



**Helpful Monitoring System for the  
Nursing Home Proposal**

# Table of Contents

I.	Concept of Operations.....	Page 3
II.	System Overview.....	Page 4
	A. Description.....	Page 5
	B. System Diagram.....	Page 8
III.	Testable Requirements.....	Page 9
IV.	Hardware list.....	Page 10

# Concept of Operations

## What is our system doing?

Our system will combine facial recognition, data logging, the Internet of Things, and portability into a single application. As a resident is staying in a nursing home, our system will detect visitors based on facial recognition and from there, perform different functions based on whether or not the visitor is on a “trusted” list. If the visitor is trusted (such as family or nurse) then information about that person will be displayed for the resident in the case the resident suffers from memory loss and can no longer remember details. If the visitor is not on the trusted list, a snapshot will be taken of that person and be sent to an emergency contact (such as the nursing home’s security) via text message.

Additionally, our application will store three types of databases/logs:

- 1) The profile of each trusted person stored as a JSON file containing fields such as name and relation to the resident.
- 2) A history log of visitors displaying the name of each visitor and time visited.
- 3) A data log that nurses can input what they did at each check-up such as medicine given and how much at what time. This is so that residents, family, or other nurses can keep track of what was done at the last check-up.

## What is the goal of our system?

For people that live in a nursing home, visitors can come by during walk-in hours at any time and anyone can freely enter your room. Today, a nurse, family member, or unexpected visitor can stop by and leave with no record of being there besides a sign-in sheet at the nursing home’s front desk. Our system aims to provide comfort and reassurance to the family of a nursing home’s resident through real time facial recognition of visitors. The resident will be able to view information about previously approved visitors stored in a database. In addition, there will be a log about what happens at specific times during a nurse check-up that relatives of the resident or other nurses can check. When an unauthorized or unknown visitor enters the resident’s room, the camera will not recognize them and a MMS message will be sent out to family members and to approved facility employees like a security guard.

## Why is our goal meaningful and important?

According to the National Care Planning Council, about 88% of the US nursing home residents are over the age of 65. About half of all residents were admitted from a health care facility. They are in care because of physical needs and often suffer from a wide arrange of mental disorders. Most of the them will never recover and be able to

take care of themselves in the future. The concept of family and the supports from their loved ones are the most valuable things to them. Our design is not only a logging system. It is a bridge that connects the residents and visitors, and aims to help them to remember every visit.

#### How will a user interact with our system?

A user can interact with the system in two different interfaces. The main interface will be through the exchange pi. This is where a user (such as resident, nurse, or family) can access the various history logs available to them, such as a senior resident viewing who visited them recently. The other point of interaction will be through the pi running OpenCV. This will be where a user can input someone new into the list of trusted people for the resident.

#### How will our system achieve the stated goals?

Social isolation has been associated with cognitive decline and depression. So maintaining visiting a senior resident will bring back positive memories and every visit will mean a lot to the resident. These memories can prompt nostalgia and are beneficial to those with dementia and memory loss. Therefore, our application will store each visit and allow the resident to review them, bringing back happy times to our senior while providing a sense of security.

# System Overview

## What are the major pieces?

\_\_\_\_\_ Our application makes use of three major pieces, each one being run on a separate raspberry pi.

1. Log repository
2. Exchange pi
3. The Eye (Raspberry pi running OpenCV)

## What is each piece supposed to do?

\_\_\_\_\_ The Log Repository will house each of the three databases and will perform the main functions of the application. There will be no user interaction with this piece, however; this is where the main transition of information occurs.

The Exchange pi will be where the primary user interaction occurs. Here, the user will be able to query each database and the requested information will be displayed on screen.

The Eye will be running OpenCV with a pi camera to perform facial recognition on each visitor in the room. After determining whether or not that visitor is authorized/trusted, it will send information to have that visitors profile displayed for the resident or send a text message out to security declaring that an unauthorized visitor is in the room.

## How will each piece accomplish its given task?

\_\_\_\_\_ The log repository pi will host a MongoDB database with three different collections (history of visitors, log of events, profiles of authorized people). The fields for each collection are as follows:

1. History of visitors
  - 1.1. Name
  - 1.2. Time of visitation
2. Log of events
  - 2.1. Name
  - 2.2. Description of the event (i.e Gave medicine, family visited, etc.)
  - 2.3. Date and Time
3. Profile of authorized visitors
  - 3.1. Name
  - 3.2. Related picture

When the log pi receives a new message via JSON format from either the exchange pi or the Eye pi; based on the command sent with the JSON file, the

repository will store it in the proper collection within MongoDB. If the message from the Eye pi specifies the visitor is authorized, then the log pi will display on a monitor the profile contents of that visitor (see figure 1). In the event an unauthorized person enters the room, the message sent to the log pi will specify as such and it will send that taken photo to a facility security officer via text message.



Figure 1: Example of profile displayed

The exchange pi will take input via a keyboard. Commands given will be flagged by -U, -H, -V, -A; each command will result in:

1. U: prompt the user for name and a description of the event
2. H: the events log being displayed for the user to view
3. V: the history log of visitors being displayed
4. A: displays the list of approved individuals

Information sent to and received from the log pi will be communicated via sockets.

The Eye pi will use the given pi cam and run OpenCV. This Rpi will use facial detection to tell when a visitor has entered the room. Upon detection, it will decide if that visitor is authorized or not and send that taken image and a boolean value to the log pi, again as a JSON format. The Eye pi also has a feature to add a new person to the list of approved individuals. The process to do this will be a series of photos and information input via keyboard to update that new profile of that individual.

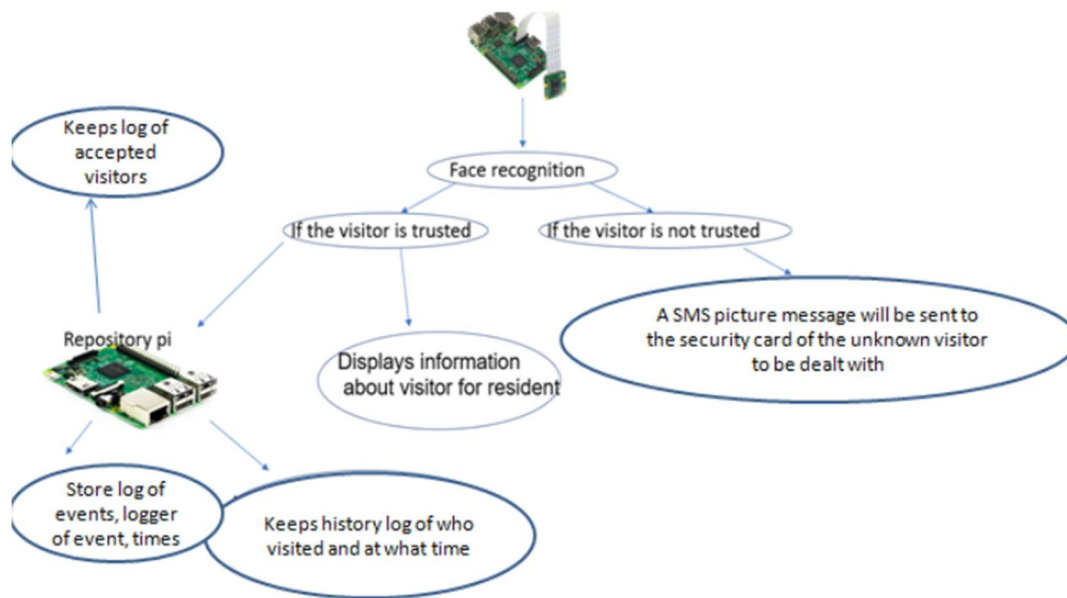


Figure 2: System flowchart

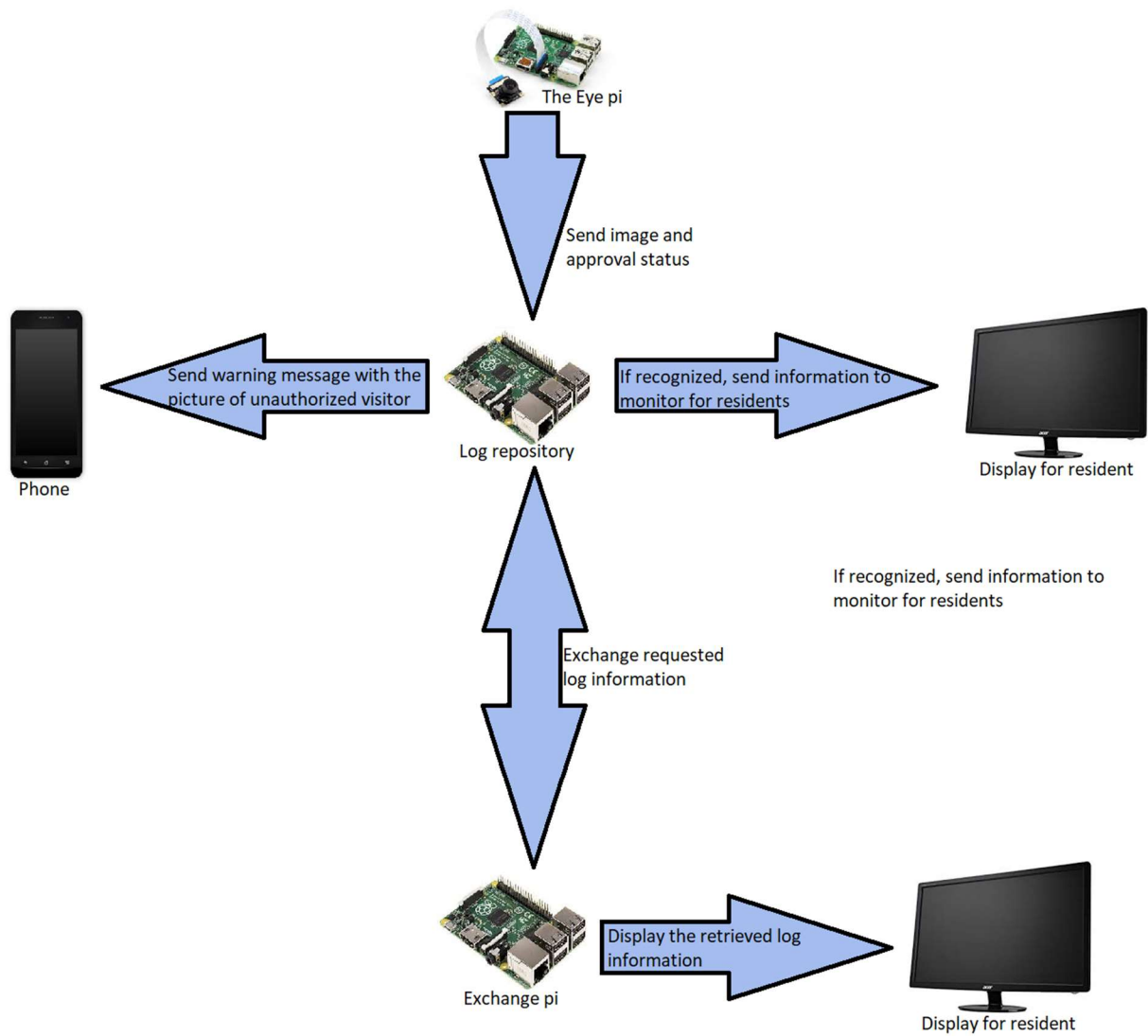


Figure 3: System Diagram



## Testable Requirements

1. The Eye Rpi must be able to recognize faces using OpenCv **-done**
2. The Eye Rpi must be able to distinguish between distinct faces using OpenCV (i.e. cannot handle twins)- **send image to log pi and the compare will be done in the exchange pi. - done**
3. The Eye Rpi must be able to run in a mode to add new acceptable users with an “-a” flag. **-under -a flag, the eye pi will send the image to approved database under log pi. (delete)**
4. The Eye Rpi must be able to run in a mode to begin recognizing visitors with an “-r” flag. **-under -r flag, send the image to log pi to compare -done**
5. The Eye Rpi must send an image of the visitor through a socket connection to the Log Rpi **-Log Pi stored the approved personals’ images -done**
6. The Eye Rpi must send a message through a socket to the Log Rpi that indicates whether a person was either approved or not approved along with their name in a JSON format (if person is recognized) **-done**
7. The message received by the Log Rpi will be stored into a MongoDB collection called Visitors and include the person’s name (if approved visitor) and the visiting time - **done**
8. The Log Rpi must display an image of any approved visitor on the monitor it is connected to **-The image will be edit to add texts: name and relationship to the residence-Done**
9. The Log Rpi must also display an approved person’s name and relation to the resident **-This will be done in 8**
10. In the event that there was an unapproved visitor an SMS must be sent using the Twilio API by the Log Rpi, with an image of the unknown visitor attached(Figure 1), to the facility security guard and indicate that there is currently an unknown visitor with the resident in the SMS message **-Done**
11. The Exchange Rpi will take 1 of 4 commands,
  - ‘U’ indicates the user would like to update a log of past events
  - ‘H’ indicates the user would like to see the log of past events stored with the ‘U’ command
  - ‘V’ indicates the user would like to see the history of visitors
  - ‘A’ indicates the user would like to view a list of ap that can visit
12. The Exchange RPI must be able to take keyboard input for a nurse to log events when they enter the ‘U’ command when prompted and send messages through a socket connection to the Log RPI to show what has been done at certain times and by who
13. The logged event must create a message in JSON format and contain the name, time, and description of the event and send that through a socket connection to the Log Rpi

- The received message on the Log Rpi will be stored into a MongoDB collection called History

**14. The Exchange RPI when given the command 'H' must display the last 10 events that were stored into the log with the 'U' command or all events if less than 10 events were stored**

15. The Exchange RPI when given the command 'V' must display the last 10 visitors that saw the resident with what their name is and visit time or all visitor informations if less than 10 visitor informations were stored

16. The Exchange RPI when given the command 'A' must display all the approved visitor's names, relationship to the resident, and their associated photographs

## Hardware List

1. Raspberry Pi 3 (x3)
  - a. Given, \$0
2. Monitor (x2)
  - a. Can we use the ones in the SWEL?
3. HDMI cable (x2)
  - a. Can we use the ones in the SWEL?
4. Raspberry pi camera Camera Board V2.1, 8MP, element14 (x1)
  - a. Given, \$0
5. Ribbon Cable (to connect to Pi's CSI connector)
6. A mobile phone (Android/IOS) (x1)
  - a. Own already, \$0