Software Requirements Specification

for

Rescue Dog Sensor Project

**Version 1.0**

**Prepared by David Schlerf  
Everett Clary  
Jacob Wingerd**

**Capstone Project UAA Fall 2017**

**9/21/2017**

**Table of Contents**

[**Introduction**](#_n2tad85iqbb8) **3**

[**Purpose**](#_3znysh7) **3**

[**Document Conventions**](#_oa4bxukeek3a) **3**

[**Intended Audience and Reading Suggestions**](#_odh4vkz565og) **4**

[**References**](#_z1hkcqmq9yei) **4**

[**Overall Description**](#_3zqmkf73wlaq) **5**

[**Product Perspective**](#_2s8eyo1) **5**

[**Product Functions**](#_nqlx4oir86u5) **5**

[**User Classes and Characteristics**](#_iuoit6lw5unh) **9**

[**Operating Environment**](#_faubv2qgyc6t) **9**

[**Design and Implementation Constraints**](#_69lhgkjqmdfu) **10**

[**User Documentation**](#_vscti4m81k31) **10**

[**Assumptions and Dependencies**](#_9xi1o57qrn6l) **10**

[**External Interface Requirements**](#_twavdkrufxkn) **11**

[**User Interfaces**](#_x7w8a68r0uf) **11**

[**Raspberry PI:  
 Terminal output for the Technician to read if the device booted and is talking   
 successfully with the server**](#_krrin7ar51eb) **11**

[**Database Server:  
 Terminal based commands to see and query data**](#_hb4brsjtqafs) **11**

[**Receiving Server:  
 Terminal based commands to see and query data**](#_sjgljfecm9rj) **11**

[**Web Interface:  
 Google Maps interface with pins for sound, contacts and path of the dog.**](#_2jxsxqh) **11**

[**Hardware Interfaces**](#_imy1uysd3g4w) **11**

[**Raspberry Pi**](#_snhkxcb48fz5) **11**

[**Microphone array**](#_my8ptkfabvc1) **11**

[**Dry contacts**](#_fmt1fea5yqd1) **11**

[**GPS device**](#_fuvigch84amd) **11**

[**LCD Screen**](#_1g1611ihg57f) **11**

[**Software Interfaces**](#_l7cngpraa77) **11**

[**Communications Interfaces**](#_ym383gtvuvdl) **11**

[**System Features**](#_mnsf1x1hoi4f) **12**

[**Other Nonfunctional Requirements**](#_3whwml4) **15**

[**Performance Requirements**](#_mv8am1b570j) **15**

[**Safety Requirements**](#_vejlsfy6bcba) **15**

[**Security Requirements**](#_mgzet0deb99s) **15**

[**For Raspberry Pi:  
 Currently none at this time**](#_yoe3hl9q8oq4) **15**

[**For Servers:  
 Admin level access required at this time**](#_uarat3yfy2x) **15**

[**For Web interface:  
 Default TCP/IP security**](#_xwrp2qk1ehvs) **15**

[**Other Requirements**](#_2uf0to9inqzk) **15**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Everett, Dave, Jake | 10/01/17 | Draft, Initial Version | 1.0 |
| Everett, Dave, Jake | 10/12/17 | Updated w/class and Dr. Cenak comments | 1.1 |

# Introduction

## Purpose

The purpose of the this project is to research and develop a device to give to a trained dog during a disaster scenario to assist with survivor rescue and recovery. The device will send back GPS coordinates of the dog’s location, accept input from the dog as a bite or a pull to capacitive devices and interpret sound from a microphone to detect human voices. The end user will be able to use a web browser that is Google Maps capable to view pins that denote the dog’s path, sound events and contact events.

Proposed hardware for the wearable device for the trained dog is a Raspberry Pi 3 using Python code. The proposed solution will be scalable both for additional rescue dogs and additional detectors for the wearable devices.

## Document Conventions

None as as of this revision of the the Spec and Recs.

## Intended Audience and Reading Suggestions

Sponsor: Sponsor would be able to bring up a website with the pins and be able to navigate the path of the dog.  
Capstone Group: We will need to be able to simulate a dog searching around and looking for survivors/people in need of rescue. We will need to be able to simulate sound events as well as the dry contact events. The dog will be trained to search for humans and bite on a device indicating someone was found. While out searching the dog will not need a handler.

## References

Code base and documentation is located online

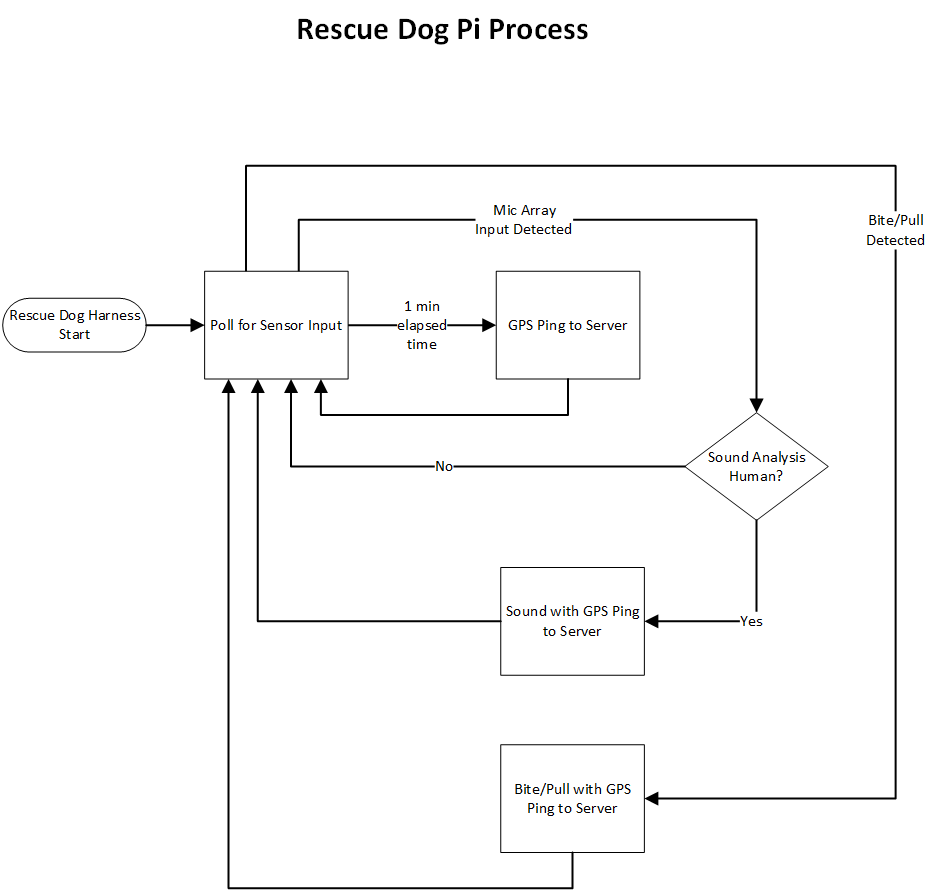
https://github.com/djschlerf/Capstone\_RescueDogSensorProject.git

# Overall Description

## Product Perspective Figure 1. Overall project overview

## Product Functions

* Raspberry Pi on Trained Dog with Devices (sensor rig)
* Server with Data Gathering (mySQL DB and Java server)
* Client Browser for Data Viewing (Chrome browser)

****Figure 2. Process overview diagram. The Raspberry Pi will be mounted to a rescue dog, and will act as a portable microcontroller to handle to processing of each peripheral and transmit results back to the server on a time basis. This feature is a high priority, as it will serve as the main communication to the field.

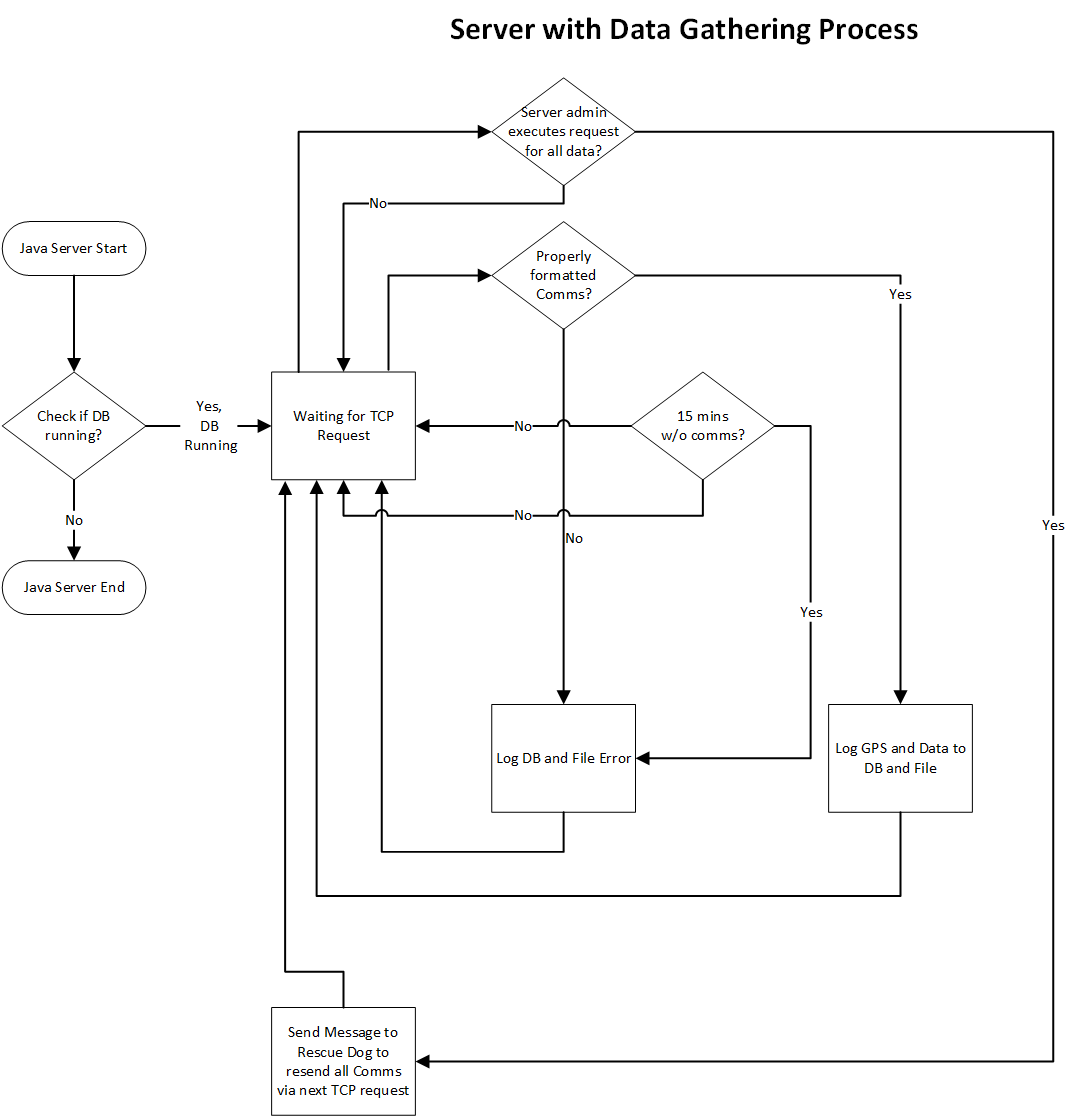
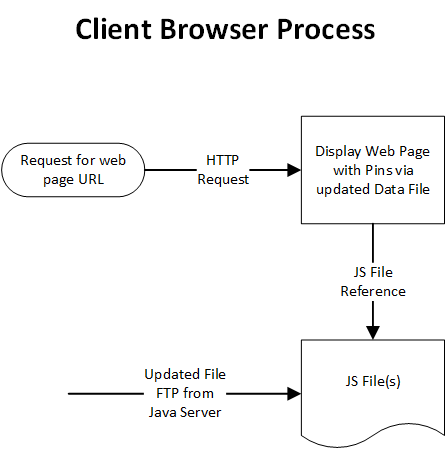
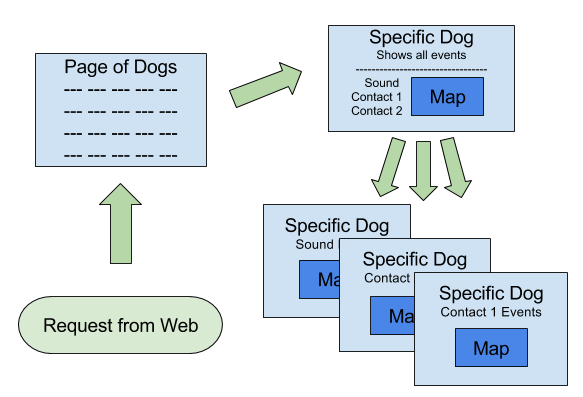
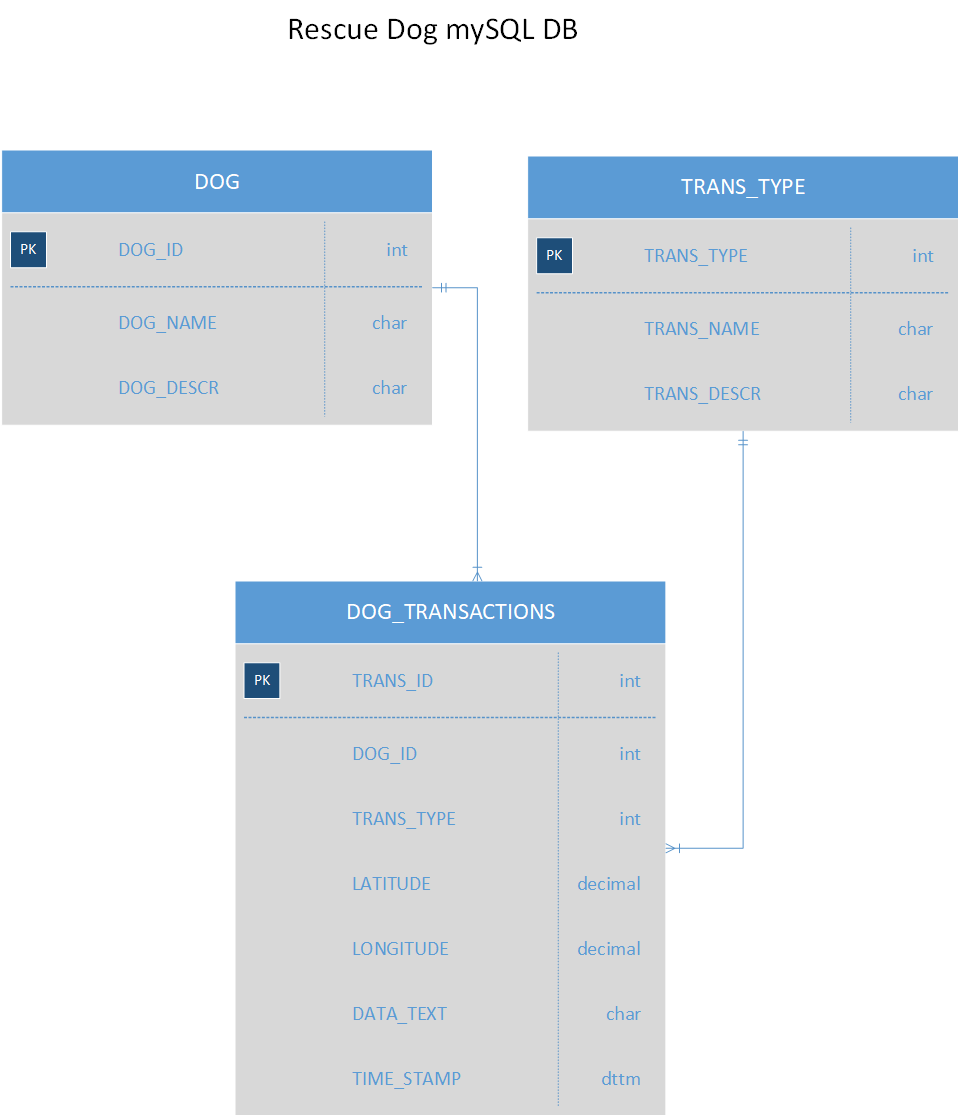
****

Figure 3. Overall Server Process. The server will written in Java and will sit in a waiting state until receiving data input from the Pi. The server will expect to receive a TCP connection request and parse acceptable input and store information in a local mySQL database on the server. This is of secondary priority since data will continue to be gathered on the Pi even if the server is down or if communications is down.

****Figure 4. Back end Process for Web Interface

****Figure 4. Client View of Web Interface with Google Map component for viewing the locations  
the dog has traveled along with the flags that denote if it found a person, or heard an event   
that can be classified as a human calling for help.

****Figure 5. Database layout and design

## User Classes and Characteristics

**Web Client Read Only User**

* Must support multiple users, scalable to 1000 simultaneous user
* A simple user interface, no training required, simple map GUI
* No subset of security access, all functions available to all users

**Server Admin**

* Infrequent, only required for troubleshooting and starting and stopping server
* Run java server, review log, stop and start process
* Run mySQL server, review logs, stop and start
* Admin access for server maintenance

**Raspberry Pi Admin**

* Infrequent hardware troubleshooting, start/stop of Pi
* Review Pi logs
* No security on the Pi, in deployment LCD will be removed effectively limiting access only to a technician with hardware to connect.

## Operating Environment

Unit on the dog:  
 Will be running a Raspberry Pi 3, with contact sensors, microphone and GPS   
 sensor

OS: Raspbian ver 07.09.2017 or better  
 Language for sensor code: Python  
Database Server:

Will be a java based listening program to listen for contact from the dog, and   
 then storing the transmission data into a MySQL (ver 5.7.19, 64bit) database running on Windows 10 Pro, build 14.393.1770 ver 1607

Web Interface:

HTML5 interface with Google Chrome ver 61.0.3163.100 or better

## Design and Implementation Constraints

A main hardware constraint will be the battery life of the dog-side system, in conjunction with the attempt at decreasing the transmission size due to the possible use of radio frequency protocol in the future. Therefore, the processing will mostly need to be done in the Raspberry Pi in the field. This will reduce the longevity of the battery.

The voice recognition algorithm will be very difficult to implement in any useful way, as the humans will not always be in need of rescue. This will require some knowledge of the area by the user in order to distinguish if the humans detected on the map are actually in distress. Implementation will also require research into an area the developers are not familiar with, and will require robust voice analysis code to be useful.

## User Documentation

GitHub - https://github.com/djschlerf/CSCE470\_Capstone\_GER\_Rec.git

## Assumptions and Dependencies

We are assuming that WiFi (in the scope of the is project) is not interrupted, since available data transmission is required.

# External Interface Requirements

## User Interfaces

## Raspberry PI ver 07.09.2017 - with Python 3: Terminal output for the Technician to read if the device booted and is talking successfully with the server

## Database Server MySQL ver 5.7.19: Terminal based commands to see and query data

## Receiving Server IIS 7 2012: Terminal based commands to see and query data

## Web Interface:Google Maps interface with pins for sound, contacts and path of the dog using Google Chrome ver 61.0.3163.100

## Hardware Interfaces

## Raspberry Pi 3 model-b https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

## Microphone array https://www.adafruit.com/product/1566

## Dry contacts (2) https://www.grainger.com/product/10C566

## GPS device https://www.adafruit.com/product/746

## LCD Screen https://learn.adafruit.com/adafruit-pitft-3-dot-5-touch-screen-for-raspberry-pi

## USB Battery Pack <https://www.adafruit.com/product/1566>

## Software Interfaces

* + 1. Terminal interface for Server
    2. Terminal interface for Database
    3. Google Chrome or Chromium browser for the map display

## Communications Interfaces

* + 1. Connection type: Wifi 80211n
    2. TCP/IP to talk to server

# System Features

* 1. **Raspberry Pi on Trained Dog with Devices**

4.1.1 Description and Priority

The Raspberry Pi will be mounted to a rescue dog, and will act as a portable microcontroller to handle to processing of each peripheral and transmit results back to the server on a time basis. This feature is a high priority, as it will serve as the main communication to the field.

4.1.2 Stimulus/Response Sequences

Status String: Will include Dog ID, Timestamp, Voice Recognition status, Input 1 status, Input 2 status, and GPS Coordinates.

Input 1 and 2: Raspberry Pi will transmit status string upon activation of dry contact inputs 1 or 2.

Voice Recognition: Raspberry Pi will use audio analysis to detect human voices via the use of a microphone array. Upon detection of a human voice, Raspberry Pi will transmit Status String as defined above.

User Defined Upload Frequency: The Raspberry Pi will transmit status string using 802.11 TCP/IP on a user defined time basis, with a default 1 minute interval.

4.1.3 Functional Requirements

req-x-1: Raspberry Pi 3B Running Linux operating System with latest Python package.

req-x-2: 2 ea Dry Contact Inputs.

req-x-3: Mic Array

req-x-4: GPS shield w/ antenna.

* 1. **Server with Data Gathering**
     1. Description and Priority  
        The server will written in Java and will sit in a waiting state until receiving data input from the Pi. The server will expect to receive a TCP connection request and parse acceptable input and store information in a local mySQL database on the server. This is of secondary priority since data will continue to be gathered on the Pi even if the server is down or if communications is down. See Figure 3.
     2. Stimulus/Response Sequences  
         On startup, server will check to see if mySQL database (figure 5) is running, exit server   
         process on error and log to file.  
         If properly formatted text string, store in DB and log to file.  
         If no comms after 15 minutes, store as error pin last known location and error   
         to log file.  
         If comms received but improperly formatted, store as error to log file.  
         If terminal input at server request resend of all comms traffic, have server   
         attach a request at next TCP connection from Pi.
     3. Functional Requirements **req-y-1:** TCP port listen on 6789 for properly formatted communication from Sensor   
         Dog Pi  
        **req-y-2:** MySQL DB for storing input messages Log messages to text log file  
         **req-y-3:** MySQL database storage of incoming sensor inputs  
         **req-y-4:** Logging of incoming data stored and messages locally   
         stored for troubleshooting  
         **req-y-5**: Error Handling: Every 15 minutes log an error  
         **req-y-6**: Export of sensor data every 15 minutes (user   
         defined)
  2. **Client Browser for Data Viewing**
     1. Description  
        Web interface with a Google Map component for viewing the   
        locations the dog has traveled along with the flags that   
        denote if it found a person, or heard an event that  
        can be classified as a human calling for help. See Figure 4.
     2. Stimulus / Response Sequences  
        User will use a Google Chrome to view the data gathered by the dog, that is displayed on Google Maps.
     3. Functional Requirements   
        **req-z-1:** Google Chrome or Chromium browser to display   
        the data collected on a Google map interface  
        **req-z-2:** Computer Mouse that can interact with the map   
        and data points  
        **req-z-2**: Selection boxes to filter data displayed

# Other Nonfunctional Requirements

## Performance Requirements

* + 1. Transmit location
    2. Identify voice
    3. Input 1 and 2 via dry contacts
    4. Update map

## Safety Requirements

Currently none at this time

## Security Requirements

## For Raspberry Pi: Currently none at this time

## For Servers:Admin level access required at this time

## For Web interface:Default TCP/IP security

# Other Requirements

*Currently none at this time*

**Appendix A: Glossary**

*Currently none at this time*

**Appendix B: Analysis Models**

*Currently none at this time*

**Appendix C: To Be Determined List**

*Currently none at this time*