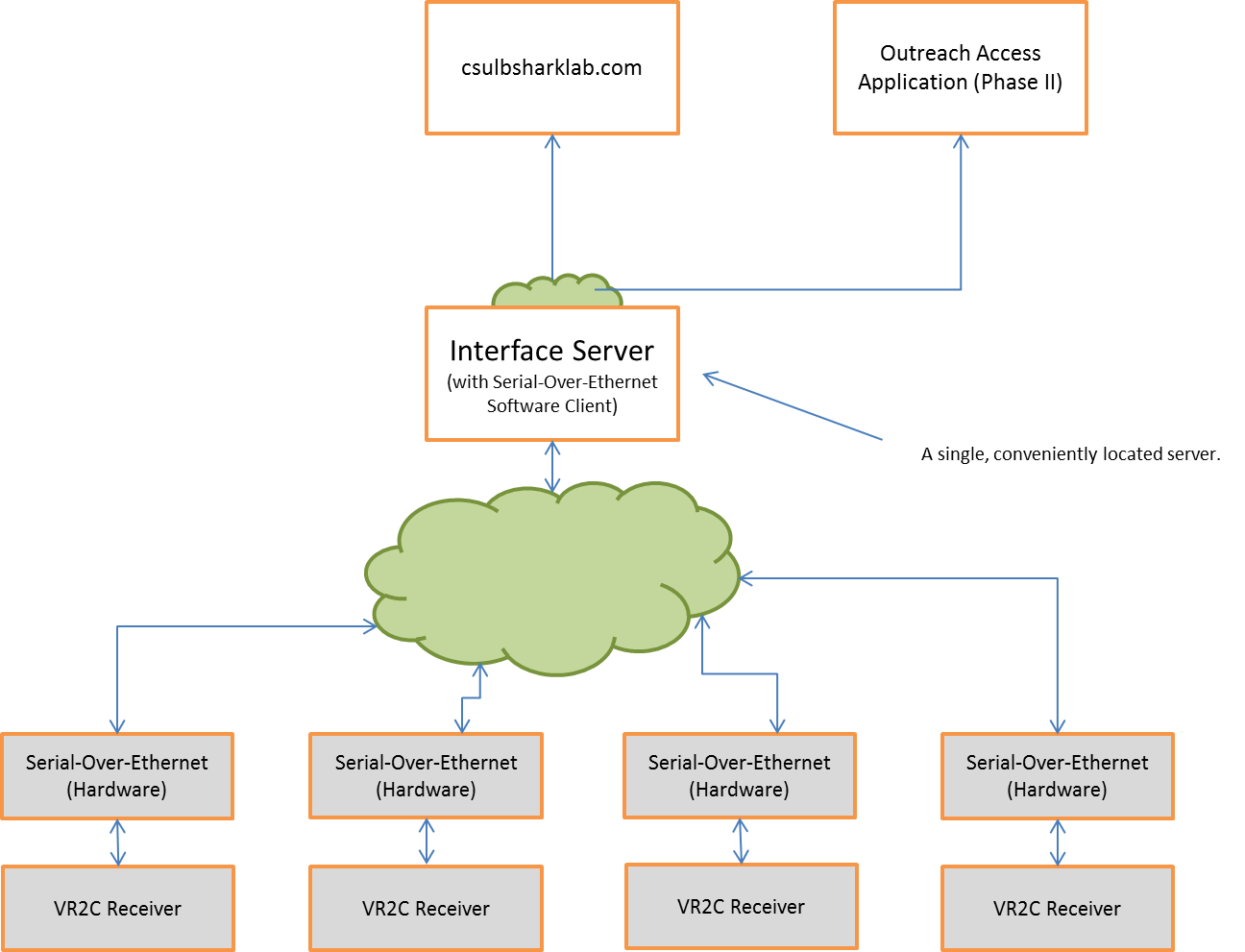
|  |  |
| --- | --- |
| Y | |
| Name | Start/stop Receiver Hardware |
| Actor(s) | Administrator |
| Pre-conditions | 1. An open connection to the receiver hardware to be stopped or started. |
| Flow-of-Control | 1. User chooses to start or stop the receiver on the server user interface. 2. A valid command is sent from the receiver software to VR2C hardware. 3. The hardware begins to send new detections if started, or is placed in storage mode if stopped. |
| Post-conditions | 1. The VR2C receiver has been started or stopped as directed by the user. |

|  |  |
| --- | --- |
| D | |
| Name | Access/query recorded “real time” data |
| Actor(s) | Administrator, User |
| Pre-conditions | 1. Network connection to csulbsharklab.com 2. Csulbsharklab.com is running. 3. User has logged into csulbsharklab.com with valid credentials. |
| Flow-of-Control |  |
| Post-conditions | Query results sent to the browser. |

|  |  |
| --- | --- |
| D | |
| Name | Store recorded data |
| Actor(s) | Hardware Receiver, MySQL database |
| Pre-conditions | 1. MySQL database server must be running and available to the server software. |
| Flow-of-Control | 1. New data arrives from receiver hardware and is properly decoded. 2. When appropriate SQL statements are generated from raw data and sent to remote MySQL database server. |
| Post-conditions | Data has been stored in remote database. |



# Appendix 1 – System Overview Diagrams (updated 10/09/2012)



# 

Main Service Loop

Responds to changes in COM port status.

## 

SQL Databases (e.g. csulbsharklab.com)

Database Module

**VR2C / serial-over-ethernet/COM port**

**VR2C / serial-over-ethernet/COM port**

Encoder

Note: Objects between common dashed lines represent serial executions (common thread of execution).

Parse/Decode Module

Email Notification Module

Event Queue/Dispatcher

Distribute intra-server messages.

Receiver Objects

Sets up and receives data asyncrhonously from VR2C receivers

## Appendix 2 – Data Flow Diagrams (Added 10/09/2012)

# “Data Browser” Motif

Serial Port Stream

VR2C Format

SQL

C# Object

Receiver Software

Database

Decoder

Database Module