University of the Philippines Manila
College of Arts and Sciences
Department of Physical Sciences and Mathematics

Fire Detection System using CCTV

A special problem in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Computer Science

Submitted by:

Christine Joyce O. Arzadon

June 2016

Permission is given for the following people to have access to this SP:

Available to the general public	Yes
Available only after consultation with author/SP adviser	No
Available only to those bound by confidentiality agreement	No

ACCEPTANCE SHEET

The Special Problem entitled “Fire Detection System using CCTV” prepared and submitted by Christine Joyce O. Arzadon in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

Marvin John C. Ignacio M.Sc.(*candidate*)
Adviser

EXAMINERS:

	Approved	**Disapproved**
1. Gregorio B. Baes, Ph.D. (*candidate*)		
2. Avegail D. Carpio, M.Sc.		
3. Perlita E. Gasmen, M.Sc. (*candidate*)		
4. Ma. Sheila A. Magboo, M.Sc.		
5. Vincent Peter C. Magboo, M.D., M.Sc.		
6. Richard Bryann L. Chua, M.Sc.		

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

Ma. Sheila A. Magboo, M.Sc.
Unit Head
Mathematical and Computing Sciences Unit
Department of Physical Sciences
and Mathematics

Marcelina B. Lirazan, Ph.D.
Chair
Department of Physical Sciences
and Mathematics

Leonardo R. Estacio Jr., Ph.D.
Dean
College of Arts and Sciences

Abstract

With the fast advancement of technology in the area of the CCTV's, different applications for the said technology is being developed right now. Aside from the fact that there is a law that dictates that all businesses in the Philippines are required to install and maintain CCTV's if they fall on at least one of three categories, people use CCTV's to monitor their properties even remotely. One relatively new application for the CCTV is fire detection. Current fire detectors are prone to errors because smoke detectors can be set off even with the smoke from a cigarette and other types of smoke that isn't produced by fire. This system makes use of computer vision in analyzing the video inputs from the CCTV camera.

Keywords: fire detection, computer vision, CCTV, OpenCV

Contents

List of Figures

List of Tables

I. Introduction

A. Background of the Study

CCTV (closed-circuit television) is a TV system in which signals are transmitted to a specific place and are monitored, primarily for surveillance and security purposes. Due to the fact that cameras are connected to the monitors and/or video recorders using private coaxial cables or wireless communication links, it became known as "closed-circuit" to indicate that access to the contents is constrained by design to those who can see it [1]. The most common types of CCTV's are: fixed CCTV camera, PTZ (Pan Tilt Zoom) CCTV camera and infrared/night vision camera [2, 3, 4].

House Bill 6672 also known as the "CCTV Act of 2012" dictates that all businesses in the Philippines are required to install and maintain CCTVs if they fall on at least one of the three following categories: (1) "business establishments employing more than 20 employees/workers"; (2) "work premises of not less than 50 meters"; and (3) "whose transactions amount to not less than PhP 50,000.00 each day" [5, 6]. However, the lawmakers didn't state anything regarding the type of CCTV camera the businesses should use that means they are allowed to use even moving cameras so long as they follow the rules stated in the law. This system will be applied to the most common type of CCTV camera which is the non-moving camera.

A human can determine if what he is seeing is a fire by using his senses. Meanwhile, a computer can also detect fire by using the fire color model proposed by Celik et al [7]. According to the fire color model, each fire pixel has its Red channel value higher than that of the Green channel and its Green channel value is higher than the Blue channel value. In addition to that, there is high saturation in the Red channel of the fire pixels.

Current fire detectors like smoke detectors and heat detectors create a lot of false positive alarms. Smoke detectors make an alarm when smoke particles "reflect light inside photoelectric smoke detectors". However, dust particles can also do this. Water steam from a hot bath can also be detected by the smoke detector because "water vapor mimics the effect of smoke" [8]. Thus when these normal circumstances happen, the smoke detector sets off even without actual fire. Meanwhile, heat detectors make an alarm when a specified temperature has been reached and/or when the rate of increase in temperature has surpassed a predetermined rate. The false positive alarms occur when these heat detectors are put in the wrong place especially on places where they are exposed to extreme conditions [9]. One specific example of a heat detector is in the form off pressurized plastic tubes. They were placed near process vessels so that if there's any fire close to the vessels the tubes will burst and the loss of pressure indicates that there is fire [10]. The biggest problem with this pressurized plastic tubes is the fact that they are one-time-use heat detectors. The tubes will have to be replaced every time they are set off. These detectors use either loud sound or flashing light in order to inform people that there is fire within the vicinity.

Existing advanced fire detection systems include SigniFire, Computer Vision Based Fire Detection Software and GKB's Video Fire Detection System. SigniFire is a camera-based flame and smoke detection system where the SigniFireIP camera has built-in flame, smoke and motion detectors. This system can also be used with existing security equipments using the SigniFire Server since it is Wintel based and uses the same algorithm as the SigniFireIP camera which also makes it capable of detecting and alarming on events [11]. The Computer Vision Based Fire Detection Software is basically what its name says; it analyzes video/image data acquired from an ordinary surveillance camera and a frame grabber [12]. It only detects the presence of fire within the view of the camera used. GKB's Video Fire Detection System has two types, the first one is the system that accepts video input from analog cameras only and the second one is the system that accepts video input from IP cameras only. Aside from detecting fire within the view of the camera, it uses SMS in notifying users of the presence of the fire [13].

B. Statement of the Problem

Current fire detectors are prone to errors because smoke detectors can be set off even with the smoke from a cigarette and other types of smoke that isn't produced by fire [10].

C. Objectives of the Study

This study aims to develop a fire detection system using CCTV. First, the system will detect the "fire-colored" area on the screen, if there is any, by inspecting its YCbCr color space and RGB color space and conforming with the fire color model. In YCbCr color space, Y refers to the brightness/luminance of the image, Cb refers to the blue-difference chrominance and Cr refers to the red-difference chrominance while in RGB color space, R refers to the Red channel, G refers to the Green channel and B refers to the Blue channel of the image. Then it will be analyzed from frame to frame because fire is said to be a moving object [7]. When the first two steps has been passed, a sound and message alert will be created by the system in case a level alert was reached. The alert levels will be set by the building administrator according to the total area of the fire region in terms of pixels.

This system will:

1. Allow the Building Occupant to:
   (a) Be notified about the occurrence of the fire through the message alert
   (b) Login to the system with the credentials provided by the Building Administrator
   (c) Edit their own profile
2. Allow the Building Administrator to:
   (a) Add / Remove Building Occupants
   (b) Edit Fire Department info

(c) Edit their own profile

(d) Set the alert levels for the fire alarm

D. Significance of the Project

Since the system will detect the fire on the screen, the problem regarding the false positive alarm caused by other types of smoke not produced by fire is resolved.

E. Scope and Limitations

The following is a list of what the system will do:

1. The system will detect only fire, if there is any, from the video input.

2. The actual output of this system is the sound and message alert. It will only be sent whenever fire was detected.

3. Only the system administrator will be allowed to only add/remove Building Occupants. Moreover, he/she will also be able to only edit Fire Department info which is also in the database.

4. Currently, the system can port with IP cameras only.

This system has the following limitations:

1. This system will not detect smoke. It will only detect and analyze the fire flames.

2. This system will not detect bluish flames. Only those fire whose color ranges from red to yellow.

F. Assumptions

The following list contains the assumptions made for this fire detection system using CCTV:

1. The fire is within the viewing range of the camera.

2. The Building Administrator is the one responsible for contacting the firefighters.

3. The system is installed in the Building Administrator's machine.

4. The url that will link the IP camera to the system is legitimate. It is advisable to try streaming the IP camera on VLC media player first.

II. Review of Related Literature

The topic of video-based fire detection has been studied for about 15-20 years now and multiple systems has been developed abroad. These systems have similar processes but differ on their implementations. The general flow of these systems is as follows: (1) receive the video input; (2) detect motion in the video; (3) check if motion detected is fire; (4) analyze the fire and (5) make an alarm.

In the work of Habiboglu et al [14], FP7 FIRESENSE follows the general flow. However, it took a different approach in analyzing the fire. Since the goal of their system is to detect wildfires in the forest from a very far distance, they focused on analyzing the smoke generated by the fire. They used correlation descriptors to obtain the characteristics of the smoke in the detected regions.

Aside from detecting fire and analyzing it, Santana et al [15] also proposed ways to increase accuracy and reduce false positive alarms. They compared different combinations of three methods, which relies on HSI, RGB and YCbCr color-spaces. This comparison resulted to the analysis of the combination of HSI and YCbCr color-spaces to generate the best result. They proposed the use of a two-dimensional Discrete Wavelet Transforms (DWT) in verifying the textured nature of the flames to reduce the false positive alarms generated by the system. They also included another process where they detect people which runs concurrently with the other processes to further reduce the false positives.

Meanwhile, the biggest difference in the work of Cetin et al [16] with other systems is their use of covariance descriptors which enables their system to work even on moving cameras. A support vector machine was used in their classification process.

Similar to FP7 FIRESENSE, the authors of [17] also focused on the smoke produced by fire. Unlike FP7 FIRESENSE, they used Fuzzy c-means clustering in order to separate the smoke regions from other moving regions. In addition to that, they also made use of Support Vector Machines (SVM) to increase the accuracy of their system. Parameters like motion, surface roughness and area randomness were analyzed to further prove that the region detected contains smoke from fire.

Just like the other systems mentioned above, the system proposed by the authors in [18] detects smoke for early fire detection. This system works by separating the smoke region from the background with the use of the analysis of threshold sets described in the article [18]. However, the authors said that if the image being analyzed is not homogeneous, it will be beneficial to use other leveling algorithms before implementing the proposed approach.

Quite similar to the work of Santana et al [15], the authors of [19] also focused their research on the fire itself and even developed their own "model of fire flame". Unlike the work of Santana et al where they used three color spaces, the authors of [19] made use of only one color space which is the RGB color space. They analyzed the red saturation and luminance of the fire region in a series of images and they came up with a pattern present on all fire regions. This pattern shows that "the luminance of the fire is decreasing gradually from the center to edge while the red saturation of the fire is increasing" [19]. In addition to that, the circularity feature and the dynamic shape feature of the flames were included in the analysis to further recognize the fire and thus reduce the false alarms.

In [20], Wirayuda et al proposed a general approach with motion detection, fire color detection, temporal analysis and spatial analysis. The difference of their system with the other systems is their use of K-means clustering algorithm to build a color look-up table to determine whether the pixel being analyzed is fire-colored.

Azmara and Aziz [21] proposed a system to alert users, using Short Message Service, of the level of flood was proposed. The system has three level sensors connected to a Micro Controller Unit (MCU). They used liquid sensors because of its unique ability to detect liquid objects. The sensors were placed 5, 7 and 9 feet from the threshold of the main water tank where it was tested. The threshold was 4 feet deep. Once the water has reached the sensors, a SMS alert will be sent to the users with SMS codes of Alert, Warning and Danger respectively.

Some of the systems mentioned above were designed to detect smoke which will indicate the presence of fire. However, the system proposed in this paper is created to detect only the fire itself. Some of the mentioned systems used Support Vector Machines but SVM won't be used for this system because it was used to recognize the fire, to formulate a fire-color model and to increase accuracy. Since there is already an established fire-color model, there will be no need to use SVM. Moreover, other ways analysis will be used to further increase analysis to avoid the use of SVM which will need data sets and may take hours to complete.

III. Theoretical Framework

A. Fire

A..1 Fire Detection

1. Common Fire Detectors in the Philippines

   The following is a list of the common fire detection equipment:

   - Heat Detectors

     They are usually placed in spots where there is a fixed temperature. It is advisable not to install these devices on places with fluctuating temperature [22].

   - Flame Detectors

     This type has three subcategories: optical, UV and IR detectors. Optical detectors have optical sensors for detecting the flames. UV detectors are said to work very fast. They have the ability to detect UV radiation at the instant of ignition. However, a three second delay is applied to prevent accidental triggers. Meanwhile, Infrared (IR) detectors look for the head radiation brought about by open flames and fire [22].

   - Smoke Detectors

     Meanwhile, this type has two subcategories: photoelectric and ionization alarms.

     Photoelectric alarms function with the use of a light source, photoelectric sensor, and beam collimating system. Once smoke has entered the optical chamber, light in the light beam path will be scattered due to the presence of the smoke particles. The scattered light will be the one to set off the sensor and thus creating an alarm [22].

     Ionization alarms have a small amount of radioactive material that passes through the ionization chamber. The radiation permits a small current between the two electrodes inside the said chamber. By the time smoke has entered the chamber, the alpha particles will be absorbed which will result to a disruption in the current and also an ionization reduction and then the alarm is set off [22].

2. Motion Detection

   Motion Detection, also called as Activity Detection, is usually "a software-based monitoring algorithm" where it will start capturing/recording the event when motion is detected within the view of the camera [23]. Basically, motion detection using camera compares sequences of images and determining whether these differences are the effects of motion [24]. Devices like motion sensors and cameras can be used in detecting motion. They are normally used in security systems where they send signals to the security system's control panel which is connected to the owner's monitoring center, alerting them of the potential threat [25]. In the proposed system, motion is detected first to identify the possible fire regions. In addition to that, doing this will lessen the number of pixels that will be checked to further prove that this pixel is part of the fire region.

   (a) Types of Motion Sensors

   There are two main types of motion sensors: the passive sensors, where they read changes in energy of the surrounding area but do not emit energy themselves and the active sensors, where they emit one of three kinds of energy (infrared light, microwave radiation and sound waves) in order to detect motion [25].

   i. Passive Infrared (PIR)

   These type of sensors are the most widely used sensors in home security systems. They detect body heat and are set off once the infrared energy levels change rapidly [25].

   ii. MicroWave (MW)

   MicroWave sensors send out microwave pulses and computes the reflection off an object in motion. Despite the fact that they cover a larger area than PIR, they are prone to electrical interferences and are more expensive [25].

iii. Dual Technology Motion Sensors

This type makes use of the combination of two sensors. They are less likely to commit a false positive alarm because both sensors must be triggered in order to activate the alarm [25].

iv. Area Reflective Type

Basically, these sensors measure the distance of the object by using infrared rays from an LED source and checks whether the said object is within the designated area [25].

v. Ultrasonic

Similar to the MicroWave, they send out ultraviolet pulses and evaluates the reflection off a moving object [25].

vi. Vibration

They detect vibration and can easily be made at home with the use of a small mass on a lever but may be unreliable [25].

A..2 Fire Classification

Assuming that the nature and size of the fire is controllable, there are 5 classifications for fire according to the types of fuel they burn.

- Class A

  Fires under this class consist of ordinary combustibles like wood, paper, textile, rubber, trash and/or plastic. These types of fire can be extinguished using water extinguishers and dry chemical fire extinguishers [26]. Water extinguishers or APW extinguishers (air-pressurized water) can be used to extinguish class A fire only. Using such extinguishers to other classes may worsen the situation. Dry chemical fire extinguishers labeled as ABC type are commonly called as multipurpose dry chemical extinguishers. This type is filled with monoammonium phosphate, a yellow powder that leaves a sticky residue which may be harmful to electrical appliances such as a computer [27].

- Class B

  Class B fires are fueled by flammable liquids like oil, kerosene, grease and gasoline. Depleting the oxygen supply by smothering works best in these types of fires [26, 27]. A regular type of dry chemical may also be used in extinguishing class B fire. This is filled with sodium bicarbonate or potassium bicarbonate which leaves a mildly corrosive residue and must be cleaned as soon as possible to reduce further damage to materials. The ABC type dry chemical may also be used for class B fires. Carbon dioxide extinguishers can be used for class B fires. These contain carbon dioxide which is a non-flammable gas and is highly pressurized and may be the cause for bits of dry ice to shoot out the nozzle [27].

- Class C

  Fires classified as class C fires involve electrical equipment and are known as energized electrical fires [26, 27]. Non-conductive extinguishing agents such as carbon dioxide extinguishers may be used for this type of fire. It is advisable to de-energize the circuit first then use the non-conductive agent [26].

- Class D

  Class D fires are caused by combustible metals like Titanium, Potassium, Sodium and Magnesium. Only dry powder agents may be used to extinguish class D fires. Such dry powder agents include graphite in powder form and sodium chloride in granular form [27].

- Class K

  Fire of this class involve cooking oils, grease or animal fat. This type of fire can be put out using fire extinguishers that contain a wet extinguishing agent which turns the cooking oil and fat that is serving as fuel to soap. The mentioned substance is said to be non-combustible and that it actually absorbs the heat from the flame which, in turn, will reduce the element needed by fire to continue burning [27].

B. Computer Vision

Humans use their eyes to see and their brains to understand what they are seeing [28]. Computer vision gives computers the ability to "perceive, process, and understand visual data such as images and videos" [29]. It is involved in "the automatic extraction, analysis and understanding of useful information" of the acquired image and/or video [28]. Computer vision has been applied in different fields including security systems, face recognition, image restoration, etc., but its application in fire detection is not yet known especially in the Philippines.

C. Fire Color Classifier

There are different ways in detecting fire from the video/image processing and one of the famous ones is by simply analyzing the Red, Blue and Green channels of an RGB image. The value of the Red channel must be greater than the average of all the Red channel values of all pixels within the image [7]. Moreover, the value of the Red channel is said to be greater than that of the Green channel and the value of the Green channel is greater than that of the Blue channel in order for the pixel to be considered as part of the possible fire region [30]. In addition to that, features present in a fire are compared to the region being analyzed to further prove the presence of fire in that region. These features include: (a) randomness of area size, which is caused by the flickering of the flame and occurs from frame to frame; (b) boundary roughness, which is the randomness or roughness of the shape of the fire since fire doesn't have a specific boundary characteristics; (c) texture, refers to the significant amount of texture characteristics in a fire because of its random nature; and (d) lastly skewness, the degree of asymmetry of a distribution around its mean that is retrieved from the histogram of the image being analyzed and for the region to be considered as fire the skewness of its distributions must have a high negative [30].

Another way of detecting fire in the image is by converting the RGB image to its HSI counterpart. In the HSI color space, H refers to the hue of the image, S refers to the saturation of the image and I refers to the intensity of the image. HSI color model is said to be often applied due to its suitability for being a more people-oriented color space as compared to the other color spaces [31]. It refers to hue, saturation and intensity [32]. For the pixel to be considered as a part of the fire region, its hue, saturation and intensity value must be between 0 and 60, 0 an 0.2, and 127 an 255, respectively [31].

D. CCTV

CCTV systems have three types which are analog CCTV system, IP (Internet Protocol) CCTV system and hybrid CCTV system. An IP CCTV system makes use of only IP cameras which are connected to the system over the network while an analog CCTV system uses only analog cameras which are connected to the system using a single coax cable for each camera and a hybrid system has both analog an IP cameras [33, 34, 35]. An analog image sensor, which can either be CCD (charge coupled device) or CMOS (complementary metal oxide semiconductor), is utilized in both cameras in their recording process. Essentially, all analog cameras use CCD sensors while IP cameras use either of the two. These sensors are the ones responsible for image capture [33].

D..1 Cameras in CCTV Systems

1. Analog Camera

   In an analog camera, the image sensors and signal sensors are used to convert the optical information into digital signals for processing and optimization, which is also the case for IP cameras, and converted back to analog signals to be transmitted to an analog device such as a video monitor or a digital video recorder (DVR) [33, 36]. Moreover, the maximum resolution of an analog camera is said to be 720x575 [35].

2. IP Camera

   An Internet Protocol camera is similar to an analog camera in capturing an image. However, they are different in a way that the IP camera doesn't convert back to analog signals in transmitting its signals. The video signals are first compressed by the cameras and are transmitted over the network to a PC or a network video recorder (NVR). IP cameras also run their own operating system, much like a computer, but their functions are fixed and the programming are not supposed to be altered by the user. As compared to analog cameras, IP cameras has much better resolution with 720x480 or 720x576 as the resolution of the lowest end IP cameras [35].

Despite the fact that IP cameras are better in surveillance than analog cameras, many still prefer analog cameras due to their low cost. An IP camera is much more expensive than an analog camera because it has a built-in encoder but usually, it doesn't come packaged with outdoor housing and infrared light emitters, which is most of the time included in analog cameras [34].

D..2 Other types of CCTV Camera

1. PTZ Camera PTZ cameras have the ability to pan/tilt/zoom where they got their name. However, the biggest disadvantage of a ptz camera also comes in its feature. Once it has changed its direction, it won't be able to see what happened in the previous direction if ever there is an event. Moreover, because of its ptz ability this type of camera can monitor large areas and can also allow its operator to control it [37].

2. Fixed Camera Fixed cameras are cheaper than the ptz camera because it is

fixed in one direction only [38]. This type of camera is perfect for observing a specific point of interest [39]. In addition to that, the most common type of a fixed camera is the dome CCTV camera which is shaped like a dome and this makes the direction of the camera difficult to tell [3].

3. Infrared Camera Infrared cameras use infrared light to capture images in what seemed to be complete darkness to our eyes. Most infrared cameras come packaged with an infrared cut filter which is used to block the infrared light during daytime so as not to distort the color in the images [40].

IV. Design and Implementation

A. Use Case Diagram

Figure 1 shows the use case diagram of the Fire Detection System using CCTV. It has two actors - the Building Administrator and the Building Occupant. The Building Administrator will be the one in charge of managing the Fire Department's information fed to the system and the addition/removal of Building Occupants. S/He will also be the one to set the alert levels.

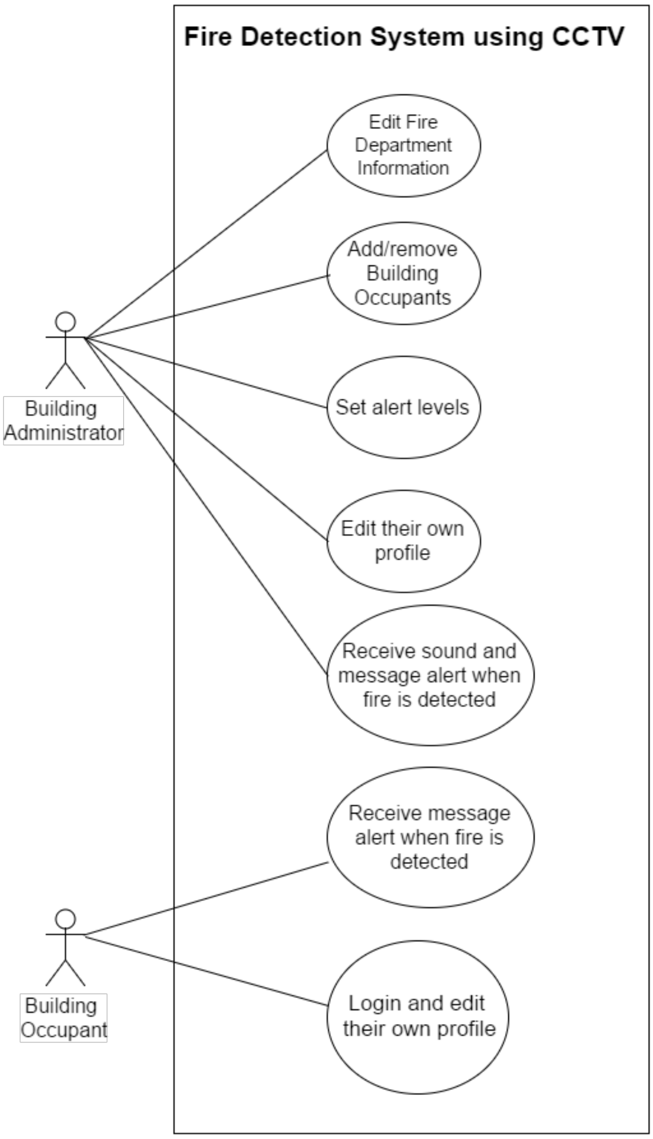

Fire Detection System using CCTV
Edit Fire Department Information
Add/remove Building Occupants
Set alert levels
Edit their own profile
Receive sound and message alert when fire is detected
Receive message alert when fire is detected
Login and edit their own profile
Building Administrator
Building Occupant

Figure 1: Use Case Diagram

B. Activity Diagram

Figure 2 shows the activity diagram of the Fire Detection System using CCTV. It shows the flow of the processes in the system. It starts by acquiring the video from the CCTV camera and the pre-processing will start with checking the RGB channel of each pixel if they follow the color model for the RGB color space. If they do follow the said color model, the coordinates of the pixel will be marked and saved to the RGB mask. Then, the RGB color space will be converted to the YCbCr color space by using the converting function in OpenCV. In the YCbCr color space, the YCbCr channel of each pixel will be checked if they follow the color model for the YCbCr color space and if they do follow the said color model, the coordinates of the pixel will be marked and saved to the YCbCr mask. The two masks generated, RGB mask and YCbCr mask, will be added to form the final fire mask. The contour of the said final fire mask will be extracted and used to draw on the original frame. The total contour area is computed and is checked whether it has reached any of the level alerts predetermined by the Building Administrator. If it reached a level alert, a sound and message alert will be created by the system.

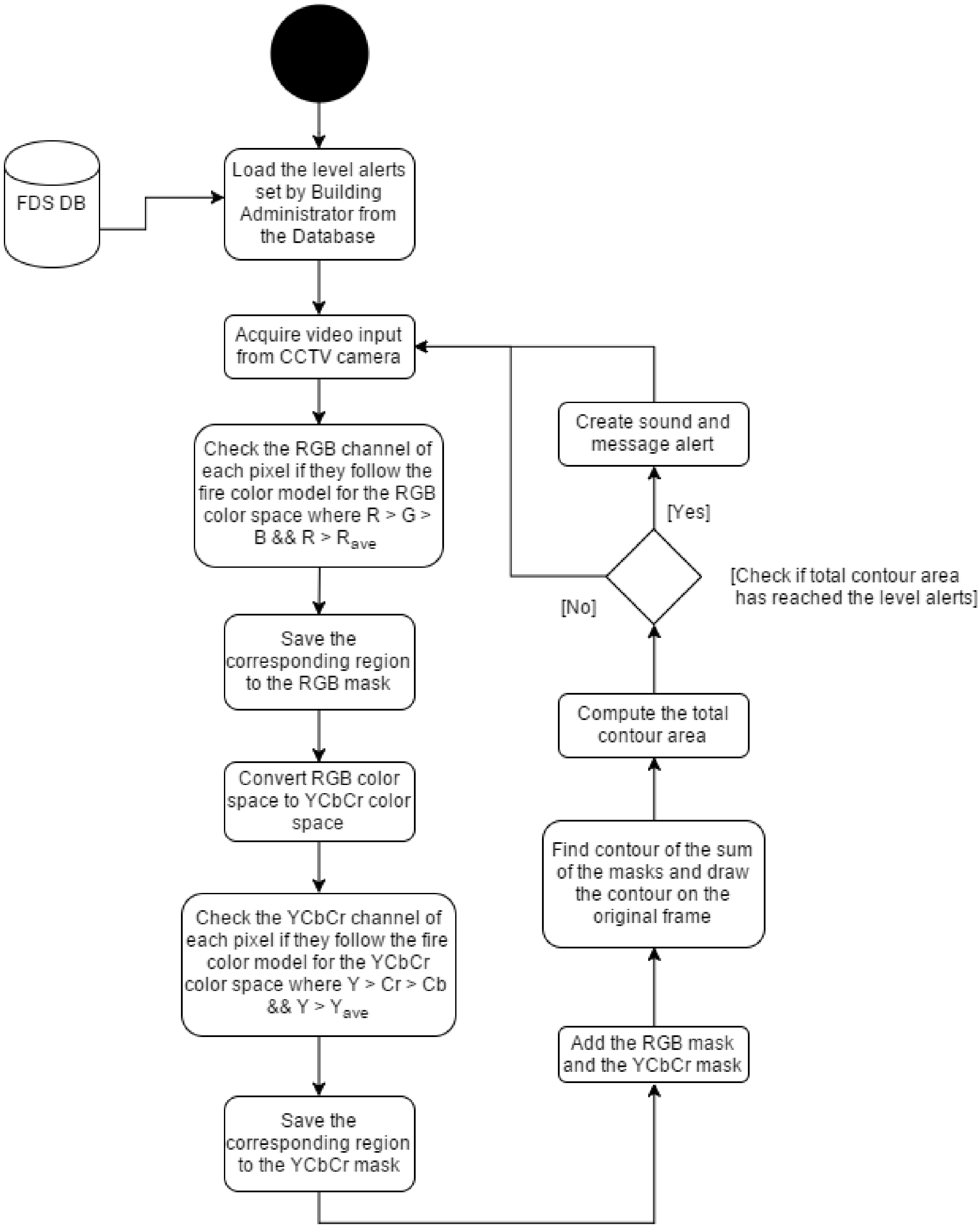
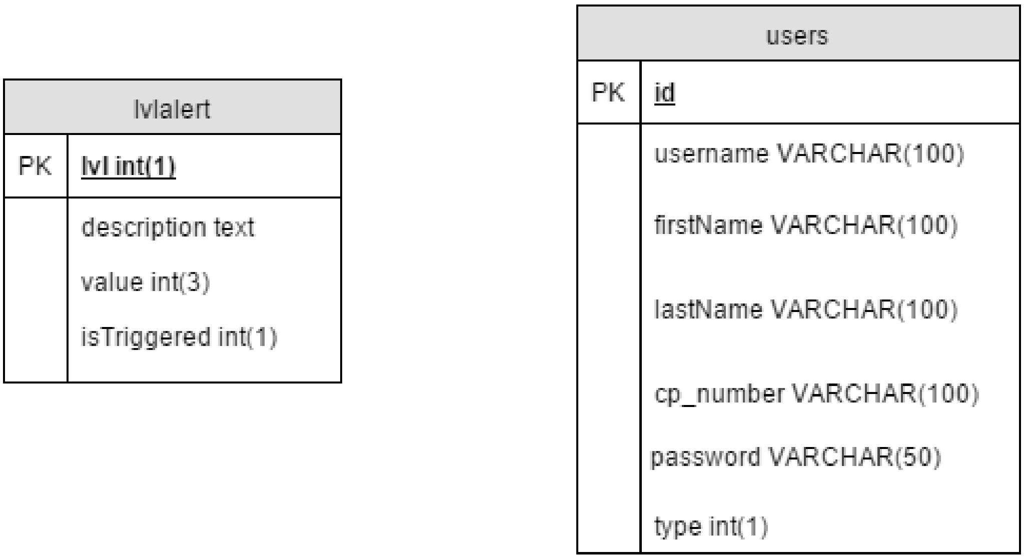

FDS DB
Load the level alerts set by Building Administrator from the Database
Acquire video input from CCTV camera
Check the RGB channel of each pixel if they follow the fire color model for the RGB color space where R > G > B && R > R_{ave}
Save the corresponding region to the RGB mask
Convert RGB color space to YCbCr color space
Check the YCbCr channel of each pixel if they follow the fire color model for the YCbCr color space where Y > Cr > Cb && Y > Y_{ave}
Save the corresponding region to the YCbCr mask
Add the RGB mask and the YCbCr mask
Find contour of the sum of the masks and draw the contour on the original frame
Compute the total contour area
[Check if total contour area has reached the level alerts]
[No]
[Yes]
Create sound and message alert

Figure 2: Activity Diagram

C. Entity Relationship Diagram

Figure 3 shows the entity relationship diagram of the Fire Detection System using CCTV. There is no direct relationship between the two tables/entities that is why there is no line connecting the lvlalert table to the users table.



lvlalert

PK **lvl int(1)**

description text

value int(3)

isTriggered int(1)

users

PK **id**

username VARCHAR(100)

firstName VARCHAR(100)

lastName VARCHAR(100)

cp_number VARCHAR(100)

password VARCHAR(50)

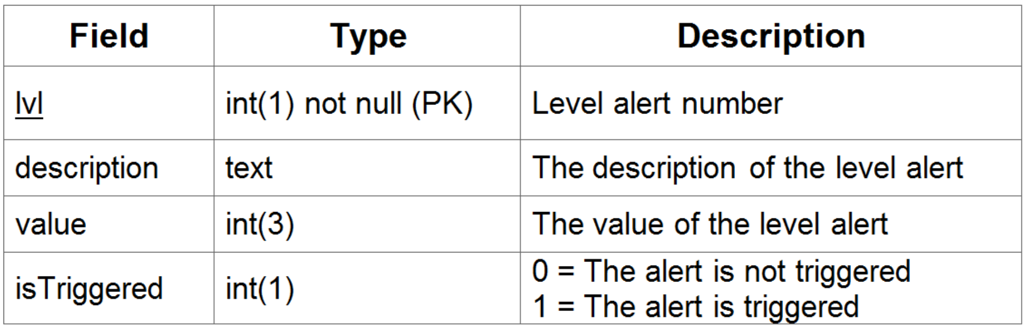
type int(1)

Figure 3: Entity Relationship Diagram

D. Data Dictionary

The database schema of the Fire Detection System contains the following tables and their fields:

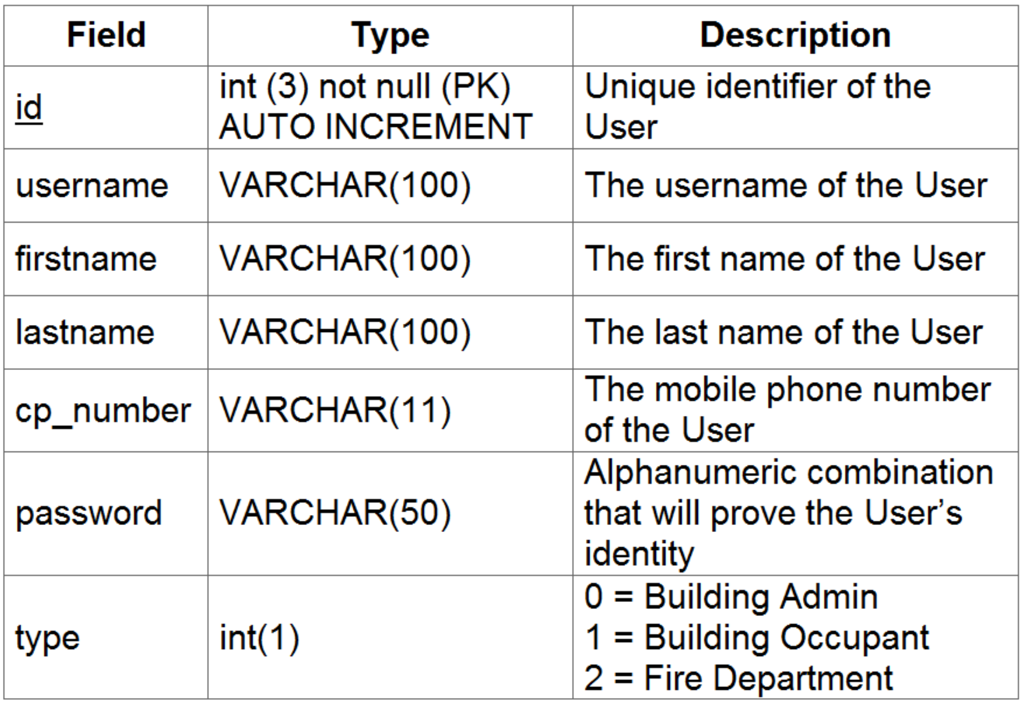
The *lvlalert* table contains the level alerts set by the Building Administrator. Initially, the *isTriggered* field is set to 0.



Field	Type	Description
lvl	int(1) not null (PK)	Level alert number
description	text	The description of the level alert
value	int(3)	The value of the level alert
isTriggered	int(1)	0 = The alert is not triggered 1 = The alert is triggered

Table 1: *lvlalert* table

The *users* table has the minimum necessary information about the users. All types of users are included in this table and they only differ in the type field.



Field	Type	Description
id	int (3) not null (PK) AUTO INCREMENT	Unique identifier of the User
username	VARCHAR(100)	The username of the User
firstname	VARCHAR(100)	The first name of the User
lastname	VARCHAR(100)	The last name of the User
cp_number	VARCHAR(11)	The mobile phone number of the User
password	VARCHAR(50)	Alphanumeric combination that will prove the User's identity
type	int(1)	0 = Building Admin 1 = Building Occupant 2 = Fire Department

Table 2: *users* table

E. System Architecture

The user interface of the Fire Detection System using CCTV has been implemented in PHP(Hypertext Preprocessor) with the use of the CodeIgniter framework that uses the MVC(Model-View-Controller) architecture. The part where the user can interact with the system is the view part and the system interacts with the MySQL database through the model. Meanwhile, the controller is the one connecting the model to the view.

F. Technical Architecture

The following were used in the development of the Fire Detection System using CCTV:

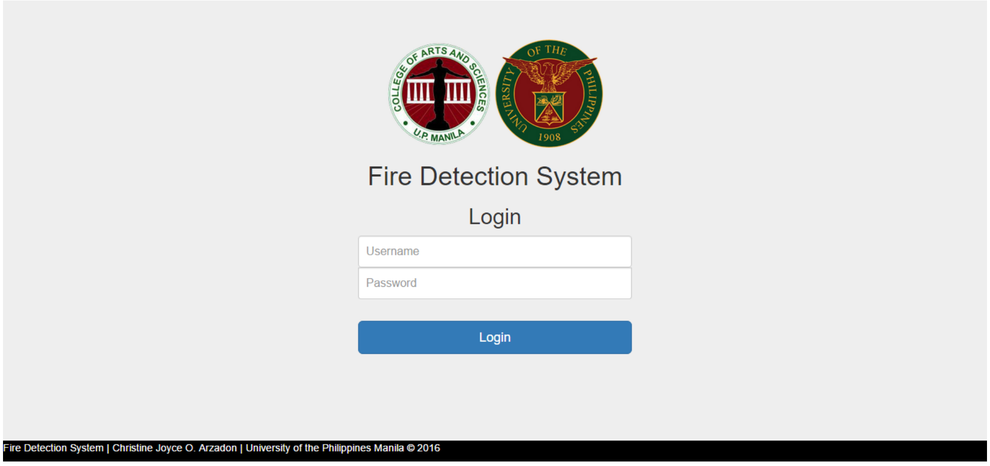
- *Database*: MySQL
- *Web Server*: Apache Tomcat
- *Programming Language*: PHP and C++
- *IDE*: Codeblocks 16.0
- *Library*: OpenCV 3.1

F..1 Minimum system requirements

- 1 GB RAM
- 1.60 GHz processor
- 32-bit operating system
- Microsoft Windows 2000/XP/Vista/7/8/10
- Reliable internet connection

V. Results

The user will need to login before being able to interact with the system. Figure 4 shows the login page where the user can use the credentials given to them. The credentials of the Building Occupant will come from the Building Administrator and the credentials of the Building Administrator will initially come from the person who will install the system.



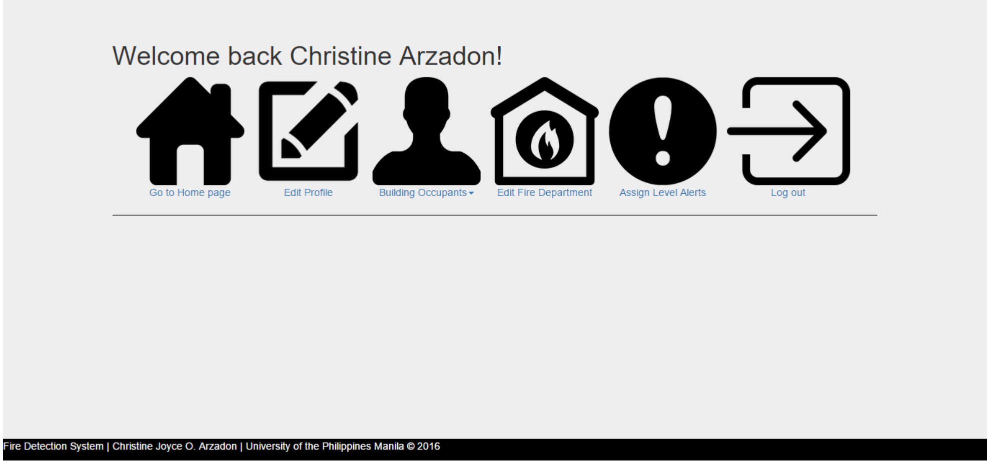
Fire Detection System

Login

Fire Detection System | Christine Joyce O. Arzadon | University of the Philippines Manila © 2016

Figure 4: Login Page

Figure 5 shows the landing page when the Building Administrator enters the system. S/He can select a functionality by pressing the icons.

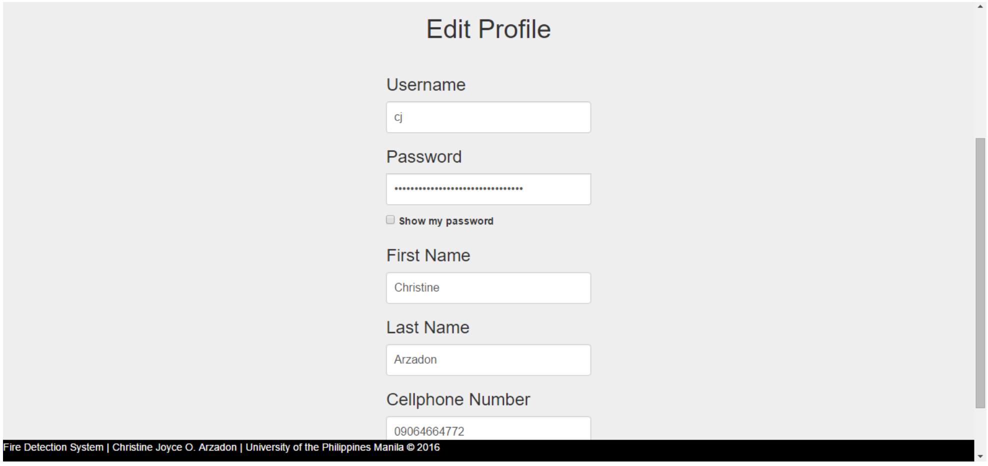


Welcome back Christine Arzadon!

Fire Detection System | Christine Joyce O. Arzadon | University of the Philippines Manila © 2016

Figure 5: Building Administrator's Home Page

Figure 6 shows the page where the Building Administrator can edit his/her own profile information.



Edit Profile

Username: cj

Password: ••••••••••••••••••••••••••••••••

□ Show my password

First Name: Christine

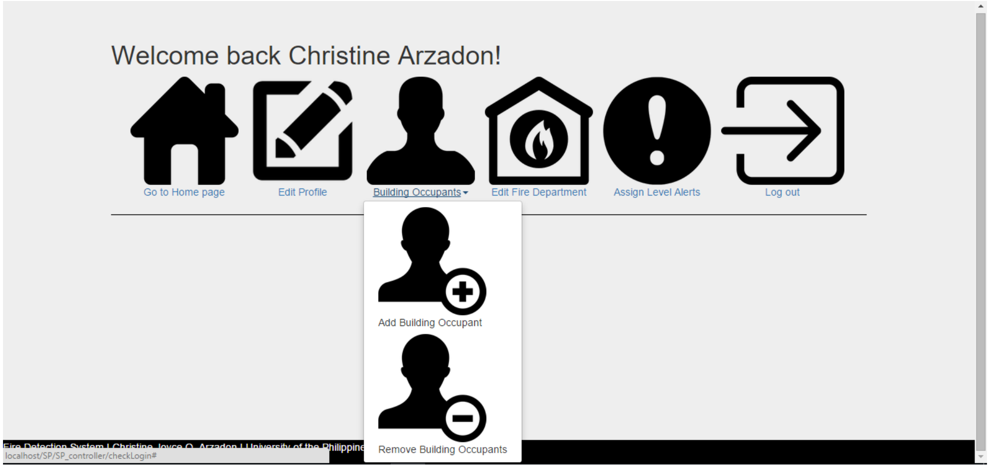
Last Name: Arzadon

Cellphone Number: 09064664772

Fire Detection System | Christine Joyce O. Arzadon | University of the Philippines Manila © 2016

Figure 6: Edit Profile Page

Figure 7 shows the Menu where it shows the capability of the Building Administrator to manage the Building Occupants.



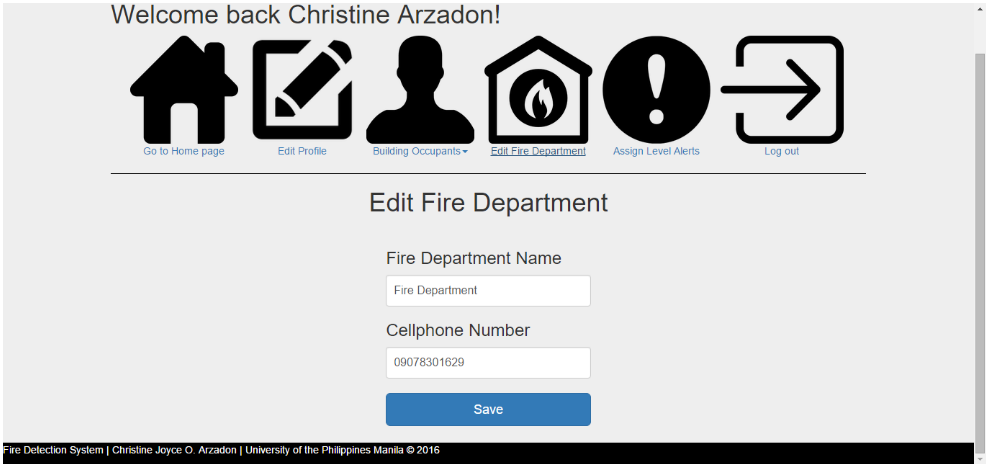
Welcome back Christine Arzadon!

Fire Detection System | Christine Joyce O. Arzadon | University of the Philippine

localhost/SP/SP_controller/checkLogin#

Figure 7: Building Occupant Toggle Menu

Figure 8 show the page where the Building Administrator can edit the Fire Department Information.


Welcome back Christine Arzadon!

Edit Fire Department

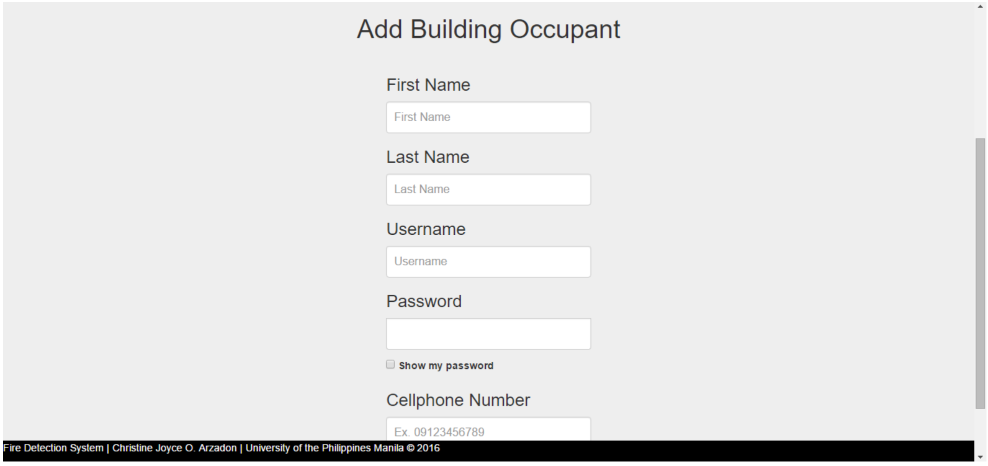
Fire Department Name
Fire Department

Cellphone Number
09078301629

Fire Detection System | Christine Joyce O. Arzadon | University of the Philippines Manila © 2016

Figure 8: Edit Fire Department Page

Figure 9 shows the page where the Building Administrator can input information regarding the new Building Occupant.


Add Building Occupant

First Name
First Name

Last Name
Last Name

Username
Username

Password

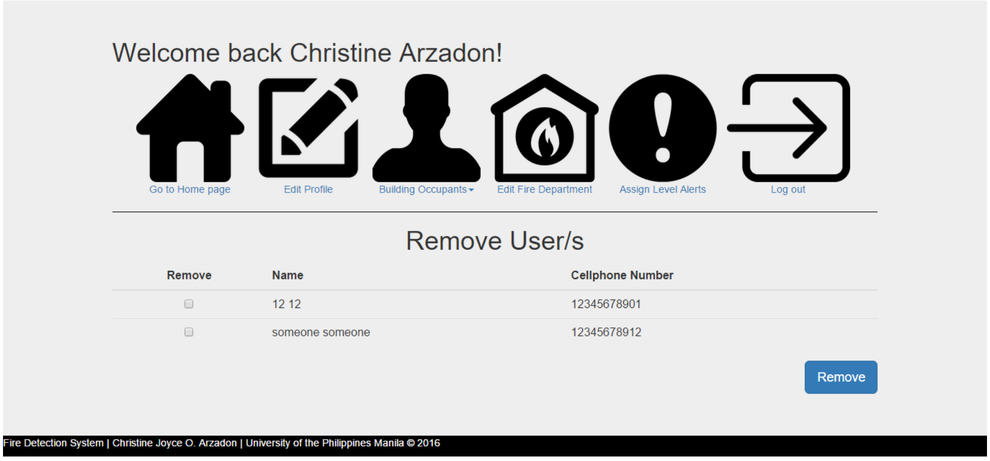
□ Show my password

Cellphone Number
Ex. 09123456789

Fire Detection System | Christine Joyce O. Arzadon | University of the Philippines Manila © 2016

Figure 9: Add Building Occupant Page

Figure 10 shows the page where the Building Administrator can remove one or more Building Occupants.



Welcome back Christine Arzadon!

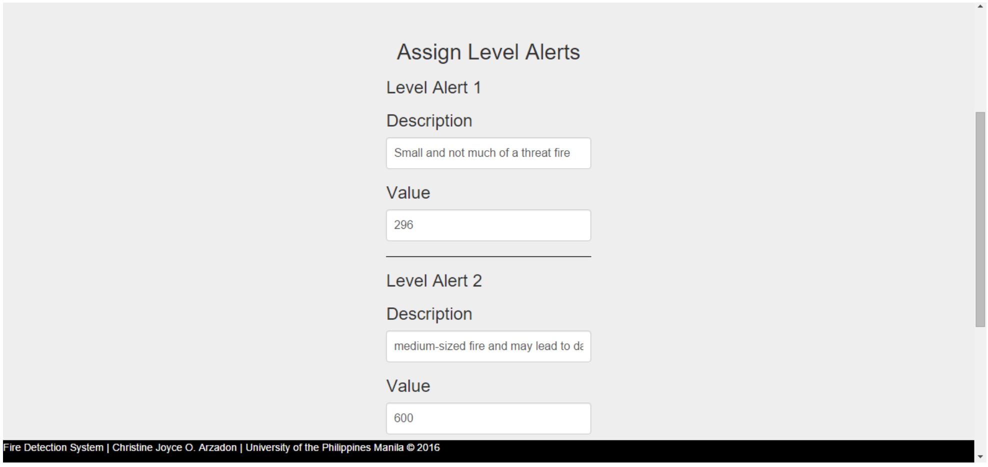
Remove User/s

Remove	Name	Cellphone Number
□	12 12	12345678901
□	someone someone	12345678912

Fire Detection System | Christine Joyce O. Arzadon | University of the Philippines Manila © 2016

Figure 10: Remove User/s Page

Figure 11 show the page where the Building Administrator can assign the level alerts needed by the fire detection part of this system. It is advisable to set the alert levels first before starting the fire detection program.



Assign Level Alerts

Level Alert 1

Description: Small and not much of a threat fire

Value: 296

Level Alert 2

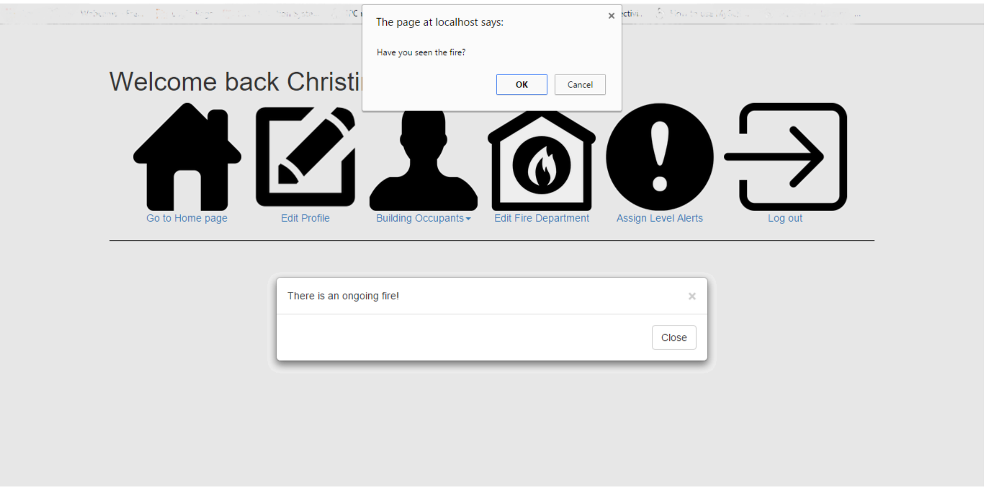
Description: medium-sized fire and may lead to da

Value: 600

Fire Detection System | Christine Joyce O. Arzadon | University of the Philippines Manila © 2016

Figure 11: Assign Level Alerts Page

Figure 12 shows what the Building Administrator's page will look like during the occurrence of a fire. The notifications will not stop until the fire has been put out. It is advisable to take care of the fire as soon as it has been detected by the system.


The page at localhost says:

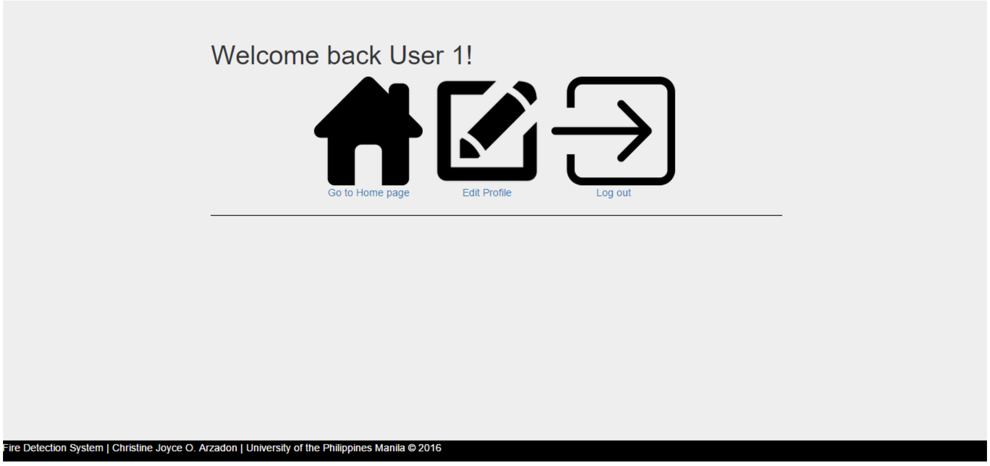
Have you seen the fire?

Welcome back Christi

There is an ongoing fire!

Figure 12: Building Administrator's page during the occurrence of a fire

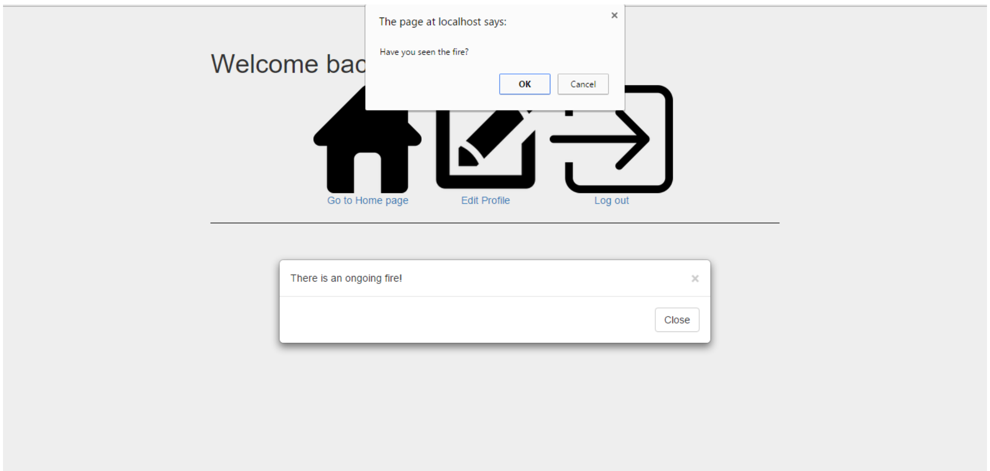
Figure 13 shows the landing page when the Building Occupant enters the system. S/He can select a functionality by pressing the icons.


Welcome back User 1!

ire Detection System | Christine Joyce O. Arzadon | University of the Philippines Manila © 2016

Figure 13: Building Occupant's Home page

Figure 14 shows what the Building Occupant's page will look like during the occurrence of a fire. The notifications will not stop until the fire has been put out. It is advisable to take care of the fire as soon as it has been detected by the system.


The page at localhost says:

Have you seen the fire?

There is an ongoing fire!

Figure 14: Building Occupant's page during the occurrence of a fire

As soon as the fire detection process is started, the program will immediately look for the fire region and draw its contour/s as shown in figure 15.

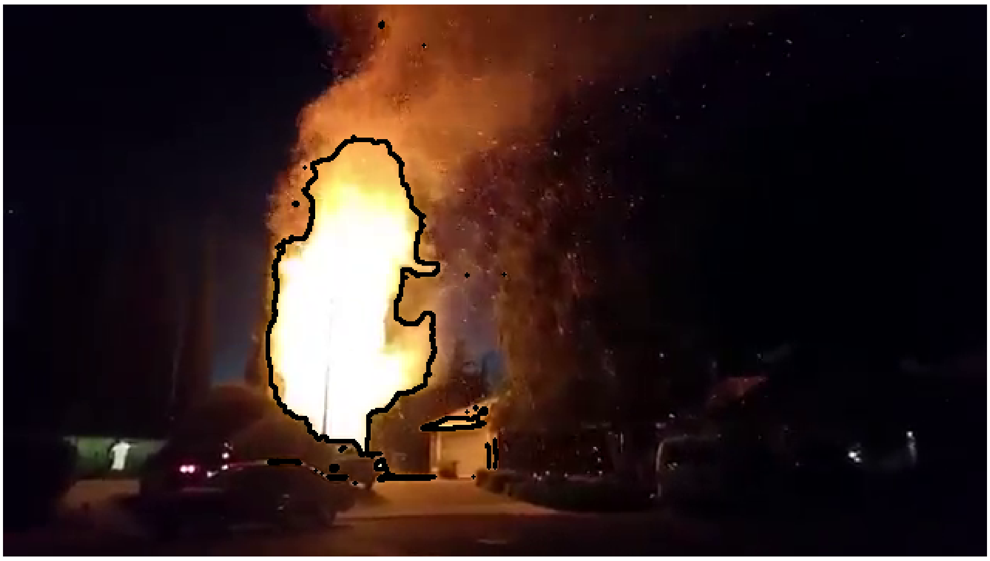


Figure 15: example of fire detected

VI. Discussions

The web interface of the Fire Detection System using CCTV allows the Building Administrator to manage Building Occupants, edit Fire Department information, set alert levels and receive sound and message alerts during the occurrence of fire. This system also allows the Building Occupant to receive message alerts during the occurrence of fire.

In the Edit Profile page, the user can edit their own information saved in the database. If the user doesn't want to change their some parts of their profile, then leaving it and not editing the corresponding textboxes will not change that part of their information. In the Remove User/s page, the Building Administrator can remove one or more Building Occupants at the same time.

For the purpose of showing the ability of the system to detect actual fire, example of fire videos was fed into the system. An example of a detected fire can be seen at figure 15. If the total contour of the detected fire has reached the level alerts set by the Building Administrator, the system will create sound and message alert to inform the users of the occurrence of fire. Figure 12 and figure 14 show what the user's page will look like during the occurrence of the fire. To show that non-fire objects were not detected by the system, the webcam of the machine used for presentation is used for the video input. For some reasons, human skin under good lighting is detected. Fortunately, the values of the level alerts are set at a reasonable point that is why no alerts were triggered.

The HSV color space may also be used in detecting fire regions. However, the equation presented at the journals used as references for the HSV color space included false positive regions. As long as the region is glowing white, the region is considered as fire region if the HSV color space is used.

The RGB and YCbCr color spaces were used in detecting the fire region. In the RGB color space, the RGB channels of each pixel is checked if they follow the color model for the RGB color space. If they do follow the said color model, the coordinates of the pixel will be marked and saved to the RGB mask. Then, the

RGB color space was converted to the YCbCr color space by using the converting function in OpenCV. In the YCbCr color space, the YCbCr channels of each pixel will be checked if they follow the color model for the YCbCr color space and if they do follow the said color model, the coordinates of the pixel will be marked and saved to the YCbCr mask. The two masks generated, RGB mask and YCbCr mask, will be added to form the final fire mask. The contour of the said final fire mask will be extracted and used to draw on the original frame. The total contour area is computed and is checked whether it has reached any of the level alerts predetermined by the Building Administrator. If it reached a level alert, a sound and message alert will be created by the system. After doing the procedure mentioned above, warm-colored areas were included because of the RGB mask and the more fire-colored regions were included because of the YCbCr mask. Because of the fact that the two masks were added, a more filtered mask was generated. The said mask excluded the non-fire region previously included before the addition of the two masks.

VII. Conclusions

In this paper, we developed a fire detection system while taking advantage of the CCTV system already installed inside the building. The users of this system are the Building Administrator or the Head of the Building and the Building Occupants or those who live inside the building. A separate C++ program is running to detect fire from the CCTV footage. A sound alert will be heard if fire is detected. This is important so as to obtain the attention of the people within the building.

VIII. Recommendations

This Fire Detection System using CCTV can be improved by looking further for ways to announce the occurrence of the fire. The automatic sending of SMS text message whenever a fire is detected can greatly improve the alerting capability of the system. It would also make the system better if manufacturers of Digital Video Recorders can include this program in their product so as to accommodate multiple cameras. Further research in computer vision in the area of detecting fire in grayscale videos can also improve the system since they are used in CCTV systems because grayscale videos use lesser memory space as compared to colored videos. Another area for improvement is the computation for the total contour area. Currently, the number of pixels is used to compute. The system would be easier to understand if the percentage of fire relative to the screen is used for computing the total contour area. Detecting smoke within the view of the camera will also fasten the fire detection process because smoke is said to be a primary indication of a possible fire. In addition to that, further improvement of the algorithm used will also increase the potential of this system. The system currently employs RGB and YCbCr color spaces in detecting the fire region and fire detection in other color spaces will greatly increase the power of the filter to exclude non-fire regions.

IX. Bibliography

[1] M. Rouse, "Cctv (closed circuit television)." http://whatis.techtarget.com/definition/CCTV-closed-circuit-television. Last accessed March 10, 2016.

[2] J. Wilson, "What type of cctv camera should i buy?." http://www.sonitrolwesterncanada.com/blog/what-type-of-cctv-camera-should-i-buy. Last accessed March 10, 2016.

[3] J. Knott, "12 common types of security cameras." http://www.cepro.com/article/12_common_types_of_security_cameras. Last accessed March 10, 2016.

[4] cosmotechph, "10 different types of cctv cameras and their purposes." https://cosmotechph.wordpress.com/2014/09/08/10-different-types-of-cctv-cameras-and-their-purposes/. Last accessed March 10, 2016.

[5] todayquestions.com, "The cctv camera security act of 2012 simplified." http://law.todayquestions.com/291427-06576690/?title=The+CCTV+Camera+Security+Act+of+2012+Simplified. Last accessed March 11, 2016.

[6] R. B. Rodriguez and M. B. Rodriguez, "House bill 6672." http://www.congress.gov.ph/download/congrec/15th/3rd/15C_3RS-29-111312.pdf. Last accessed April 13, 2016.

[7] T. Celik, H. Demirel, H. Ozkaramanli, and M. Uyguroglu, "Fire detection using statistical color model in video sequences," *Journal of Visual Communication and Image Representation*, vol. 18, no. 2, pp. 176 – 185, 2007.

[8] L. Russell, “Smoke detector problems and how to solve them.” http://www.home-security-systems-answers.com/smoke-detector-problems.html. Last accessed March 11, 2016.

[9] K. Willette, “Fire service guide to reducing unwanted fire alarms.” https://www.clackamasfire.com/documents/fireprevention/reducingunwantedfirealarms.pdf. Last accessed April 13, 2016.

[10] D. White, “Developments in optical fire detection.” http://www.fireworld.com/Archives/tabid/93/articleType/ArticleView/articleId/86730/Developments-in-Optical-Fire-Detection.aspx. Last accessed March 10, 2016.

[11] Fike, “Signifire video flame, smoke and intrusion detection system.” http://www.fike.com/products/signifire-video-flame-smoke-intrusion-detection-system/. Last accessed April 17, 2016.

[12] A. E. Cetin, “Computer vision based fire detection software.” http://signal.ee.bilkent.edu.tr/VisiFire/. Last accessed April 17, 2016.

[13] GKB, “Gkb video fire detection system.” http://www.gkbsecurity.com/VFDS/Download/VFDS_Intraduction.rar. Last accessed April 17, 2016.

[14] Y. H. Habiboglu, O. Gunay, and A. E. Cetin, “Real-time wildfire detection using correlation descriptors,” in *Signal Processing Conference, 2011 19th European*, pp. 894–898, Aug 2011.

[15] P. Santana, P. Gomes, and J. Barata, “A vision-based system for early fire detection,” in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pp. 739–744, Oct 2012.

[16] Y. H. Habibolu, O. Gnay, and A. E. etin, “Flame detection method in video using covariance descriptors,” in *Acoustics, Speech and Signal Process-*

ing (ICASSP), 2011 IEEE International Conference on, pp. 1817–1820, May 2011.

[17] T. X. Tung and J.-M. Kim, “An effective four-stage smoke-detection algorithm using video images for early fire-alarm systems,” *Fire Safety Journal*, vol. 46, no. 5, pp. 276 – 282, 2011.

[18] V. Vasyukov and E. Kalennikova, “An adaptive procedure of smoke and background discrimination in the early fire detection video system,” in *Strategic Technology (IFOST), 2011 6th International Forum on*, vol. 2, pp. 844–847, Aug 2011.

[19] J. Chi, C. Wu, and Y. Zhang, “Early fire disaster features research and detection method design in video images,” in *Photonics and Optoelectronics (SOPO), 2011 Symposium on*, pp. 1–4, May 2011.

[20] W. T. A. Budi and I. S. Suwardi, “Fire alarm system based-on video processing,” in *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on*, pp. 1–7, July 2011.

[21] W. A. H. W. M. Asmara and N. H. Aziz, “Sms flood alert system,” in *Control and System Graduate Research Colloquium (ICSGRC), 2011 IEEE*, pp. 18–22, June 2011.

[22] Lumar-Fire-Protection-Inc., “What are the differengt types of fire alarm and detection equipment?.” http://www.lumarfireprotection.com/what-are-different-types-of-fire-detection-equipment/. Last accessed June 1, 2016.

[23] Quinstreet-Enterprise, “motion detection.” http://www.webopedia.com/TERM/M/motion_detection.html. Last accessed April 13, 2016.

[24] S. Henry-Stocker, “How motion detection cameras work.” http://www.computerworld.com/article/2761095/it-management/

how-motion-detection-cameras-work.html. Last accessed April 13, 2016.

[25] safewise, "The beginner's guide to motion sensors." http://www.safewise.com/resources/motion-sensor-guide. Last accessed April 13, 2016.

[26] Falck-Safety-Services, "Classes of fire - a, b, c, d, and k." http://www.falckproductions.com/resources/fire-safety-and-firewatch/classes-of-fire-a-b-c-d-and-k/. Last accessed June 1, 2016.

[27] fire extinguisher101, "Fire extinguisher types (what kind do i use?)." http://www.fire-extinguisher101.com/. Last accessed June 1, 2016.

[28] A. Clark, "What is computer vision?." http://www.bmva.org/visionoverview. Last accessed April 5, 2016.

[29] Microsoft-Research, "Computer vision: Teaching computers to understand the visual world." http://research.microsoft.com/en-us/research-areas/computer-vision.aspx. Last accessed April 5, 2016.

[30] P. V. K. Borges, J. Mayer, and E. Izquierdo, "Efficient visual fire detection applied for video retrieval," in *Signal Processing Conference, 2008 16th European*, pp. 1–5, Aug 2008.

[31] C. Yu, Z. Mei, and X. Zhang, "A real-time video fire flame and smoke detection algorithm," *Procedia Engineering*, vol. 62, pp. 891 – 898, 2013. 9th Asia-Oceania Symposium on Fire Science and Technology.

[32] black ice, "Hsi color space - color space conversion." http://www.blackice.com/colorspaceHSI.htm. Last accessed March 11, 2016.

[33] Aventura-Technologies, "How a cctv camera works?." http://www.aventuracctv.com/analog_vs_ip/default.asp?index=1. Last accessed March 11, 2016.

[34] Kit-Communications, “Ip vs. analog: Video security: Which is better for me?.” http://www.kit-communications.com/ip-vs-analog-video.htm. Last accessed March 11, 2016.

[35] A. Majidimehr, “Understanding analog and ip tv cctv cameras.” http://www.monitoryourassets.com/ip-vs-analog/. Last accessed March 11, 2016.

[36] Abus.com, “Analog video surveillance the tried-and-tested classic.” http://www.abus.com/eng/Commercial-Security/Video-Surveillance/IP-HD-SDI-analog-HD-or-analog/Analog/Analog-technology. Last accessed March 11, 2016.

[37] J. Honovich, “Should you use ptz cameras?.” http://www.securityinfowatch.com/article/10492590/should-you-use-ptz-cameras. Last accessed March 11, 2016.

[38] CLV-Resources-Sdn-Bhd, “Types of cctv cameras.” http://www.clvresources.com/Type-of-CCTV-Cameras/. Last accessed March 11, 2016.

[39] Video-Surveillance, “Cctv video surveillance security camera system.” https://www.videosurveillance.com/cctv-technology/intro-to-cctv.asp. Last accessed March 11, 2016.

[40] K. Bowyer, “An overview of true day/night, infrared ip cameras and infrared lighting.” http://www.networkwebcams.co.uk/blog/2011/02/21/infrared-ip-cameras-security-surveillance-lighting/. Last accessed March 11, 2016.

X. Appendix

A. Source Code

Listing 1: main.cpp

```
//opencv
#include ”opencv2/imgcodecs.hpp”
#include ”opencv2/imgproc.hpp”
#include ”opencv2/videoio.hpp”
#include <opencv2/highgui.hpp>
#include <opencv2/video.hpp>
//C
#include <stdio.h>
//C++
#include <iostream>
#include <conio.h>
#include <sstream>
#include <fstream>
#include <string>
#include <iomanip>
#include <mysql.h>
//SQLAPI
#include <SQLAPI.h>

using namespace cv;
using namespace std;

// Global variables
Mat frame; //current frame
Mat fgMaskMOG2; //fg mask fg mask generated by
     MOG2 method
Mat fr;
vector<Mat> channels;
Ptr<BackgroundSubtractor> pMOG2; //MOG2
     Background subtractor
int keyboard; //input from keyboard
void processVideo();
void checkFrame(Mat frame);
void getValuesFromDB();
void setValueToDB();
SAConnection con;    // create connection
     object
SACommand cmd;       // create command object

vector<vector<Point> > contour;
int lvl1, lvl2, lvl3; // level alerts
float finalArea;

string cam;
int main()
{
    //create GUI windows
    //namedWindow(”Frame”);
    //namedWindow(”FG Mask MOG 2”);
    //create Background Subtractor objects
    getValuesFromDB();
    //setValueToDB();
    pMOG2 = createBackgroundSubtractorMOG2();
         //MOG2 approach
    processVideo();
    //destroy GUI windows
    destroyAllWindows();
    return EXIT_SUCCESS;

}

void getValuesFromDB(){
try{
    con.Connect(”fds”, ”root”, ””,
         SA_MySQL_Client);
    // associate a command with connection
         connection can also be specified in
         SACommand constructor
    cmd.setConnection(&con);
    cmd.setCommandText(”SELECT␣*␣FROM␣‘
         lvlalert‘”);
    cmd.Execute();
    while(cmd.FetchNext()){
        long lvl = cmd.Field(”lvl”).asLong();
        if(lvl == 1){
            lvl1 = cmd.Field(”value”).asLong()
                ;
        }else if(lvl == 2){
            lvl2 = cmd.Field(”value”).asLong()
                ;
        }else if(lvl == 3){
            lvl3 = cmd.Field(”value”).asLong()
                ;
        }
    }
    con.Destroy();
    //cout << ”Connected” << endl;
    }catch(SAException &x)
    {
        // SAConnection::Rollback()
        // can also throw an exception
        // (if a network error for example),
        // we will be ready
        try
        {
            // on error rollback changes
            con.Rollback();
        }
        catch(SAException &)
        {
        }
        // print error message
        printf(”%s\n”, (const char*)x.ErrText
             ());
    }
}

void setValueToDB(){
try{
    con.Connect(”fds”, ”root”, ””,
         SA_MySQL_Client);
    // associate a command with connection
         connection can also be specified in
         SACommand constructor
    cmd.setConnection(&con);
    cmd.setCommandText(”UPDATE␣‘lvlalert‘␣SET␣
         ‘isTriggered‘␣=␣’1’␣WHERE␣‘lvlalert
         ‘.‘lvl‘␣=␣1;”);
    cmd.Execute();
    con.Destroy();

    //cout << ”Connected” << endl;
    }catch(SAException &x)
    {
        // SAConnection::Rollback()
        // can also throw an exception
        // (if a network error for example),
        // we will be ready
        try
        {
            // on error rollback changes
            con.Rollback();
        }
        catch(SAException &)
        {
        }
        // print error message
        printf(”%s\n”, (const char*)x.ErrText
             ());
    }

}

void processVideo() {
    //create the capture object
    //VideoCapture capture(”videos/nightFire.
         mp4”);
    //VideoCapture capture(”videos/Gas
         Explosion.mp4”);
    //VideoCapture capture(”videos/Natural
         Fire Tinder.mp4”);
    //VideoCapture capture(”videos/Small Fire.
         mp4”);
    //VideoCapture capture(”videos/small
         magnesium fire.mp4”);
    //VideoCapture capture(”videos/unnamed.mp4
         ”);
    //VideoCapture capture(”videos/dayFire.mp4
         ”);
    //VideoCapture capture(0);

        //webcam
    cout << ”Input␣the␣url␣link␣to␣your␣IP␣
         camera:␣” << endl;
    cin >> cam;
    VideoCapture capture(cam);


    if(!capture.isOpened()){
        //error in opening the video input
        cerr << ”Unable␣to␣open␣video␣file:␣\n
             (make␣sure␣this␣link␣is␣working␣
             or␣else␣the␣program␣won’t␣start)”
              << endl;
        exit(EXIT_FAILURE);
    }
    //read input data. ESC or 'q' for quitting
    while((char)keyboard != 'q' && (char)
         keyboard != 27){
        //read the current frame
        if(!capture.read(frame)) {
            cerr << "Unable_to_read_next_frame
                 ." << endl;
            cerr << "Exiting..." << endl;
            exit(EXIT_FAILURE);
        }
        checkFrame(frame);
        int c1, c2, c3 = 0;
        for(int i = 0; i < 10; i++){
            if(finalArea > lvl3){
                c3++;
                capture.read(frame);
                checkFrame(frame);
            }else if(finalArea > lvl2){
                c2++;
                capture.read(frame);
                checkFrame(frame);
            }else if(finalArea > lvl1){
                c1++;
                capture.read(frame);
                checkFrame(frame);
            }
        }
        if(finalArea > lvl3 && c3 > 7){
            for(int i = 0; i < 4;i++){
                Beep(6000,500);
            }
            setValueToDB();
        }else if(finalArea > lvl2 && c2 > 7){
            for(int i = 0; i < 4;i++){
                Beep(3000,500);
            }
            setValueToDB();
        }else if(finalArea > lvl1 && c1 > 7){
            for(int i = 0; i < 4;i++){
                Beep(1000,500);
            }
            setValueToDB();
        }
        //get the input from the keyboard
        keyboard = waitKey( 30 );
    }
    //delete capture object
    capture.release();
}

    void checkFrame(Mat frame){
        // HSV Mask
        Mat hsv = frame.clone();
        Mat hsvMask;
        cvtColor(hsv, hsvMask, CV_BGR2HSV);
        inRange(hsvMask.clone(), Scalar(0, 0,
             127), Scalar(60, 0.2, 255), hsv);

        // RGB Mask
        Mat rgb = frame.clone();
        Scalar means = mean(frame);
        for(int i = 0; i<frame.rows; i++)
        {
            for(int j = 0; j<frame.cols; j++)
            {
                float blue = frame.at<Vec3b>(i
                    ,j)[0];
                float green = frame.at<Vec3b>(
                    i,j)[1];
                float red = frame.at<Vec3b>(i,
                    j)[2];
                //float gr = green/(red+1);
                //float br = blue/(red+1);
                //float bg = blue/(green+1);
                if((red > green) && (green >
                    blue) && (red > means[2])
                    )
                   //&& (gr >= 0.25) && (gr <=
                        0.65) &&
                   //(br >= 0.05) && (br <=
                        0.45) &&
                   //(bg >= 0.2) && (bg <=
                        0.6))
                {
                    rgb.at<Vec3b>(i,j)[0] =
                        255;
                    rgb.at<Vec3b>(i,j)[1] =
                        255;
                    rgb.at<Vec3b>(i,j)[2] =
                        255;
                }
                else
                {
                    rgb.at<Vec3b>(i,j)[0] = 0;
                    rgb.at<Vec3b>(i,j)[1] = 0;
                    rgb.at<Vec3b>(i,j)[2] = 0;
                }
            }
        }

        // YCbCr Mask
        Mat ycrcb = frame.clone();
        Mat ycrcbMask;
        cvtColor(ycrcb, ycrcbMask,
             CV_BGR2YCrCb);
        Scalar meansY = mean(ycrcbMask);
        for(int i = 0; i<ycrcbMask.rows; i++)
        {
            for(int j = 0; j<ycrcbMask.cols; j
                 ++)
            {
                int y = ycrcbMask.at<Vec3b>(i,
                     j)[0];
                int cr = ycrcbMask.at<Vec3b>(i
                     ,j)[1];
                int cb = ycrcbMask.at<Vec3b>(i
                     ,j)[2];
                if((y > cr) && (cr > cb) && (y
                      > meansY[0]) && (cr >
                     meansY[1]))
                {
                    ycrcb.at<Vec3b>(i,j)[0] =
                         0;
                    ycrcb.at<Vec3b>(i,j)[1] =
                         0;
                    ycrcb.at<Vec3b>(i,j)[2] =
                         0;
                }
                else
                {
                    ycrcb.at<Vec3b>(i,j)[0] =
                         255;
                    ycrcb.at<Vec3b>(i,j)[1] =
                         255;
                    ycrcb.at<Vec3b>(i,j)[2] =
                         255;
                }
            }
        }

        //imshow("YCrCb Mask", ycrcb);
        //imshow("RGB", rgb);
        cvtColor(hsv.clone(), hsv, CV_GRAY2BGR
             );
        ycrcb = rgb - ycrcb.clone();
        cvtColor(ycrcb.clone(), ycrcb,
             CV_BGR2GRAY);
        findContours(ycrcb.clone(), contour,
             CV_RETR_EXTERNAL,
             CV_CHAIN_APPROX_NONE);
        drawContours(frame, contour, -1,
             Scalar(0,0,0), 2);

        erode(ycrcb, ycrcb, Mat());
        dilate(ycrcb, ycrcb, Mat());

        //calculate area of fire region
        double area = 0;
        for(int i = 0;i < contour.size();i++){
            area += contourArea(contour[i]);
        }
        finalArea = area/contour.size();
        //cout << finalArea << endl;

        //update the background model
        pMOG2->apply(ycrcb, fgMaskMOG2, -1);
        erode(fgMaskMOG2, fgMaskMOG2, Mat());
        dilate(fgMaskMOG2, fgMaskMOG2, Mat());

        erode(rgb, rgb, Mat());
        dilate(rgb, rgb, Mat());

        //erode(hsv, hsv, Mat());
        //dilate(hsv, hsv, Mat());

        //findContours(fgMaskMOG2.clone(),
             contour,CV_RETR_EXTERNAL,
             CV_CHAIN_APPROX_NONE);
        //drawContours(frame, contour, -1,
             Scalar(255,255,255), 1);

        imshow("Original", frame);
    }
```

Listing 2: SP_controller.php

```
<?php
class SP_controller extends CI_Controller {

    function __construct()
    {
        parent::__construct();
    }

    public function index()
    {
```

```
    $data['main_content'] = 'login';
    $data['title'] = 'Login␣Page';
    $this->load->view('includes/template',
        $data);
}

public function homePage()
{
    if($this->session->userdata('
        isLoggedIn')){
        $this->load->model('SP_model');
        if($this->session->userdata('type'
            ) == 0){
            $query = $this->SP_model->
                getData('users');
            $lvl = $this->SP_model->
                getData('lvlalert');
            $data['records'] = $query;
            $data['lvl'] = $lvl;
            $data['main_content'] = 'user0
                ';
            $data['title'] = 'Home␣Page';
            $this->load->view('includes/
                template', $data);
        }else if($this->session->userdata(
            'type') == 1){
            $data['main_content'] = 'user1
                ';
            $lvl = $this->SP_model->
                getData('lvlalert');
            $query = $this->SP_model->
                getData('users');
            $data['lvl'] = $lvl;
            $data['records'] = $query;
            $data['title'] = 'Home␣Page';
            $this->load->view('includes/
                template', $data);
        }
    }
    else{
        $this->index();
    }
}

public function logOut()
{
    $this->session->unset_userdata('
        isLoggedIn');
    $this->index();
}

public function checkLogin()
{
    $this->load->model('SP_model');
    $query = $this->SP_model->login();

    if(!$query==false)
    {
        foreach($query as $querys)
        {
            $id = $querys->id;
            $username = $querys->username;
            $firstName = $querys->
                firstName;
            $lastName = $querys->lastName;
            $type = $querys->type;
            $cpNumber = $querys->cpNumber;
            $password = $querys->password;
        }
        $data = array(
            'isLoggedIn' => true,
            'id'         => $id,
            'username'   => $username,
            'firstName'  => $firstName,
            'lastName'   => $lastName,
            'cpNumber'   => $cpNumber,
            'password'   => $password,
            'type'       => $type // 0:
                building admins; 1:
                building occupants; 2:
                fire dept
        );
        $this->session->set_userdata($data
            );
        $this->homePage();
    }
    else
    {
        echo "<script␣type='text/
            javascript'>alert('Wrong␣
            password/username.');</script
            >";
        $this->index();
    }
}

public function updateUser()
{
    $this->load->model('SP_model');
    $id = $this->session->userdata('id');
    $data = array(
        'username'  => $this->input->post(
            'username'),
        'password'  => $this->input->post(
            'password'),
        'firstName' => $this->input->post(
            'firstName'),
        'lastName'  => $this->input->post(
            'lastName'),
        'cpNumber'  => $this->input->post(
            'cpNumber')
    );
    $this->SP_model->updateUser($id, $data
        );
    //$this->homePage();
    echo "<script␣type='text/javascript'>
        alert('Profile␣Updated!');</
        script>";
    redirect('SP_controller/homePage');

}

public function addBuildingOccupant()
{
    $this->load->model('SP_model');
    $data = array(
        'username'  => $this->input->post(
            'username1'),
        'password'  => md5($this->input->
            post('password1')),
        'firstName' => $this->input->post(
            'firstName1'),
        'lastName'  => $this->input->post(
            'lastName1'),
        'cpNumber'  => $this->input->post(
            'cpNumber1'),
        'type'      => 1
    );
    $this->SP_model->add('users', $data);
    echo "<script␣type='text/javascript'>
        alert('Building␣Occupant␣added.')
        ;</script>";
    redirect('SP_controller/homePage');
}

public function removeUsers()
{
    if(isset($_POST['records']))
    {
        $ids = $this->input->post('records
            ');
        foreach ($ids as $id) {
            $this->load->model('SP_model')
                ;
            $this->SP_model->remove($id, '
                users');
        }
        echo "<script␣type='text/
            javascript'>alert('Users␣
            removed!');</script>";
    }
    else
    {
        echo "<script␣type='text/
            javascript'>alert('No␣user␣
            was␣selected!');</script>";
    }
    redirect('SP_controller/homePage');
}

public function updateFireDept()
{
    $id = $this->input->post('id');
    $data = array(
        'username'  => $this->input->post(
            'usernameF'),
        'cpNumber'  => $this->input->post(
            'cpNumberF')
    );
    $this->load->model('SP_model');
    $this->SP_model->updateUser($id, $data
        );
    echo "<script␣type='text/javascript'>
        alert('Fire␣Department␣Info␣
        Updated!');</script>";
    redirect('SP_controller/homePage');
}

public function editLevel(){
    $this->load->model(SP_model);
    for($i = 1;$i < 4; $i++){
        $data = array(
            'description' => $this->input
                ->post('description'.$i),
            'value' => $this->input->post(
                'value'.$i)
        );
        $this->SP_model->updateLevelAlert(
                $i, $data);
        }
        echo "<script␣type='text/javascript'>
            alert('Level␣alerts␣Updated!');</
            script>";
        redirect('SP_controller/homePage');
    }

    public function updateTrigger($level){
        $this->load->model('SP_model');
        $data = array(
            'lvl' => $level,
            'isTriggered' => 0
            );
        $this->SP_model->updateLevelAlert(
            $level, $data);
        redirect('SP_controller/homePage');
    }
}

?>
```

Listing 3: SP_model.php

```
<?php
class SP_model extends CI_Model{

        public function getData($table)
        {
                $query = $this->db->get($table
                    );
                return $query->result();
        }

        public function login()
        {
                $this->db->where('username',
                    $this->input->post('
                    username'));
                $this->db->where('password',
                    md5($this->input->post('
                    password')));
                $query = $this->db->get('users
                    ');

                if($query->num_rows() == 1){
                        return $query->result
                            ();
                }
                return false;
        }

        public function updateUser($id, $data)
            {
                $this->db->where('id', $id);
                $this->db->update('users',
                    $data);
        }

        public function remove($id, $table){
                $this->db->where('id', $id);
                $this->db->delete($table);
        }

        public function add($table, $data){
                $this->db->insert($table,
                    $data);
        }

        public function updateLevelAlert(
            $level, $data){
                $this->db->where('lvl', $level
                    );
                $this->db->update('lvlalert',
                    $data);
        }
}
/*
        | ENCRYPTION FUNCTIONS
        |
        | encrypt( data, password, iv )
        |   - custom encrypt function
        |
        | decrypt( data, password, iv )
        |   - custom decrypt function

        public function encrypt( $data,
            $password, $iv )
        {
                return openssl_encrypt($data,
                    'aes-256-cbc', $password,
                    OPENSSL_RAW_DATA, $iv );
        }

        public function decrypt( $data,
            $password, $iv  )
        {
                return openssl_decrypt($data,
                    'aes-256-cbc', $password,
                    OPENSSL_RAW_DATA, $iv  );
        }
                                                /*

        | COOKIE FUNCTIONS
        |
        | getAdminCookie()
        |   - gets the cookie value of "
            theAdminCookie" and returns an
            object
        |
        | setAdminCookie( array )
        |   - sets the cookie value for "
            theAdminCookie"
        |
        | hasAdminCookie()
        |   - returns a value if the main
            cookie "theAdminCookie" is
            initialized
        |
        | deleteAdminCookie()
        |   - deletes the "theAdminCookie"
        |
        | constructAddAdminCookie()
        |   - constucts the "addAdminCookie"

        public function getAdminCookie()
        {
                $password = base64_encode(md5(
                    date('l jS \of F Y')));
                $iv = substr(base64_encode(md5
                    (date("Ymd"))),0,16);

                $unserialized_cookie =
                    base64_decode(get_cookie
                    ("theAdminCookie"));
                $decrypted_cookie = $this->
                    TAPT_Model->decrypt(
                    $unserialized_cookie,
                    $password, $iv );
                $cookie = json_decode(
                    $decrypted_cookie);

                return $cookie;
        }

        public function setAdminCookie( $array
            )
        {
                $password = base64_encode(md5(
                    date('l jS \of F Y')));
                $iv = substr(base64_encode(md5
                    (date("Ymd"))),0,16);

                $json = json_encode( $array );
                $encrypted_cookie = $this->
                    TAPT_Model->encrypt(
                    $json, $password, $iv );
                $encoded_cookie =
                    base64_encode(
                    $encrypted_cookie);
                set_cookie( "theAdminCookie",
                    $encoded_cookie,
                    (60*60*24) );
        }

        public function hasAdminCookie()
        {
                return (get_cookie("
                    theAdminCookie")) ? true
                    : false;
        }

        public function deleteAdminCookie()
        {
                delete_cookie("theAdminCookie
                    ");
        }

        public function
            constructAddAdminCookie()
        {
                $array["first_name"] = "";
                $array["last_name"] = "";
                $array["username"] = "";
                $array["password"] = "";
                $array["password_verify"] =
                    "";

                return json_decode(json_encode
                    ($array));
        }
*/
```

Listing 4: header.php

```
<!doctype>
<html>
        <head>
                <meta charset=”utf-8”>
                <meta http-equiv=”X-UA-
                     Compatible” content=”IE=
                     edge”>
                <meta name=”viewport” content=
                     ”width=device-width,␣
                     initial-scale=1”>
                <title><?php echo $title; ?></
                     title>
                <link href=”<?php␣echo␣
                     base_url();?>assets/css/
                     bootstrap.min.css” rel=”
                     stylesheet”>
                <link href=”<?php␣echo␣
                     base_url();?>assets/css/
                     signin.css” rel=”
                     stylesheet”>
                <script src=”<?php␣echo␣
                     base_url();?>assets/js/
                     jquery.min.js”></script>
                <script src=”<?php␣echo␣
                     base_url();?>assets/js/
                     bootstrap.min.js”></
                     script>
                <style>
                img{
                        height: 150px;
                }
                footer{
                        position: fixed;
                        bottom: 0px;
                        background-color:
                             black;
                        color: white;
                        height: 5%;
                        width: 100%;
                }
                </style>
        </head>
        <body>
```

Listing 5: template.php

```
<?php $this->load->view('includes/header'); ?>

<?php $this->load->view($main_content); ?>

<?php $this->load->view('includes/footer'); ?>
```

Listing 6: footer.php

```
</body>
<footer id=”footer” class=”black”>
        <div id=”123”>
                Fire Detection System |
                     Christine Joyce O.
                     Arzadon | University of
                     the Philippines Manila &
                     copy; 2016
        </div>
</footer>
</html>
```

Listing 7: login.php

```
<?php echo validation_errors() ?>
<form
        name=”login”
        method=”POST”
        class=”form-signin”
        action='<?php␣echo␣base_url();?>
             SP_controller/checkLogin'>
<div align=”center”>
        <img src=”<?php␣echo␣base_url();?>
             assets/images/cas_logo.png”>
        <img src=”<?php␣echo␣base_url();?>
             assets/images/up_logo.png”>
        <h1>Fire Detection System</h1>
</div>
<h2 class=”form-signin-heading” style=”text-
     align:␣center;”>Login</h2>
<input
        type=”text”
        name=”username”
        class=”form-control”
        placeholder=”Username” required>
<input
        type=”password”
        name=”password”
        class=”form-control”
        placeholder=”Password” required>
        <br>
<input
        class=”btn␣btn-lg␣btn-primary␣btn-
             block”
        type=”submit”
        value=”Login”>
</form>
```

Listing 8: user0.php

```
<div class=”container” style=”width:␣80%;”>
        <h1>Welcome back
                <?php
                        echo $this->session->
                             userdata('
                             firstName'). ”␣”
                             . $this->session
                             ->userdata('
                             lastName');
                ?>!
        </h1>
        <ul id=”home” class=”list-inline”
             align=”center”>
                <li><a href=”#” id=”h”>
                <figure>
                        <img alt=”Go␣to␣Home␣
                             page” src=”<?php␣
                             echo␣base_url()
                             ;?>assets/images/
                             home.png”>
                        <figcaption align=”
                             center”>Go to
                             Home page</
                             figcaption>
                </figure>
                </a></li>
                <li><a href=”#” id=”e”>
                <figure>
                        <img alt=”Edit␣Profile
                             ” src=”<?php␣echo
                             ␣base_url();?>
                             assets/images/
                             edit.png”>
                        <figcaption align=”
                             center”>Edit
                             Profile</
                             figcaption>
                </figure>
                </a></li>
                <li class=”dropdown”><a href=”
                     #” data-toggle=”dropdown”
                     >
                        <figure>
                                <img src=”<?
                                     php␣echo␣
                                     base_url
                                     ();?>
                                     assets/
                                     images/
                                     user.png”
                                     >
                                <figcaption>
                                     Building
                                     Occupants
                                     <span
                                     class=”
                                     caret”></
                                     span></
                                     figcaption
                                     >
                        </figure>
                        </a>
                                <ul class=”
                                     dropdown-
                                     menu”>
                                        <li><a

                                             href
                                             =
                                             ”
                                             #
                                             ”

                                             id
                                             =
                                             ”
                                             a
                                             ”
                                             >
                                        <
                                             figure
                                             >
                                                <
                                                     img

                                                     src
```

```
                       =      <li><a href="#" id="f">
                       "      <figure>
                       <?          <img src="<?php␣echo␣
                       php            base_url();?>
                       ␣              assets/images/
                       echo           fire_dept.png">
                       ␣           <figcaption>Edit Fire
                       base_url       Department</
                       ()             figcaption>
                       ;?>    </figure>
                       assets </a></li>
                       /      <li><a href="#" id="l">
                       images <figure>
                       /           <img src="<?php␣echo␣
                       add_user       base_url();?>
                       .              assets/images/
                       png            alert.png">
                       "           <figcaption>Assign
                       >              Level Alerts</
                                      figcaption>
                    <         </figure>
                       figcaption </a></li>
                       >      <li><a href="<?=base_url()?>
                       Add       SP_controller/logOut">
                              <figure>
                       Building    <img alt="Log␣out" src
                                      ="<?php␣echo␣
                       Occupant       base_url();?>
                       </             assets/images/
                       figcaption     logout.png">
                       >           <figcaption align="
               </                     center">Log out</
                                      figcaption>
                  figure      </figure>
                  >           </a></li>
               </a></    </ul>
                  li     <hr style="color:␣black;␣background-
                  >         color:␣black;␣height:␣1px;" />
               <li><a         <div id="edit" style="display:
                                 ␣none;">
                  href             <h1 align="center">
                  =                   Edit Profile</h1>
                  "                <form
                  #                     name="
                  "                        editProfile
                                           "
                  id                    method="POST"
                  =                     class="form-
                  "                        signin"
                  r                     action="<?php␣
                  "                        echo␣
                  >                        base_url
               <                           ();?>
                  figure                   SP_controller
                  >                        /
                    <                      updateUser
                       img                 ">
                                   <h3>Username</h3>
                       src         <input
                       =                type="text"
                       "                name="username
                       <?                  "
                       php              class="form-
                       ␣                   control"
                       echo             Value=<?=$this
                       ␣                   ->session
                       base_url            ->
                       ()                  userdata(
                       ;?>                 'username
                       assets              ')?> />
                       /           <h3>Password</h3>
                       images      <input
                       /                type="password
                       remove_user         "
                       .                name="password
                       png                 "
                       "                class="form-
                       >                   control␣
                                           input-
                    <                      password"
                       figcaption       title="Don't␣
                       >                   do␣
                       Remove              anything␣
                                           if␣you␣
                       Building            don't␣
                                           want␣your
                       Occupants           ␣password
                       </                  ␣to␣
                       figcaption          change."
                       >                Value=<?=$this
               </                          ->session
                                           ->
                  figure                   userdata(
                  >                        'password
               </a></                      ')?> />
                  li               <div>
                  >                     <input id="
          </ul>                            checkbox-
</li>                                      unmask"
```

```
           class="
           checkbox-
           unmask"
           type="
           checkbox"
            />
        <label for="
           checkbox-
           unmask"
           class="
           label-
           unmask">
           Show my
           password
           </label>
    </div>
    <h3>First Name</h3>
    <input
        type="text"
        name="
           firstName
           "
        class="form-
           control"
        Value=<?=$this
           ->session
           ->
           userdata(
           '
           firstName
           ')?> />
    <h3>Last Name</h3>
    <input
        type="text"
        name="lastName
           "
        class="form-
           control"
        Value=<?=$this
           ->session
           ->
           userdata(
           'lastName
           ')?> />
    <h3>Cellphone Number</
       h3>
    <input
        type="text"
        name="cpNumber
           "
        class="form-
           control"
        Value=<?=$this
           ->session
           ->
           userdata(
           'cpNumber
           ')?> />
    <br />
    <input
        class="btn␣btn
           -lg␣btn-
           primary␣
           btn-block
           "
        type="submit"
        value="Save">
    </form>
</div>
<div id="add" style="display:␣
   none;">
    <h1 align="center">Add
          Building
       Occupant</h1>
    <form
        name="addUser"
        method="POST"
        class="form-
           signin"
        action="<?php␣
           echo␣
           base_url
           ();?>
           SP_controller
           /
           addBuildingOccupant
           ">
    <h3>First Name</h3>
    <input
        type="text"
        name="
           firstName1
           "
        class="form-
           control"
        placeholder="
           First␣
           Name"
        required>
```

```
    <h3>Last Name</h3>
    <input
        type="text"
        name="
           lastName1
           "
        class="form-
           control"
        placeholder="
           Last␣Name
           "
        required>
    <h3>Username</h3>
    <input
        type="text"
        name="
           username1
           "
        class="form-
           control"
        placeholder="
           Username"
        required>
    <h3>Password</h3>
    <input
        type="password
           "
        name="
           password1
           "
        class="form-
           control␣
           inputpw"
        required>
    <div>
        <input id="
           unmask"
           class="
           unmask"
           type="
           checkbox"
           >
        <label for="
           unmask"
           class="
           label-
           unmask">
           Show my
           password
           </label>
    </div>
    <h3>Cellphone Number</
       h3>
    <input
        type="text"
        name="
           cpNumber1
           "
        class="form-
           control"
        placeholder="
           Ex.␣
           09123456789
           "
        required>
    <br />
    <input
        class="btn␣btn
           -lg␣btn-
           primary␣
           btn-block
           "
        type="submit"
        value="Add">
    </form>
</div>
<div id="del" style="display:␣
   none;">
<h1 align="center">Remove User
   /s</h1>
<form
    name="removeUsers"
    method="POST"
    class="form"
    action="<?php␣echo␣
       base_url();?>
       SP_controller/
       removeUsers">
    <table class="table">
        <thead>
        <tr>
            <th
               style
               =
               "
               text
               -
               align
               :
```

```
                         ␣
                         center
                         ;
                         ”
                         >
                         Remove
                         </
                         th
                         >
                   <th>
                         Name
                         </
                         th
                         >
                   <th>
                         Cellphone

                         Number
                         </
                         th
                         >
          </tr>
          </thead>
          <tbody>
<?php
          foreach(
               $records
               as
               $record){
                   if(
                         $record
                         −>
                         type
                         ==1)
                         {
?>
          <tr>
                   <td
                         style
                         =
                         ”
                         text
                         −
                         align
                         :
                         ␣
                         center
                         ;
                         ”
                         >
                             <
                                  input

                                  name
                                  =
                                  ”
                                  records
                                  []
                                  ”

                                  value
                                  =
                                  ”
                                  <?=
                                  $record
                                  −>
                                  id
                                  ?>
                                  ”

                                  type
                                  =
                                  ”
                                  checkbox
                                  ”

                                  />
                   </td>
                   <td>
                             <?=
                                  $record
                                  −>
                                  firstName
                                  .
                                  ”
                                  ␣
                                  ”
                                  .
                                  $record
                                  −>
                                  lastName
                                  ?>
                   </td>
                   <td>
                             <?=
                                  $record
                                  −>
                                           cpNumber
                                           ?>

                             </td>
                   </tr>
          <?php
                             }
                   }
          ?>
                   </tbody>
          </table>
          <input
                   type=”submit”
                   value=”Remove”
                   class=”btn␣btn
                        −lg␣btn−
                        primary”
                   style=”float:␣
                        right;”>
</form>
</div>
<div id=”fireDept” style=”
     display:␣none;”>
          <h1 align=”center”>
               Edit Fire
               Department</h1>
          <form
                   name=”
                        editFireDept
                        ”
                   method=”POST”
                   class=”form−
                        signin”
                   action=”<?php␣
                        echo␣
                        base_url
                        ();?>
                        SP_controller
                        /
                        updateFireDept
                        ”>
          <?php
                   foreach(
                        $records
                        as
                        $record){
                            if(
                                 $record
                                 −>
                                 type
                                 ==2)
                                 {
          ?>

          <input
                   type=”hidden”
                   name=”id”
                   Value=”<?=
                        $record−>
                        id?>”>
          <h3>Fire Department
               Name</h3>
          <input
                   type=”text”
                   name=”
                        firstNameF
                        ”
                   class=”form−
                        control”
                   Value=”<?=
                        $record−>
                        firstName
                        ?>”
                   required>
          <h3>Cellphone Number</
               h3>
          <input
                   type=”text”
                   name=”
                        cpNumberF
                        ”
                   class=”form−
                        control”
                   Value=”<?=
                        $record−>
                        cpNumber
                        ?>”
                   required>
          <br />
          <?php
                   }
          }
          ?>
          <input
                   class=”btn␣btn
                        −lg␣btn−
                        primary␣
                        btn−block
                        ”
```

```
                type="submit"
                value="Save">
        </form>
</div>
<div id="lvlAlert" style="
     display:_none;_">
        <form
                name="
                     editLevel
                     "
                method="POST"
                class="form-
                     signin"
                action="<?php_
                     echo_
                     base_url
                     ();?>
                     SP_controller
                     /
                     editLevel
                     ">
        <h2 align="center">
             Assign Level
             Alerts</h2>
        <?php
                foreach($lvl
                     as $level
                     ){
        ?>
                <h3>Level
                     Alert <?=
                     $level->
                     lvl?></h3
                     >
                <h3>
                     Description
                     </h3>
                <input
                        type="
                             textarea
                             "
                        name="
                             description
                             <?=
                             $level
                             ->
                             lvl
                             ?>
                             "
                        class=
                             "
                             form
                             -
                             control
                             "
                        value=
                             "
                             <?=
                             $level
                             ->
                             description
                             ?>
                             "
                        >
                <h3>Value</h3>
                <input
                        type="
                             number
                             "
                        name="
                             value
                             <?=
                             $level
                             ->
                             lvl
                             ?>
                             "
                        class=
                             "
                             form
                             -
                             control
                             "
                        value=
                             "
                             <?=
                             $level
                             ->
                             value
                             ?>
                             "
                        >
                <hr style="
                     color:_
                     black;_
                     background
                     -color:_
                     black;_
```

```
                                   height:_1
                                   px;" />
                  <?php
                          }
                  ?>
                  <input
                          class="btn_btn
                               -lg_btn-
                               primary_
                               btn-block
                               "
                          type="submit"
                          value="Save">
                  </form>

          </div>
</div>
<?php
          for($i = 0;$i < 3;$i++){
          if($lvl[$i]->lvl==($i+1)){
                  if($lvl[$i]->isTriggered==1){
?>

          <div class="modal-dialog">
          <div class="modal-content" id="myModal
               ">
          <div class="modal-body">
                  <button type="button"
                       class="close"
                       data-dismiss="
                       modal" data-show=
                       "true">&times;</
                       button>
             There is an ongoing fire!
          </div>
          <div class="modal-footer">
            <button type="button" class="btn_btn
                 -default" data-dismiss="modal">
                 Close</button>
          </div>
     </div>
</div>
             <script type='text/javascript'
                  >
                  function confirmation
                       () {
                          var answer =
                               confirm("
                               Have_you_
                               seen_the_
                               fire?")
                          if (answer){
                                  window
                                       .
                                       location
                                       =
                                       "
                                       <?
                                       php
                                       _
                                       echo
                                       _
                                       base_url
                                       ()
                                       ;?>
                                       SP_controller
                                       /
                                       updateTrigger
                                       /
                                       "
                                       +
                                       <?=
                                       $lvl
                                       [
                                       $i
                                       ]->
                                       lvl
                                       ?>;
                          }
                          else{
                                  ;
                          }
                  }
                  confirmation();
             </script>
<?php
                  }
          }
          }
?>
```

```
<script>
setInterval(function(){
      location.reload();
 },60000);

var pw = $(".input-password"),
        cb = $("#checkbox-unmask"),
        mask = true;

cb.on("click", function() {

        if (mask === true) {
                mask = false;
                pw.attr("type", "text");
        } else {
                mask = true;
                pw.attr("type", "password");
        }

});

var pw1 = $(".inputpw"),
        cb1 = $("#unmask"),
        mask1 = true;

cb1.on("click", function() {

        if (mask1 === true) {
                mask1 = false;
                pw1.attr("type", "text");
        } else {
                mask1 = true;
                pw1.attr("type", "password");
        }

});

var pw2 = $(".inputpwF"),
        cb2 = $("#unmaskF"),
        mask2 = true;

cb2.on("click", function() {

        if (mask2 === true) {
                mask2 = false;
                pw2.attr("type", "text");
        } else {
                mask2 = true;
                pw2.attr("type", "password");
        }

});

$(document).ready(function () {
        $('#h').click(function () {
        $('#edit').hide();
        $('#add').hide();
        $('#del').hide();
        $('#fireDept').hide();
        $('#lvlAlert').hide();
        $('#home').show();
    });
    $('#e').click(function () {
        $('#add').hide();
        $('#del').hide();
        $('#fireDept').hide();
        $('#lvlAlert').hide();
        $('#edit').show();
    });
    $('#a').click(function () {
        $('#edit').hide();
        $('#del').hide();
        $('#fireDept').hide();
        $('#lvlAlert').hide();
        $('#add').show();
    });
    $('#r').click(function () {
        $('#edit').hide();
        $('#add').hide();
        $('#fireDept').hide();
        $('#lvlAlert').hide();
        $('#del').show();
    });
    $('#f').click(function () {
        $('#add').hide();
        $('#del').hide();
        $('#edit').hide();
        $('#lvlAlert').hide();
        $('#fireDept').show();
    });
    $('#l').click(function () {
        $('#add').hide();
        $('#del').hide();
        $('#fireDept').hide();
        $('#edit').hide();
        $('#lvlAlert').show();
    });
});
</script>
```

Listing 9: user1.php

```
<div class="container" style="width:␣60%;">
        <h1>Welcome back
                <?php
                        echo $this->session->
                             userdata('
                             firstName'). "␣"
                             . $this->session
                             ->userdata('
                             lastName');
                ?>!
        </h1>

        <ul id="home" class="list-inline"
             align="center">
                <li>
                <a href="#" id="h">
                <figure>
                        <img alt="Go␣to␣Home␣
                             page" src="<?php␣
                             echo␣base_url()
                             ;?>assets/images/
                             home.png">
                        <figcaption align="
                             center">Go to
                             Home page</
                             figcaption>
                </figure>
                </a>
                </li>
                <li>
                <a href="#" id="e">
                <figure>
                        <img alt="Edit␣Profile
                             " src="<?php␣echo
                             ␣base_url();?>
                             assets/images/
                             edit.png">
                        <figcaption align="
                             center">Edit
                             Profile</
                             figcaption>
                </figure>
                </a>
                </li>
                <li>
                <a href="<?=base_url()?>
                      SP_controller/logOut">
                <figure>
                        <img alt="Log␣out" src
                             ="<?php␣echo␣
                             base_url();?>
                             assets/images/
                             logout.png">
                        <figcaption align="
                             center">Log out</
                             figcaption>
                </figure>
                </a>
                </li>
        </ul>
        <hr style="color:␣black;␣background-
             color:␣black;␣height:␣1px;" />
                <div id="edit" name="edit"
                      style="display:␣none;">
                        <h1 align="center">
                             Edit Profile</h1>
                        <form
                                name="
                                     editProfile
                                     "
                                method="POST"
                                class="form-
                                     signin"
                                action="<?php␣
                                     echo␣
                                     base_url
                                     ();?>
                                     SP_controller
                                     /
                                     updateUser
                                     ">
                        <h3>Username</h3>
                        <input
                                type="text"
                                name="username
                                     "
                                class="form-
                                     control"
                                Value=<?=$this
                                     ->session
                                     ->
                                     userdata(
                                     'username
                                     ')?> />
                        <h3>Password</h3>
                        <input
                type="password
                     "
                name="password
                     "
                id="password"
                class="form−
                     control␣
                     input−
                     password"
                title="Don't␣
                     do␣
                     anything␣
                     if␣you␣
                     don't␣
                     want␣your
                     ␣password
                     ␣to␣
                     change."
                Value=<?=$this
                     −>session
                     −>
                     userdata(
                     'password
                     ')?> />
            <div>
                <input id="
                     checkbox−
                     unmask"
                     class="
                     checkbox−
                     unmask"
                     type="
                     checkbox"
                      />
                <label for="
                     checkbox−
                     unmask"
                     class="
                     label−
                     unmask">
                     Show my
                     password
                     </label>
            </div>
            <h3>First Name</h3>
            <input
                type="text"
                name="
                     firstName
                     "
                class="form−
                     control"
                Value=<?=$this
                     −>session
                     −>
                     userdata(
                     '
                     firstName
                     ')?> />
            <h3>Last Name</h3>
            <input
                type="text"
                name="lastName
                     "
                class="form−
                     control"
                Value=<?=$this
                     −>session
                     −>
                     userdata(
                     'lastName
                     ')?> />
            <h3>Cellphone Number</
                 h3>
            <input
                type="text"
                name="cpNumber
                     "
                class="form−
                     control"
                Value=<?=$this
                     −>session
                     −>
                     userdata(
                     'cpNumber
                     ')?> />
            <br />
            <input
                class="btn␣btn
                     −lg␣btn−
                     primary␣
                     btn−block
                     "
                type="submit"
                value="Save">
            </form>
        </div>
    </div>
<?php
```

```
    for($i = 0;$i < 3;$i++){
    if($lvl[$i]−>lvl==($i+1)){
        if($lvl[$i]−>isTriggered==1){
?>

    <div class="modal−dialog">
    <div class="modal−content" id="myModal
       ">
    <div class="modal−body">
            <button type="button"
               class="close"
               data−dismiss="
               modal" data−show=
               "true">&times;</
               button>
        There is an ongoing fire!
    </div>
    <div class="modal−footer">
     <button type="button" class="btn␣btn
       −default" data−dismiss="modal">
       Close</button>
    </div>
  </div>
  </div>
        <script type='text/javascript'
          >

            function confirmation
              () {
                var answer =
                  confirm("
                  Have␣you␣
                  seen␣the␣
                  fire?")
                if (answer){
                    window
                      .
                      location

                      =

                      "
                      <?
                      php
                      ␣
                      echo
                      ␣
                      base␣url
                      ()
                      ;?>
                      SP␣controller
                      /
                      updateTrigger
                      /
                      "

                      +

                      <?=
                      $lvl
                      [
                      $i
                      ]−>
                      lvl
                      ?>;
                }
                else{
                    ;
                }
            }

            confirmation();
        </script>
<?php
            }
    }
    }
?>

<script>
setInterval(function(){
   location.reload();
 },60000);


var pw = $(".input−password"),
cb = $("#checkbox−unmask"),
pword = $("#password"),
mask = true;

cb.on("click", function() {
    if (mask === true) {
        mask = false;
        pw.attr("type", "text");
    } else {
        mask = true;
        pw.attr("type", "password");
        }
});

$(document).ready(function () {
    $('#e').click(function () {
        $('#report').hide();
        $('#edit').show();
    });
        $('#r').click(function () {
        $('#edit').hide();
        $('#report').show();
    });
        $('#h').click(function () {
        $('#report').hide();
        $('#edit').hide();
        $('#home').show();
    });
});
</script>
```

Listing 10: routes.php

```
<?php
defined('BASEPATH') OR exit('No␣direct␣script␣
    access␣allowed');

/*
| -------------------------------------------------------------------------
| URI ROUTING
| -------------------------------------------------------------------------
| This file lets you re-map URI requests to
    specific controller functions.
|
| Typically there is a one-to-one relationship
    between a URL string
| and its corresponding controller class/
    method. The segments in a
| URL normally follow this pattern:
|
|       example.com/class/method/id/
|
| In some instances, however, you may want to
    remap this relationship
| so that a different class/function is called
    than the one
| corresponding to the URL.
|
| Please see the user guide for complete
    details:
|
|       https://codeigniter.com/user_guide/
    general/routing.html
|
| -------------------------------------------------------------------------
| RESERVED ROUTES
| -------------------------------------------------------------------------
|
| There are three reserved routes:
|
|       $route['default_controller'] = '
    welcome';
|
| This route indicates which controller class
    should be loaded if the
| URI contains no data. In the above example,
    the "welcome" class
| would be loaded.
|
|       $route['404_override'] = 'errors/
    page_missing';
|
| This route will tell the Router which
    controller/method to use if those
| provided in the URL cannot be matched to a
    valid route.
|
|       $route['translate_uri_dashes'] = FALSE
    ;
|
| This is not exactly a route, but allows you
    to automatically route
| controller and method names that contain
    dashes. '-' isn't a valid
| class or method name character, so it
    requires translation.
| When you set this option to TRUE, it will
    replace ALL dashes in the
| controller and method URI segments.
|
| Examples:     my-controller/index     ->
    my_controller/index
|               my-controller/my-method ->
    my_controller/my_method
*/
$route['default_controller'] = 'SP_controller'
    ;
$route['404_override'] = '';
$route['translate_uri_dashes'] = FALSE;

$route['login'] = 'SP_controller/login';
```

XI. Acknowledgement

Makailang beses ko nang inisip na hindi ko kayang tapusin itong system ko on time. Kaya lang nung nagsimula na akong mag-puyat kaka-code, sinabi ko na sa sarili ko na ipagpapatuloy ko na itong system na 'to kahit wala munang docu or kung anong functionalities man lang ang matapos ko. Wala talaga akong kasiguraduhan kung matatapos ko ba talaga siya or kung ilang percent na ako. Umabot pa nga ung pagka-desperada kong matapos tong system na 'to to the point na kinonsider kong mag-check-in sa Sogo Hotel kasi walang internet sa bahay. At kulang na lang tuluyan na akong mag-dorm dun kina Cristine kasi gabi-gabi na akong andun.

Pero thank GOD natapos na rin!!! Kaya magsisimula na akong magpasalamat sa mga taong sobrang supportive sakin.

Unang una na si MAMA. Ma thank you po sa lahat ng suportang binigay niyo sakin 'di lang para sa SP na 'to kundi sa lahat lahat na. Sorry kung palagi na lang kayong nahihirapan sakin but I promise you, makakahinga ka na rin po ng maluwang dahil gagraduate na meeeeee.

Next si PAPA. Sobrang thank you po sa lahat ng suportang binigay niyo sakin. Kahit na araw-araw kong pinapasakit ulo niyo kakagising sakin para lang hindi ako malate. Thank you po talaga. Pati na rin sa tatlo kong kapatid na tiniis ung pagiging masungit ko sa bahay dahil kulang ako sa tulog. Maraming salamat mga kapatid.

Syempre hindi pwedeng hindi ko mabanggit ung Block12 Hotties. Sa totoo lang, naiirita na ko kapag sinasabihan niyo akong gawin ko na ung SP ko pero dahil sa walang sawa niyong pag-remind sakin kaya na-push akong gumawa rin. Nakaka-pressure din kasi halos lahat kayo nakapag-defense na kaya pinush ko talaga na makapag-defense din ako para sa Graduation Squad goals ang peg natin. Oh diva!!!

Syempre nasa listahan rin ng mga taong pasasalamatan ko ang adviser ko na si Sir Marvin John Ignacio. Sir Thank you sa maganda niyong strategy nung defense at nung proposal. Maraming salamat po sa mga advise niyo sakin. Dahil po dun

kaya ko nagawa kung ano man ang nagawa ko. Thank you rin po sa pagpupush niyo sakin na maka-graduate. Thank you po kasi pinag-defense niyo ko. Thank you po nang sobra.

Sunod ka na Cristine. Thank you kasi hinayaan mo kong mag-overnight ng maraming gabi sa dorm niyo. Thank you kasi sinabayan niyo (kayo ni Rain) akong gumawa kaya nakapag-concentrate ako sa SP ko. Dahil sa mga overnight na yun kaya ko natapos itong system na ito. Pasabi na rin kay Kuya dun na Thank you ha.

Special mention kay Rachelle Aquino at Isabella Inosantos. Ang dalawa kong ka-group sa Machine Problem sa CS 138. Sorry kasi halos wala akong ginawa para sa MP natin. Ung tanging edits ko nga dun natanggal din pero oks lang. Thank you kasi hinayaan niyo lang akong gumawa ng SP ko habang nage-MP kayo. I will pay my debts just not right now kasi wala pa akong trabaho sa ngayon. Libre ko kayo pag may pera na ko he he he.

Isa pa sa mga nakatulong sakin para matapos ito ay si Kopiko 78 at si Paracetamol. Thank you mga Brad. Kung di dahil sa inyo baka tulog na lang ako everyday at baka sumabog na sa sakit ung ulo ko.

At sa iba pa na tumulong sakin pero hindi ko naisama sa Acknowledgement page ng docu ko kasi minamadali ko na ung document ng SP ko, THANK YOU.