Deliverable J - User Manual

Volunteer Call Bell System

For

GNG 2101A

Professor H. Anis



By Group A3

Group Members:

Samuel McPhail 8587665

Douglas Wise 300026629

Amayas Said Amer 300084339

RunQiu Li 300082280

# Abstract

The presented system involves a central API and web server that interact with a database to facilitate the notification of hospital volunteers to patients’ requests. When a patient POSTs a request containing a room number and description, the web server sends a JSON formatted package to the API, where this server adds a timestamp and inputs the request into the database. When a volunteer accesses the web app, the server returns the database’s contents through the GET method, ordered from oldest to most recent. Volunteers can then delete a request when it has been tended to. The web app can be accessed through any personal wireless device. Additionally, a peripheral device, a button, can be used to access the service if a personal wireless device cannot be used by the individual.

Table of Contents

[Abstract 2](#_Toc26386941)

[List of Tables 4](#_Toc26386942)

[List of Figure 5](#_Toc26386943)

[Introduction 6](#_Toc26386944)

[System Layout and Features 7](#_Toc26386945)

[Overview 7](#_Toc26386946)

[Features 7](#_Toc26386947)

[Hardware 7](#_Toc26386948)

[Server 7](#_Toc26386949)

[Button 7](#_Toc26386950)

[Software 8](#_Toc26386951)

[Server 8](#_Toc26386952)

[Button 8](#_Toc26386953)

[Operation 8](#_Toc26386954)

[Server 8](#_Toc26386955)

[Button 9](#_Toc26386956)

[Health and Safety 9](#_Toc26386957)

[Conclusions and Recommendations 10](#_Toc26386958)

[Appendix A – Login Information 11](#_Toc26386959)

[Appendix B – Images 12](#_Toc26386960)

# List of Tables

[Table 1 - Server Passwords 11](#_Toc26383781)

[Table 2 - Web Application Passwords 11](#_Toc26383782)

# List of Figures

[Figure 1 -Node-RED flow for HTTP POST Method 13](#_Toc26386961)

[Figure 2 - Wiring Diagram for Raspberry Pi Zero W and Button 13](https://uottawa-my.sharepoint.com/personal/smcph070_uottawa_ca/Documents/GNG2101%20Presentation/Deliverable%20J.docx#_Toc26386962)

[Figure 3 – Pi zero button wiring 14](#_Toc26386963)

[Figure 4 - Raspberry Pi with fan wired to pin 4 and 6 14](https://uottawa-my.sharepoint.com/personal/smcph070_uottawa_ca/Documents/GNG2101%20Presentation/Deliverable%20J.docx#_Toc26386964)

[Figure 5 - Volunteer Web Application 15](https://uottawa-my.sharepoint.com/personal/smcph070_uottawa_ca/Documents/GNG2101%20Presentation/Deliverable%20J.docx#_Toc26386965)

[Figure 6 - Patient Web Application 15](https://uottawa-my.sharepoint.com/personal/smcph070_uottawa_ca/Documents/GNG2101%20Presentation/Deliverable%20J.docx#_Toc26386966)

# Introduction

The product’s inception stemmed from Saint Vincent Hospital’s (SVH) needs for a method to automate the delegation of non-nursing tasks to volunteers. Currently, the volunteers rely on the direction of the nursing staff to provide direction as to where they can provide assistance. Volunteer’s abilities in the hospital are limited to reduce liability; they cannot touch patients, provide medical assistance or handle sensitive information. When the nursing staff is directing volunteers, they are not efficiently utilizing their time to provide medical assistance to patients.

SVH is looking to automate the delegation process through a perspective call bell system that would allow nursing staff to remain on task and allow volunteers to operate with more autonomy. The overall gain of such a system would be to improve the efficiency of the hospital. What was achieved was a functioning system that will allow patients to submit requests for assistance to the volunteers through the use of a button or a simple web application. These requests are then viewed by the volunteer, carried out, and deleted from the database. The essence of the system is unseen to the user; the server that interacts with the web apps and the database servers at a foundation for the system, processing posted requests and storing them in the database. Due to the simplicity of the system, no further changes will be needed to be made to the server, allowing easy implementation in a hospital.

On the market, few to no options exist that provide hospitals or nursing homes the ability to submit non-urgent assistance requests to staff and volunteers. Additionally, no other design developed as a part of GNG2101 project have been able to deliver a working product that fulfils the needs of the client until now. The system produced is streamlined to ensure usability, it is adaptable to the setting, and it is expandable to allow more points of entry to the system that could be made specifically for patients (through personal computer interfaces or a virtual assistant such as Alexa).

# System Layout and Features

## Overview

The volunteer call bell system is centred around an API server and database. Additionally, a web app is employed to provide the server data. For non-browser submissions to the server/database, IBM’s Node-RED service sends data to the server from a peripheral button device.

## Features

The volunteer call bell system features two web applications, one for use by volunteers, seen in Appendix B Figure 5, and patients, seen in Appendix B Figure 6. These separate web applications interact with the API server. The web app allows easy interaction with the server and database by the volunteer to manage and view requests.

## Hardware

### Server

The server is hosted on a Raspberry Pi 3 A+ computer. It is housed in a 3D printed case with a 5V fan mounted inside connected to pins 4 and 6. The board is to be plugged into a wall outlet and the start-up will occur. See figure 4 for wiring diagram.

### Button

The button is made of a Raspberry Pi Zero W computer. It is housed in a 3D printed case with 16mm pushbutton mounted to its lid. Additionally, the button is wired via male jumper wires to GPIO pins 37 and 39 (GND). See figures 2 and 3 for wiring diagram.

## Software

### Server

The Raspberry Pi 3 A+ board is fully operational with the Raspbian Buster operating system. Node-Red and RaspAP are both installed on the computer which provide IoT services and a a wireless network manager. Additionally, VNC Viewer is activated on the Raspberry Pi to allow headless operation of the device through a wireless network. Server and wireless information are located in Appendex A, table 1.

On the desktop in folder *server\_p3*, hold all files to initiate and maintain the server. For further expansion of the system and the initialisation of other servers, the files can be found at the following URL:

<https://github.com/ASA11599/SVH-CallBell/tree/master/final>

### Button

The button utilizes Node-RED to facilitate the HTTP POST request submitted via the wireless network to the server. The Node-RED flow connects a GPIO in Node from pin 37 to a switch node that, when activated, will send the HTTP request to the server IP.

## Operation

### Server

To initiate the server, the following operations are followed:

1. Plug-in the server board to a power supply and outlet; on a wireless computer, connect to ServerPi network (see Appendix A for password)
2. Open two terminal windows, type command *ssh* [*pi@192.168.50.1*](mailto:pi@192.168.50.1), when prompted, enter password *serverpi* in each
3. In the first terminal, type command *cd Desktop/server\_p3/api,*  followed by node api
4. The line *API server now listening at 192.168.50.1 on port 8080* will be shown
5. In the second terminal, type command *cd Desktop/server\_p3/www* followed by node web
6. The line *server now listening at 192.168.50.1 on port 3000* will be shown
7. The web server and api server are now listening and can be accessed through the URL [*http://192.168.50.1:3000*](http://192.168.50.1:3000)
   1. See images in Appendix B

### Button

The button has been preset through Node-RED to initiate upon start-up. The operations to set up are as follows:

* Ensure Web and API servers are functioning by following the server setup procedure above.
* Plug in button board and allow for connection to wireless network to occur
* The button is now ready to submit requests to the volunteer page of the web app.

To change the room number and description the button sends to the api server, the URL submitted in Node-RED must be changed. It can be accessed through a web browser at [*http://192.168.50.137:1880*](http://192.168.50.137:1880). In the HTTP Request Node the form of the URL passed to the API server is:

http://192.169.50.1:8080/api?room=*ROOM*&description=*MESSAGE*

where:

*ROOM* is the room number in which the button is to be located

*MESSAGE* is the description that accompanies the room number

NOTE: the message syntax sees that spaces are replaced with “%20”.

Example: “I need help” becomes “I%20need%20help”

## Usage of System

### Patient

The patent is to use the patient web app by accessing it though the URL: <http://192.168.50.1:3000>. Once there, the patient clicks on the patient button, and enters the app using the password *password.* From there, the patient fills the first field with their room number and the second field with a description. The patient app can be seen in Appendix B, Figure 6.

### Volunteer

The volunteer is to use the volunteer web app by accessing it though the URL: <http://192.168.50.1:3000>. Once there, the volunteer clicks on the volunteer button, and enters the app using the password *password.* From there, the volunteer has access to all requests that have been made to the server. The auto-refresh function will allow the list of submitted requests to be kept up to date, with new requests added to the bottom of the list. When a request is carried out and completed, the volunteer will use the delete button beside that request to remove it from the list.

# Health and Safety

The health and safety risks that are associated with use of the system stem from the confusion between using the volunteer call bell system and the emergency nurse call bell system. Patients that are not able to differentiate between the two systems should not access the volunteer system via button to minimize the risk associated with a patient using the non-urgent volunteer system in an emergency situation. Additionally, risk is minimized for those using personal electronic devices (phones, laptops, tablets, etc.) as accessing the system in its current form requires the user to connect to a separate network, reducing the risk of false alarms.

# Conclusions and Recommendations

The volunteer call bell system was successfully developed for the client and can now be implemented in the hospital setting where it is intended to be used. This system, more specifically the server and database, will serve as the basis for future expansion. Using a combination of a Raspberry Pi 3A+ to host the server, a patient and volunteer specific web application, web and API servers and a database, this system is completely functional and can be easily implemented in a hospital. All system functions have been successfully created and tested thoroughly by the development team and can further be tested and implemented in a hospital environment.

To further develop the volunteer call bell system, the web app, button hardware and wireless network are areas which can benefit from further development. The web app, although completely functional, is not stylized or personalized for the client, SVH. This is a minor and easily made change and if further project work is continued will be implemented shortly. The button hardware can be reduced and made more cost effectively with minimal development. Early on in the design process, the decision was made to utilize a Raspberry Pi Zero W computer board in order to provide multiple levels of redundancy if WiFi could not be used to transfer data as it possesses Bluetooth transmission/receiving capabilities. Knowing that WiFi can be used, a printed circuit board would be used to consolidate components into a single board and decrease production time.

# Appendix A – Login Information

Table 1 - Server Passwords

|  |  |
| --- | --- |
| Credential | Value |
| Server Hostname | ServerPi |
| Server Password | serverpi |
| RaspAP username | Admin |
| RaspAP password | secret |
| VNC Username | Pi |
| VNC password | serverpi |
| Node-RED Port | 1880 |
| Wireless Network Name | ServerPi |
| Wireless Network Password | serverpi |

Table 2 - Web Application Passwords

|  |  |
| --- | --- |
| Web Application | Password |
| Volunteer | Password |
| Patient | password |

# Appendix B – Images

A screenshot of a computer

Description automatically generated

Figure 1 -Node-RED flow for HTTP POST Method

A screenshot of a cell phone

Description automatically generated

Figure - Wiring Diagram for Raspberry Pi Zero W and Button

A close up of a device

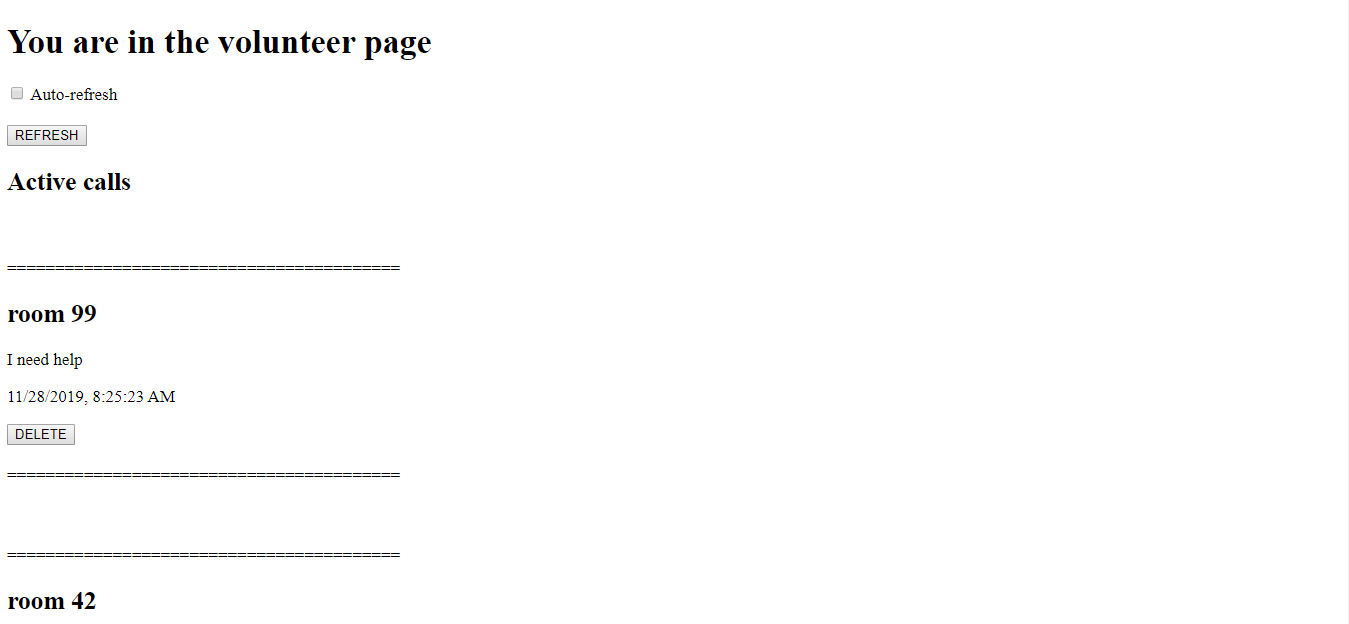
Description automatically generated

Figure 3 – Pi zero button wiring

A picture containing object

Description automatically generated

Figure - Raspberry Pi with fan wired to pin 4 and 6

A screenshot of a computer

Description automatically generated

Figure - Volunteer Web Application

Figure - Patient Web Application