



**TRƯỜNG ĐẠI HỌC FPT**



# Home Security System

*Capstone Project*

# THE PEOPLE



**Phan Duy Hung**  
Supervisor



**Truong Van Cuong**  
Team Leader



**Dinh Tru Ngoc Diep**  
Developer



**Vo Tuan Hung**  
Developer



**Dinh Thanh Dung**  
Tester

# CONTENTS

1

## Introduction

Details about project Home Security System

2

## System Requirements

Comprehensive description of requirements

3

## System Design

How to develop ideas

4

## Future Works

Discussion about the future of HSS

5

## Demo

Run demo prototype of Home Security System

6

## Question & Answer

Q&A between team and committee

# INTRODUCTION

---

- ❑ BACKGROUND AND IDEAS

- ❑ EXISTING SYSTEMS

- ❑ THE SCOPE

- ❑ TOOLS AND SOFTWARE

A close-up, slightly blurred photograph of a Raspberry Pi single-board computer. The green PCB is populated with various components, including a silver USB-A to micro-USB adapter, a black micro-USB cable, a yellow Ethernet cable, and a black power cable. A small, square, silver component is visible on the left. The Raspberry Pi logo is printed on the board. The image is overlaid with a semi-transparent dark blue filter.

# BACKGROUND

# BACKGROUND AND IDEAS

- **The term “Internet of Things” (IoT)**
  - System of interrelated computing devices, mechanical, digital machines and objects
- **IoT Requirement**
  - Transfer data over a network without human-to-human or human-to-computer interaction
- **Types of IoT communication**
  - Device-to-Device Communication
  - Device-to-Cloud Communication



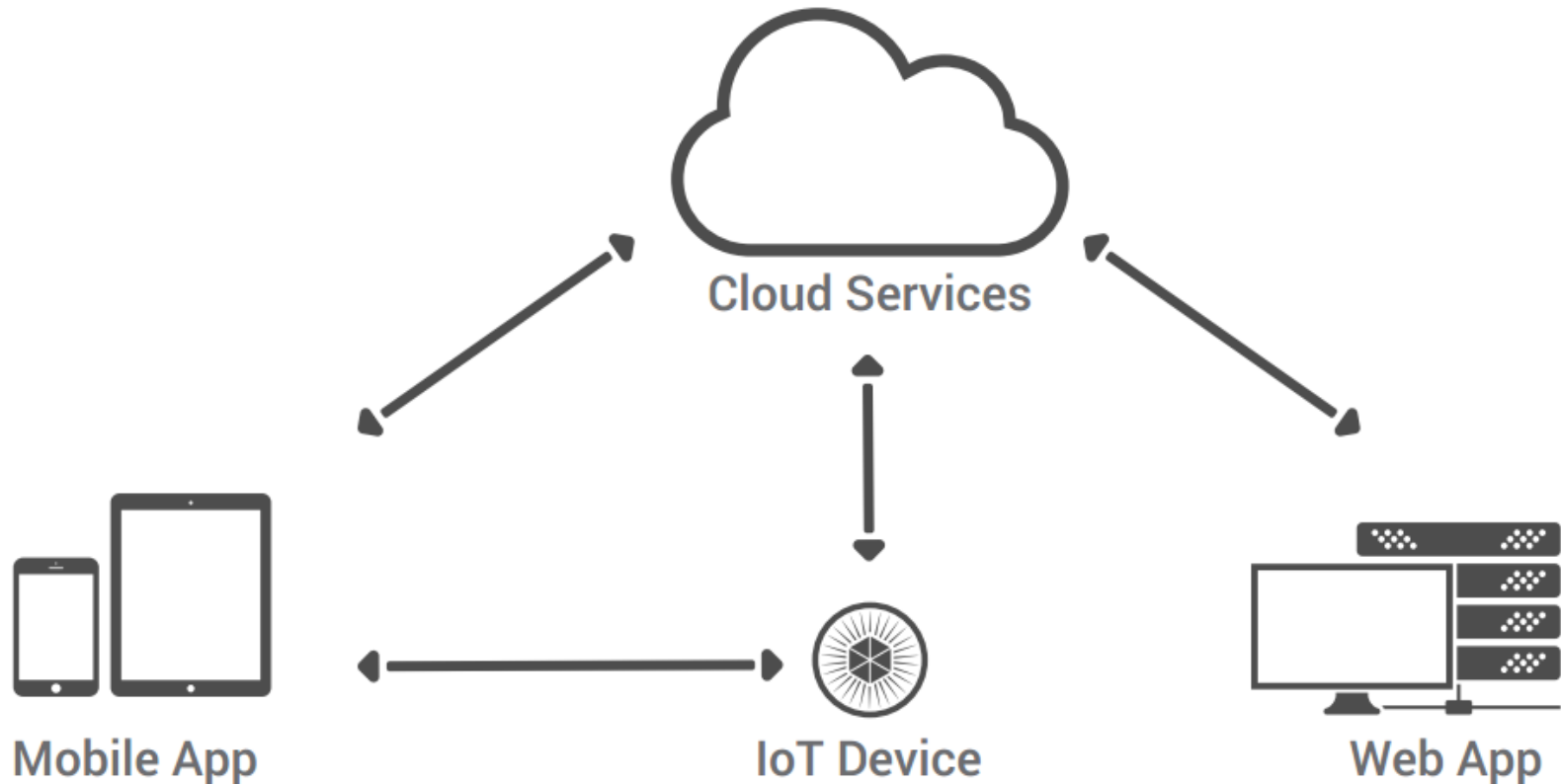
# BACKGROUND AND IDEAS (CONT.)

## ❑ Device-to-Device communication



# BACKGROUND AND IDEAS (CONT.)

## ❑ Device-to-Cloud communication



*Communication Model of HSS Project*



A close-up photograph of a Raspberry Pi 4 single-board computer. The board is populated with several components: a silver M.2 NVMe SSD is connected to the M.2 slot; a black USB-A to USB-C adapter is plugged into the USB-A port; a black USB-C to HDMI adapter is plugged into the USB-C port; and a black USB-A to Ethernet adapter is plugged into the Ethernet port. The board is also connected to a power source via a black USB-C to USB-A cable. The text "EXISTING SYSTEMS" is overlaid in white, bold, sans-serif font across the center of the image.

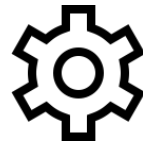
# EXISTING SYSTEMS

# EXISTING SYSTEMS

## ❑ IDEAL Security System



\$250



Working

Connect to telephone landline  
Alarm by calling directly to phone  
Lose memory when power failure  
2 door sensors, 1 motion sensor  
remote control, telephone dialer

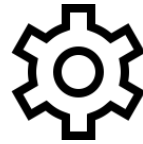
# EXISTING SYSTEMS (CONT.)

## ❑ SimpliSafe House Security System



\$260

\$24.99/month for smartphone app



Working

Very simple to use

Plug and Play

Includes door sensors, motion detectors, CO detector, panic button for emergencies etc...

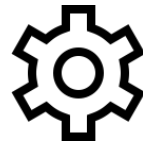
# EXISTING SYSTEMS (CONT.)

## ❑ iSmartAlarm Premium Package



\$199

\$349 for including iCamera



Working

Smartphone application

2 door sensors, 1 motion sensor

2 remote tags

Very easy to integrate with  
additional sensors

## IDEAS AND OBJECTIVES

**“91% of people keep  
their smartphone within  
3 feet 24 hours a day”**

*- Morgan Stanley -*

# THE SCOPE



## **Internet**

WLAN or Ethernet



## **Real-time Handling**

Send and receive data  
in real-time



## **Sensor**

Motion and Door detection  
A push button as doorbell



## **Audio Quality**

64kbps



## **Connection**

HSS-Board and HSS-App  
in a local network



## **Image Quality**

JPEG 640x480 pixel

# TOOLS AND SOFTWARES

## ☐ Programming Language



python<sup>TM</sup>



## ☐ Tools and Software



Qt



Github



Google Drive



Astah



Arduino IDE



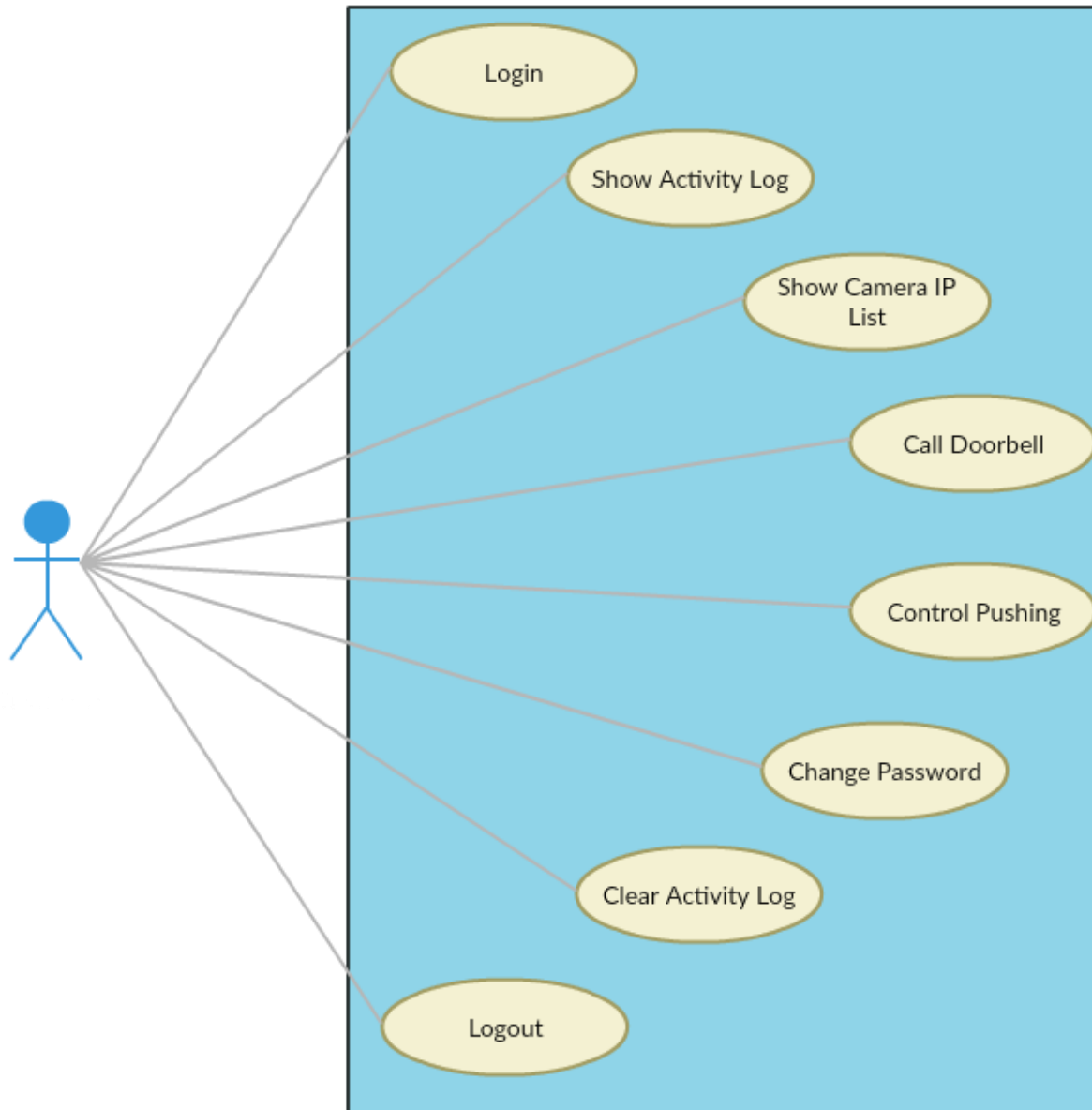
# SYSTEM REQUIREMENTS

---

- ❑ FUNCTIONAL REQUIREMENTS

- ❑ NON-FUNCTIONAL REQUIREMENTS

# FUNCTIONAL REQUIREMENTS



# NON-FUNCTIONAL REQUIREMENTS

- **Safety**

The supply voltage is totally safe with user (5V)

- **Reliability**

Availability

Low Failure Rate

- **Security**

Always require authenticating process

- **Usability**

Well-formed graphical UI

Informative error messages

# SYSTEM DESIGN

---

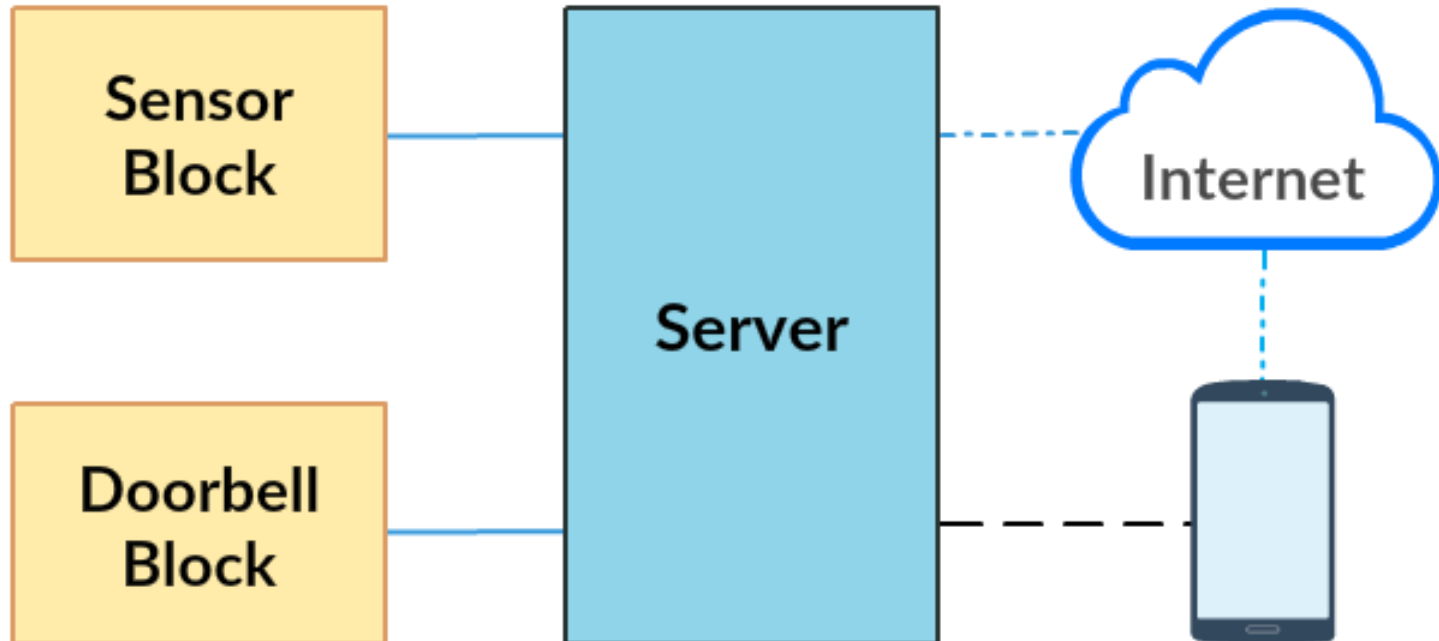
☐ SYSTEM ARCHITECTURE

☐ HARDWARE DESIGN

☐ SOFTWARE DESIGN

# SYSTEM ARCHITECTURE

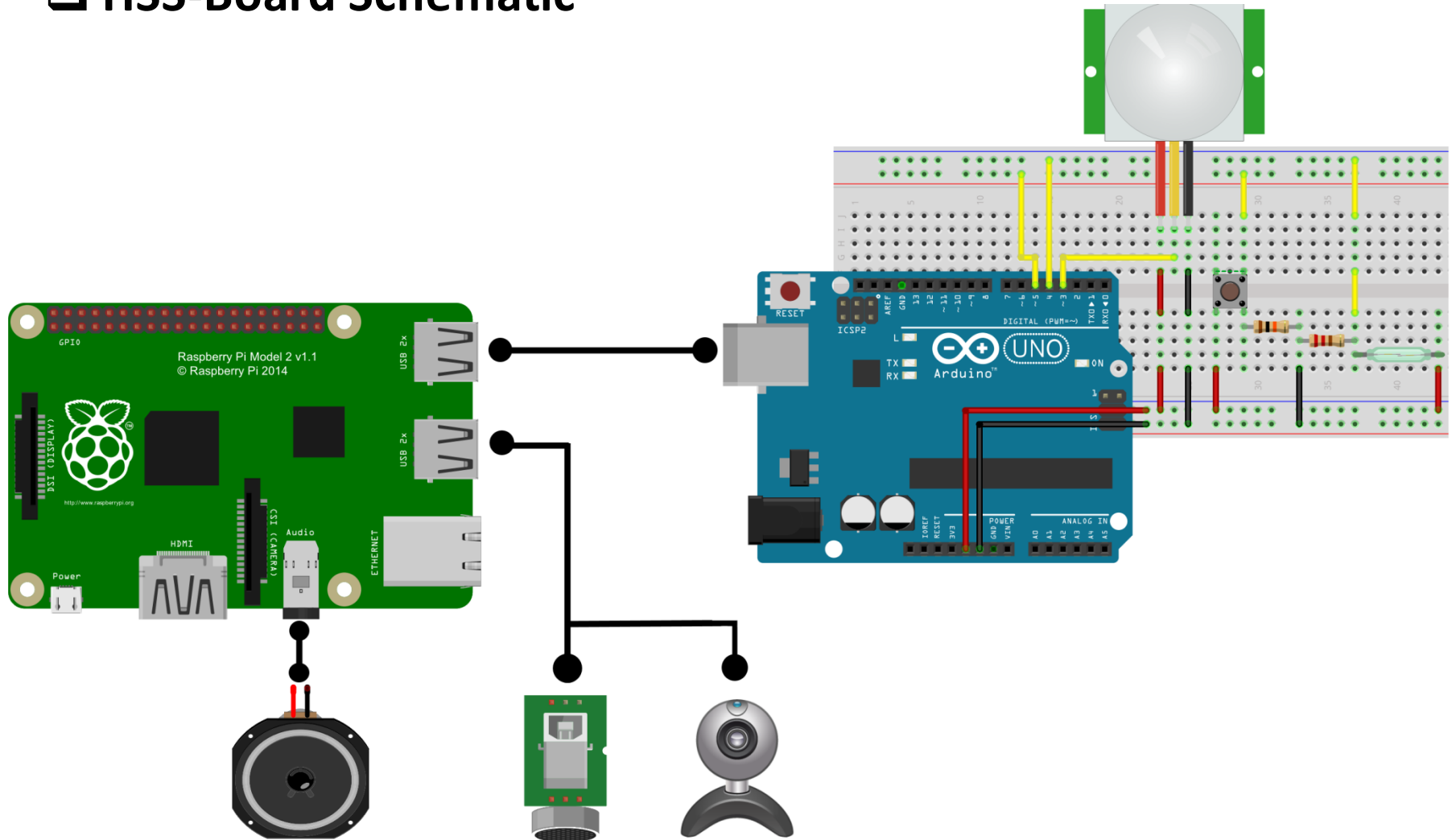
## □ Overview



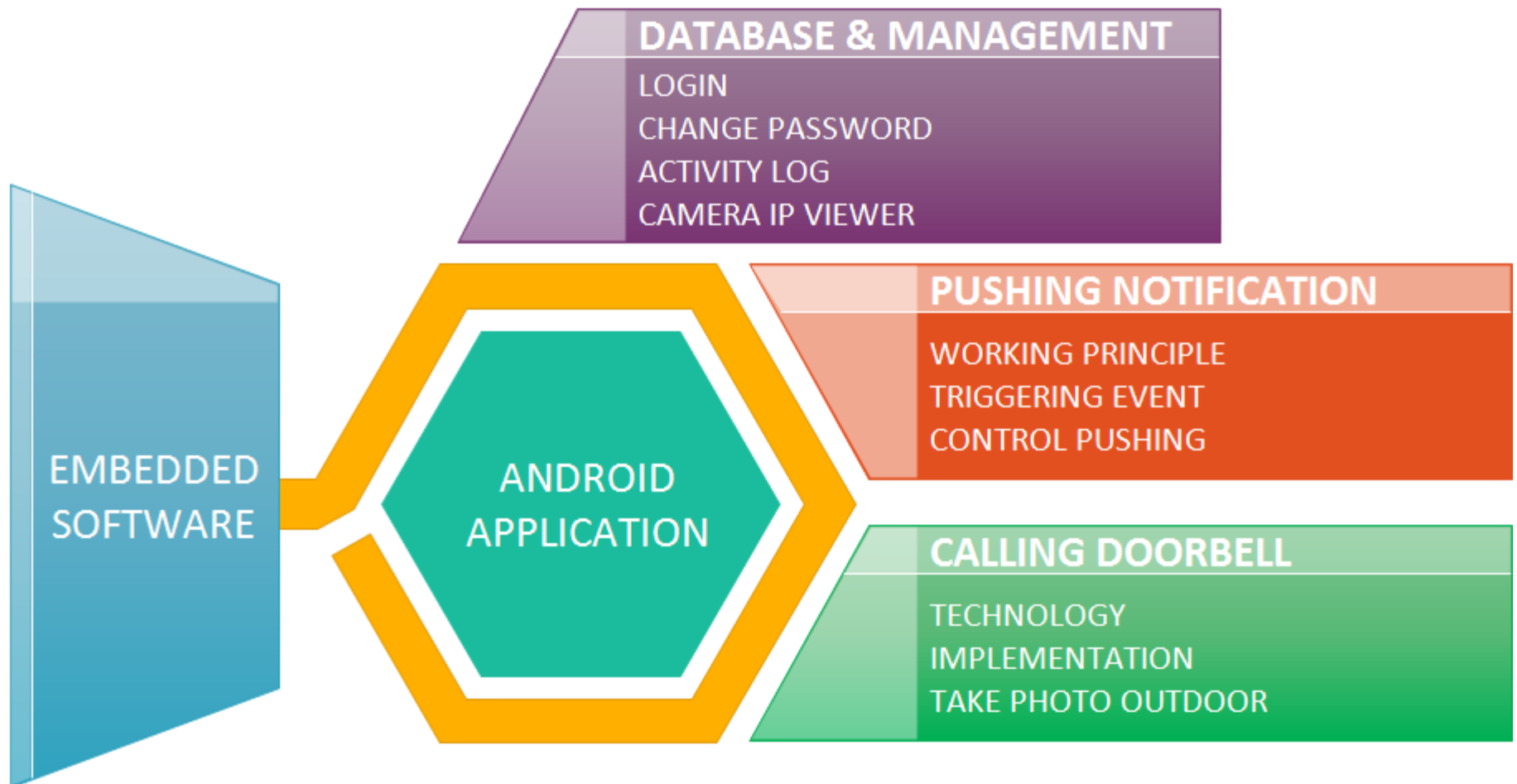
- **HSS-Board:** Devices in wired connection
- **HSS-App:** Android app and third-party services

# HARDWARE DESIGN

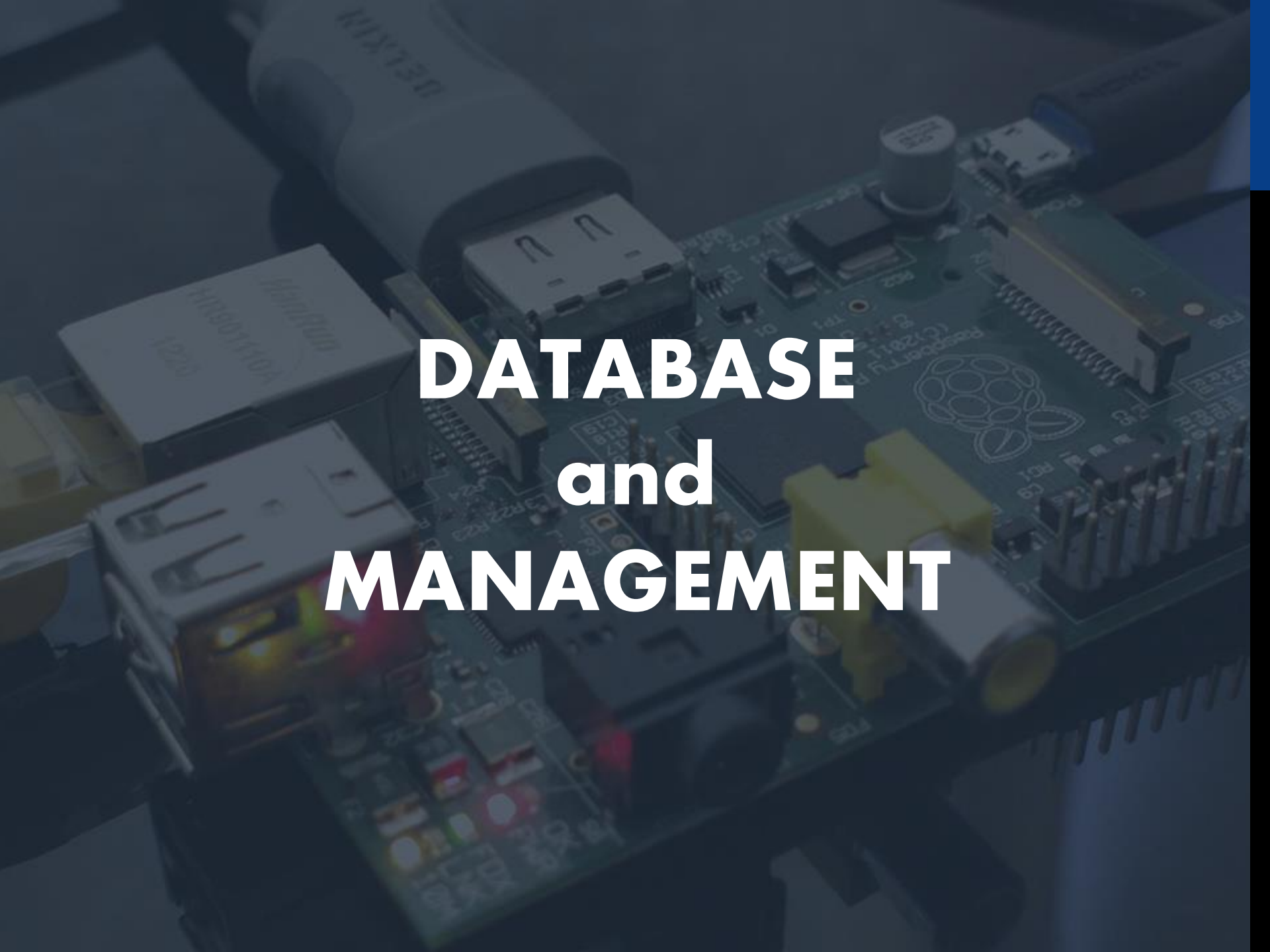
## ❑ HSS-Board Schematic



# SOFTWARE DESIGN





A close-up photograph of a Raspberry Pi single-board computer. The green PCB is populated with various components, including a silver USB-A to micro-USB adapter, a black micro-USB cable, a yellow Ethernet cable, and a black HDMI cable. The Raspberry Pi logo is visible on the board. The image is overlaid with a semi-transparent dark blue filter.

# **DATABASE and MANAGEMENT**

# SOFTWARE DESIGN

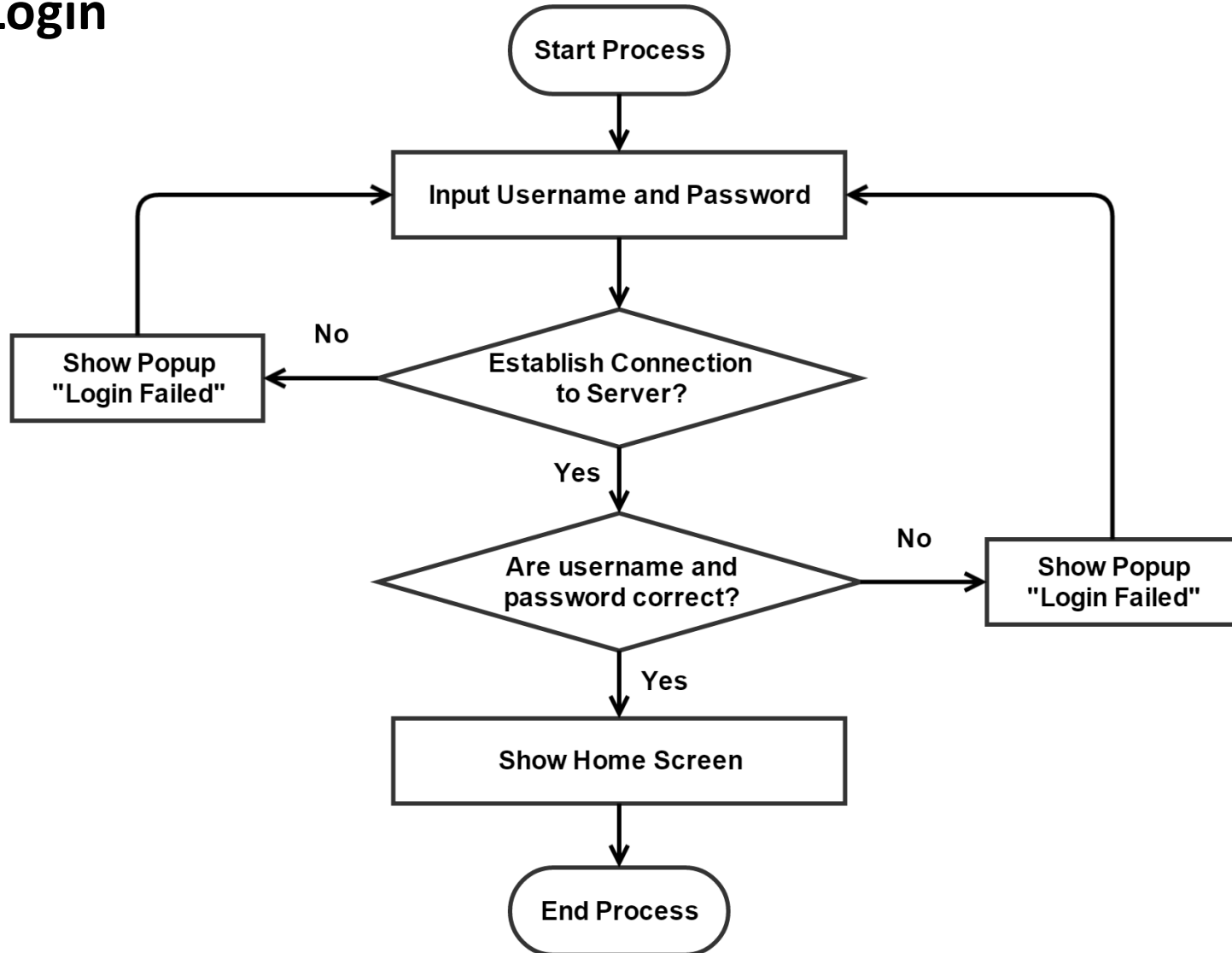
## ❑ Database



Name	Meaning
Username	Username for login
Password	Password for login
Time	Time of triggered event
Pushmessage	Message shown in Activity Log screen
CameraName	Name of Camera IP
Url	URL of Camera IP

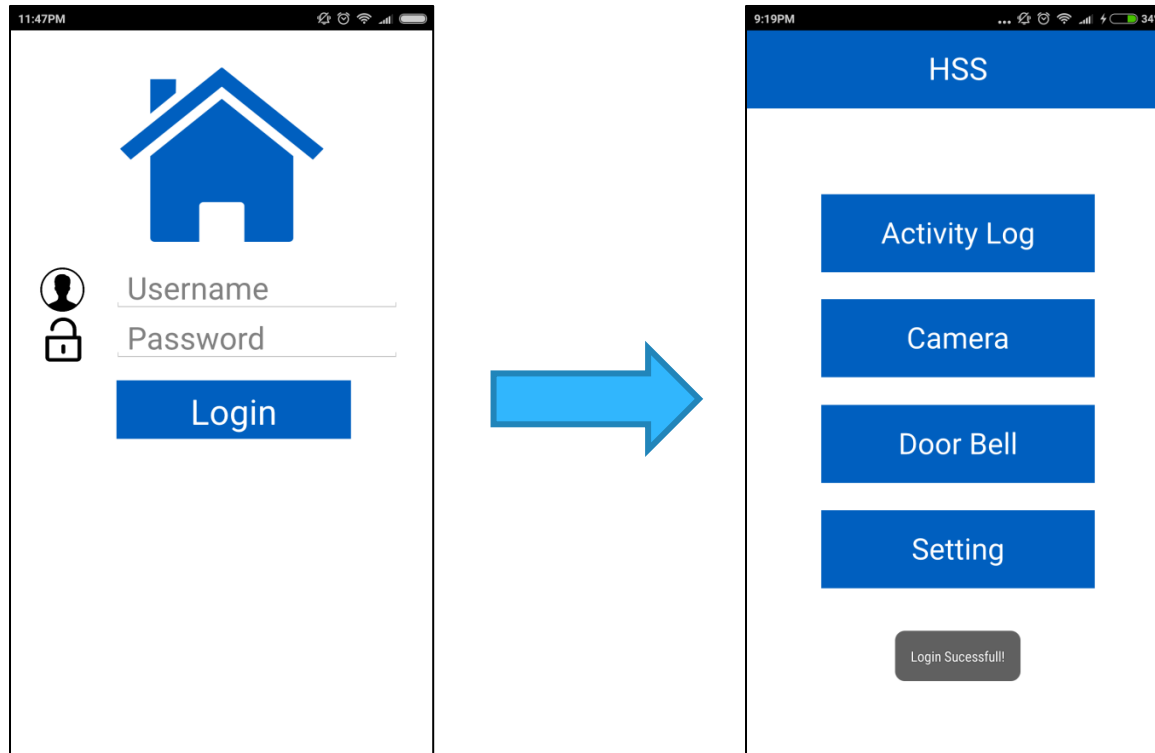
# SOFTWARE DESIGN (CONT.)

## ❑ Login



# SOFTWARE DESIGN (CONT.)

## ❑ Login



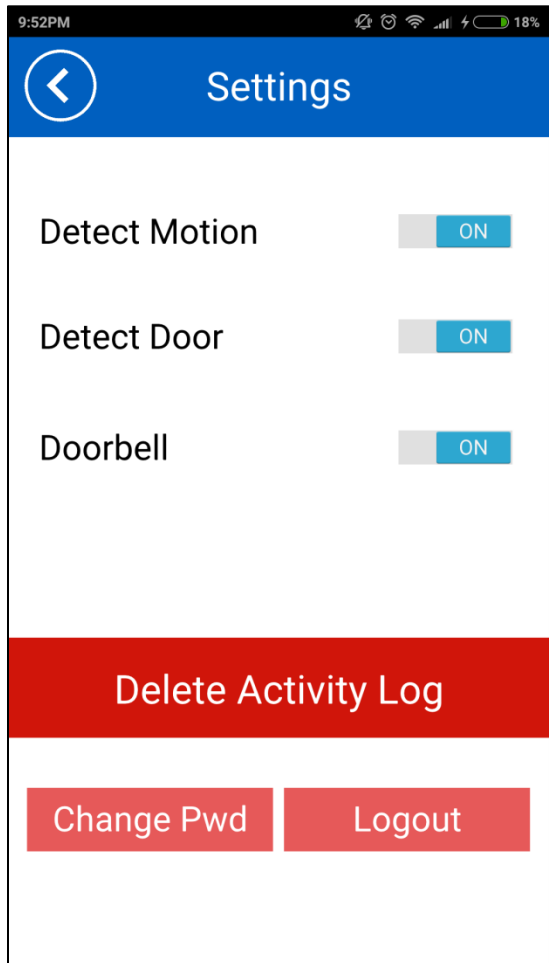
- **Popup on Screen**

Login Failed!

Login Sucessfull!

# SOFTWARE DESIGN (CONT.)

## ❑ Change Password



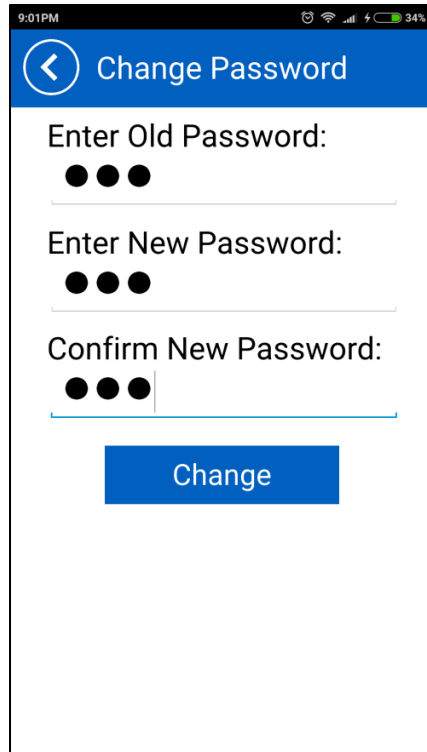
Change Pwd



Change Password Screen

# SOFTWARE DESIGN (CONT.)

## ❑ Change Password



9:01PM

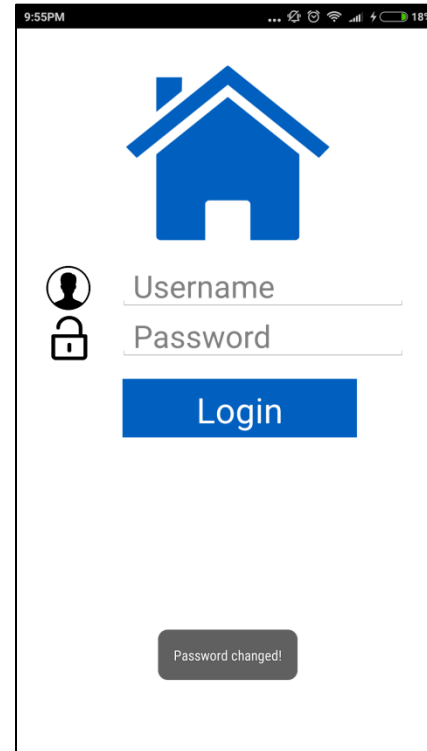
Change Password

Enter Old Password:  
● ● ●

Enter New Password:  
● ● ●

Confirm New Password:  
● ● ●

Change



9:55PM

Username

Password

Login

Password changed!

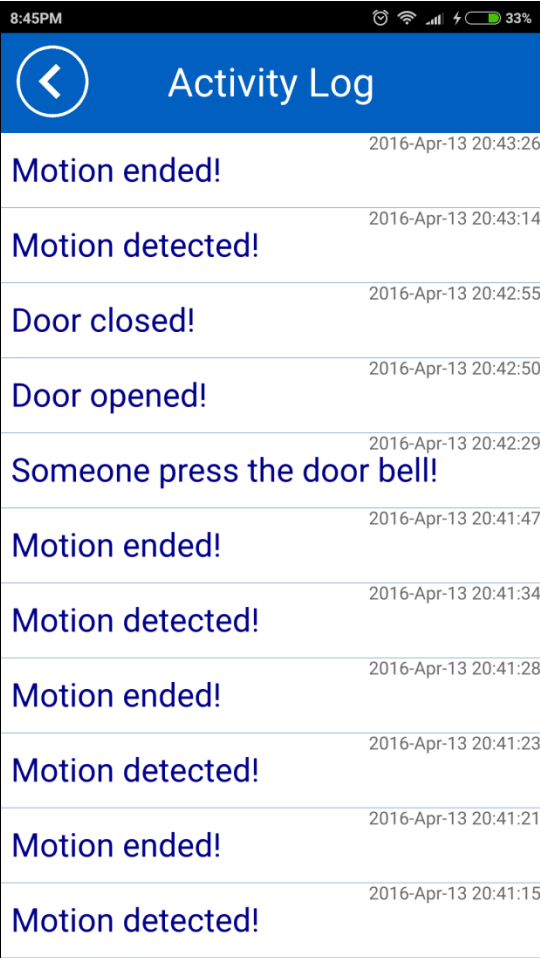
- **Popup on Screen**

Password unchanged!

Password changed!

# SOFTWARE DESIGN (CONT.)

## ❑ Show Activity Log

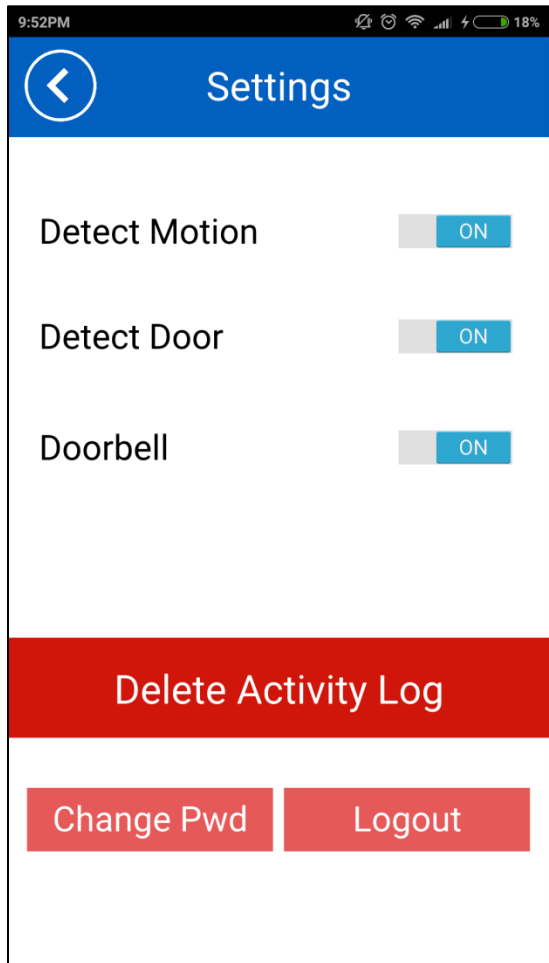


8:45PM	
33%	
<	Activity Log
Motion ended!	2016-Apr-13 20:43:26
Motion detected!	2016-Apr-13 20:43:14
Door closed!	2016-Apr-13 20:42:55
Door opened!	2016-Apr-13 20:42:50
Someone press the door bell!	2016-Apr-13 20:42:29
Motion ended!	2016-Apr-13 20:41:47
Motion detected!	2016-Apr-13 20:41:34
Motion ended!	2016-Apr-13 20:41:28
Motion detected!	2016-Apr-13 20:41:23
Motion ended!	2016-Apr-13 20:41:21
Motion detected!	2016-Apr-13 20:41:15



# SOFTWARE DESIGN (CONT.)

## ❑ Delete Activity Log



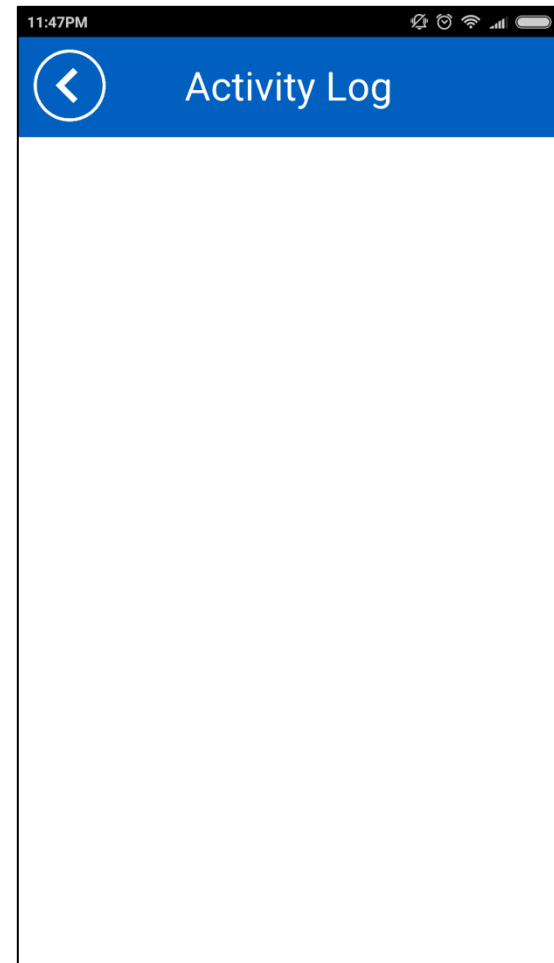
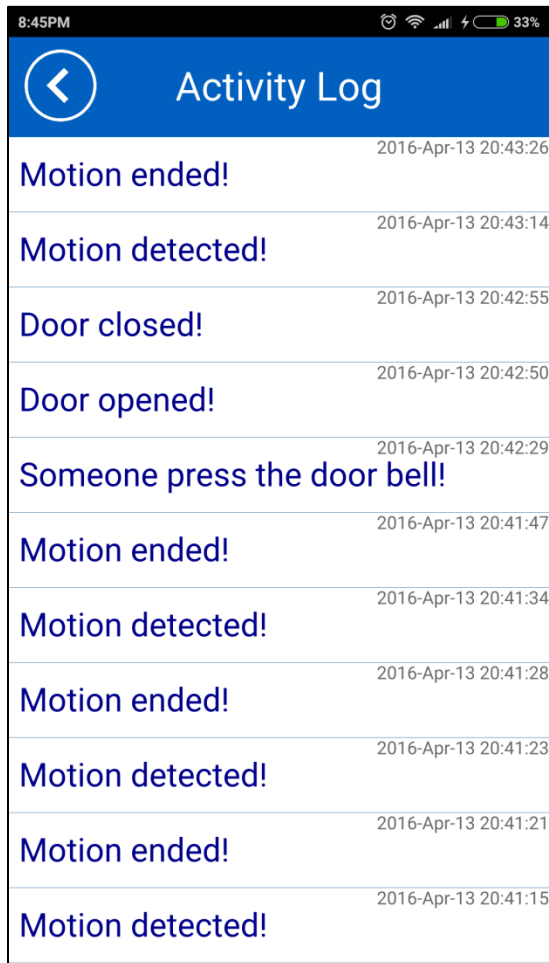
Delete Activity Log



Activity log deleted!

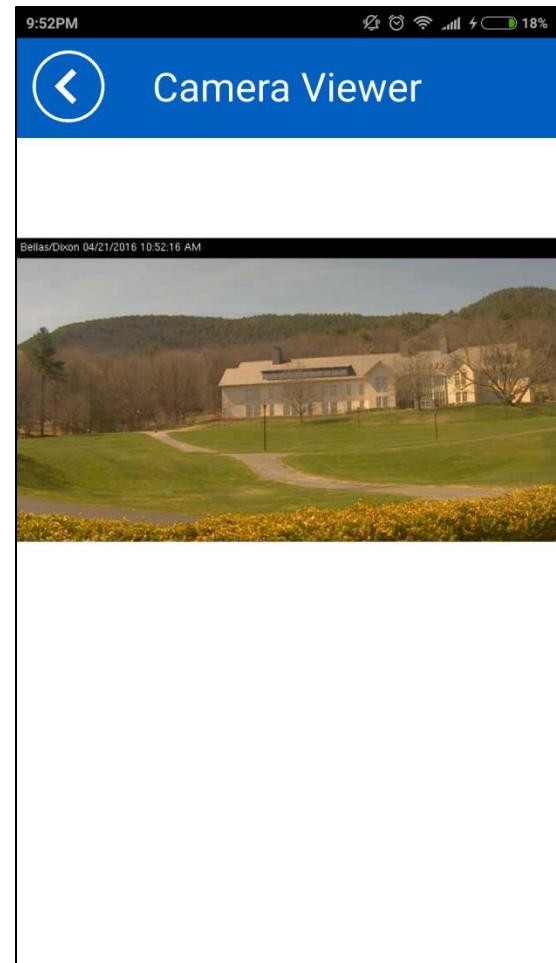
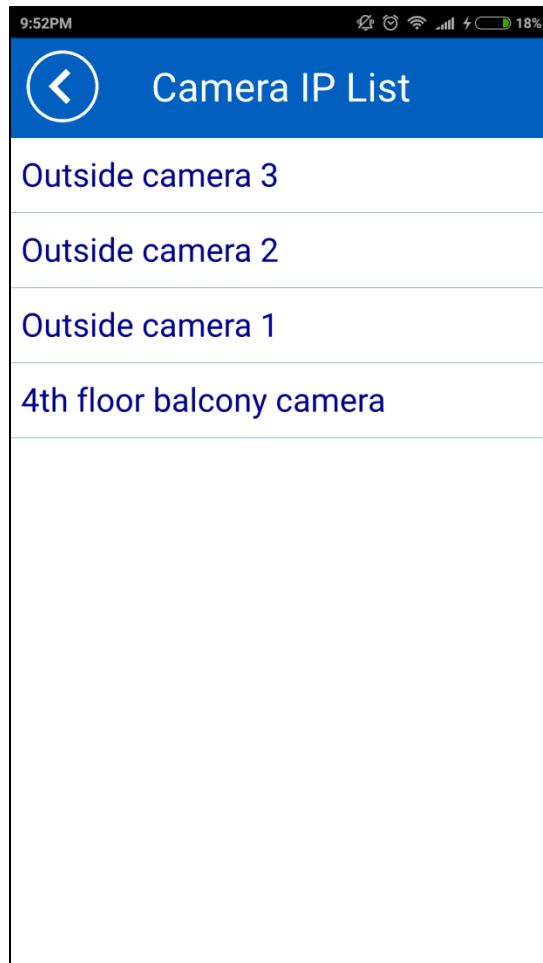
# SOFTWARE DESIGN (CONT.)

## ❑ Delete Activity Log



# SOFTWARE DESIGN (CONT.)

## ■ View Camera IP



A close-up photograph of a Raspberry Pi 4 Model B computer board. The board is green and populated with various components including a micro-SD card, a USB-C port, a USB-A port, a 3.5mm audio jack, and a 40-pin GPIO header. The text "PUSHING NOTIFICATION" is overlaid in large, white, bold, sans-serif capital letters across the center of the image. The background is slightly blurred, showing other electronic components and cables.

# **PUSHING NOTIFICATION**

# SOFTWARE DESIGN (CONT.)

## ❑ Pushing Notification

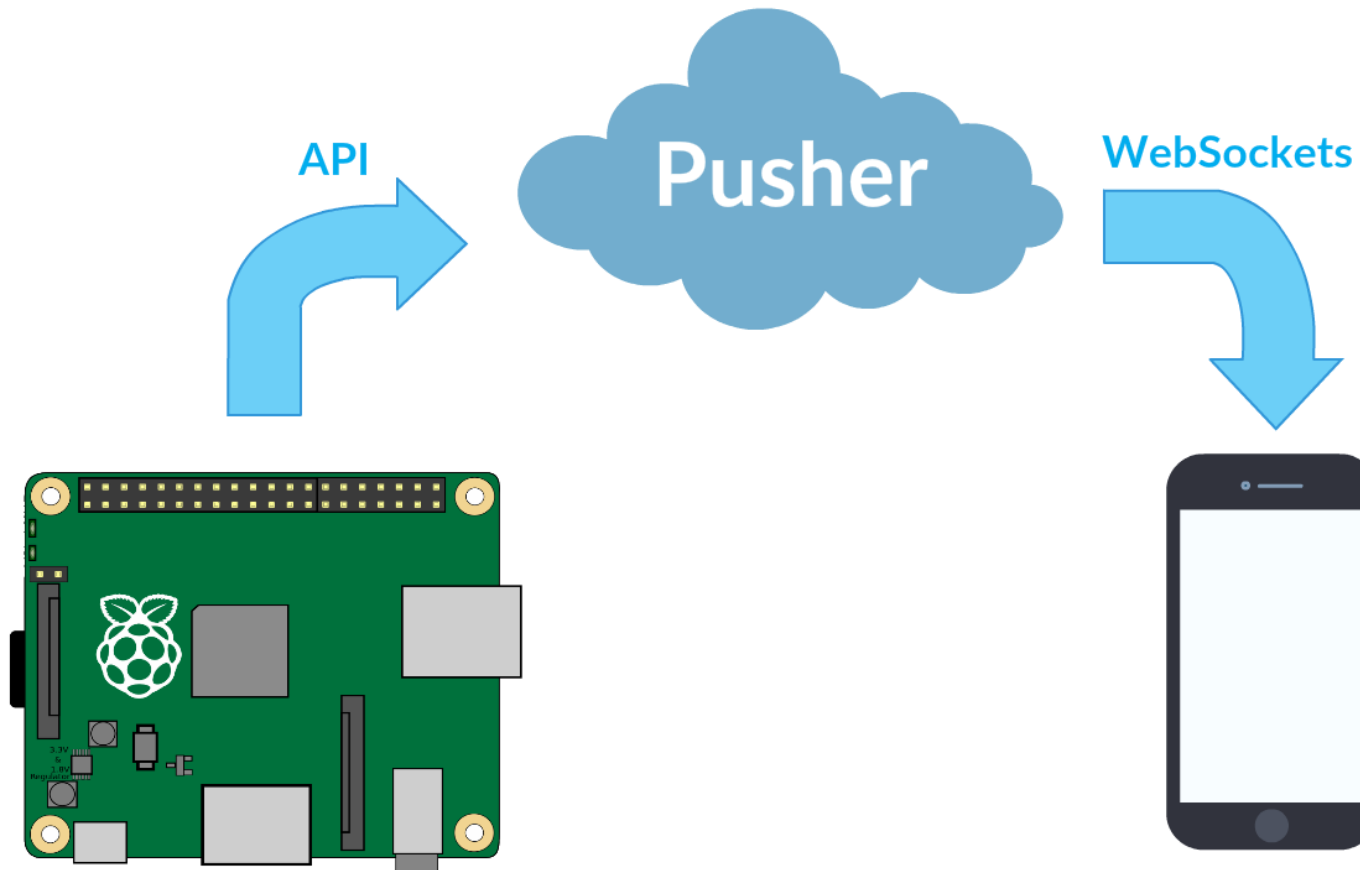
### ■ Pusher

- A service for message communication between web and mobile applications
- Deliver over 160 billion messages to more than 5.4 billion devices



# SOFTWARE DESIGN (CONT.)

## ❑ Pushing Notification



# SOFTWARE DESIGN (CONT.)

## ❑ Pushing Notification

### ■ On Raspberry Pi

- Connect to Pusher server
- Send message to a channel

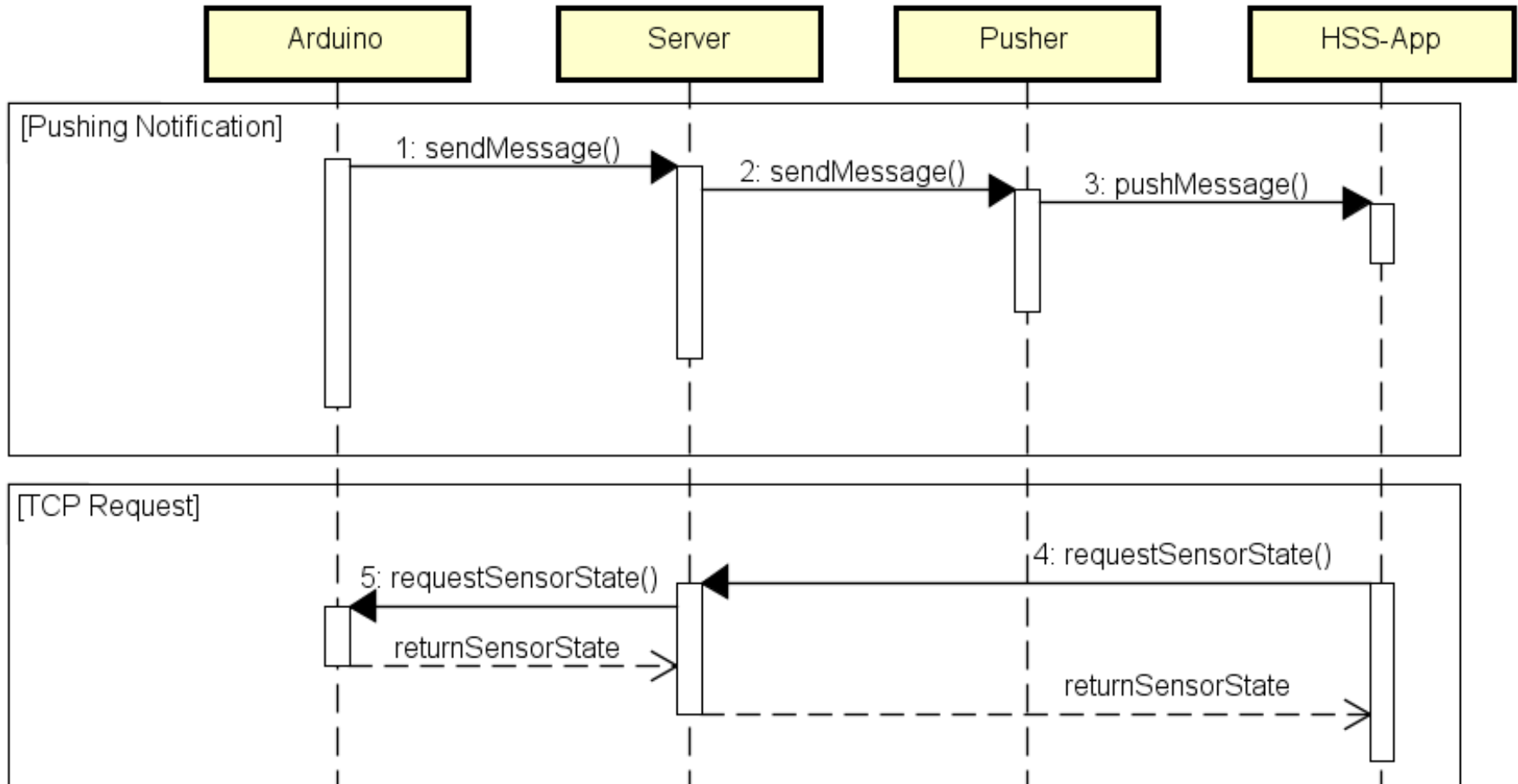
### ■ On HSS-App

- Connect to Pusher server
- Subscribe to Raspberry Pi channel
- Listen for message
- Show notification when message arrived



# SOFTWARE DESIGN (CONT.)

## ❑ Client – Server Communication



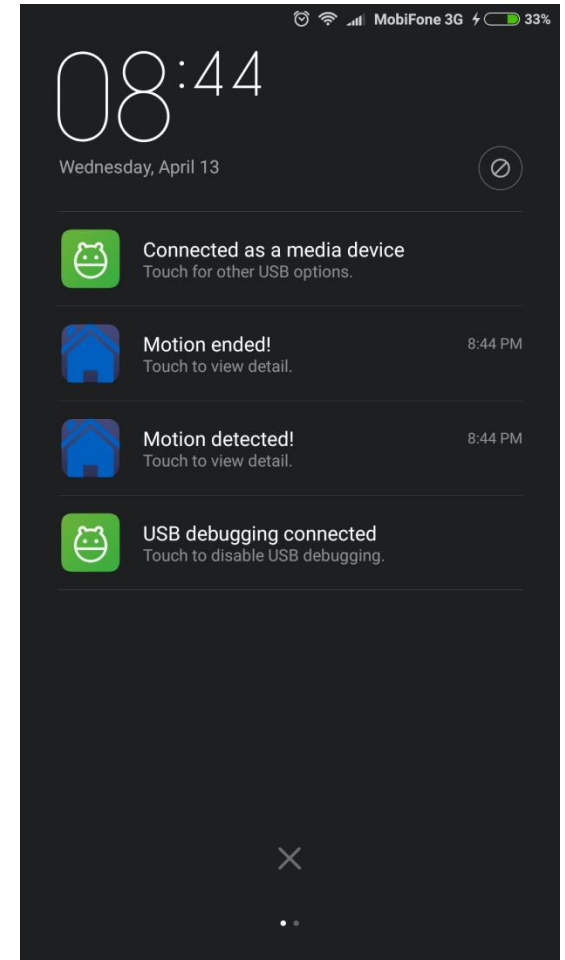
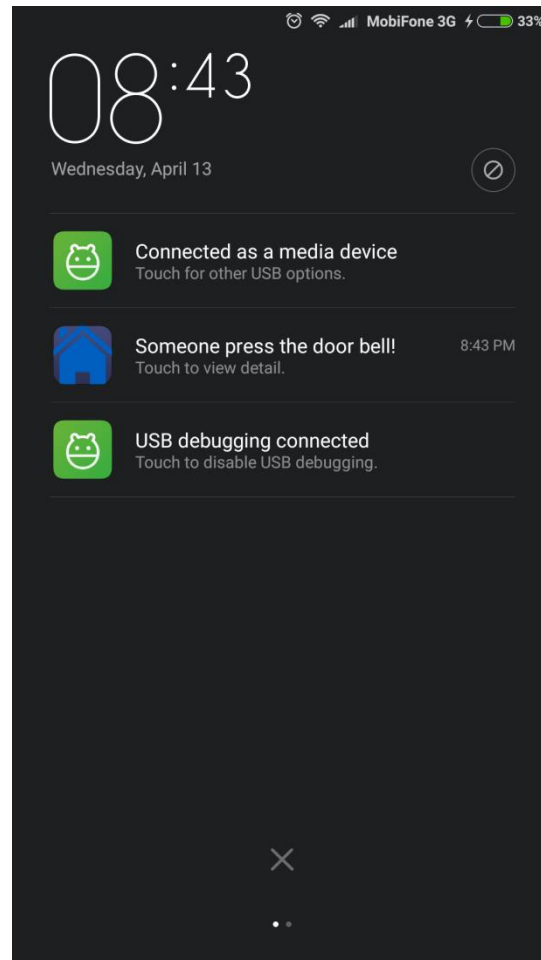
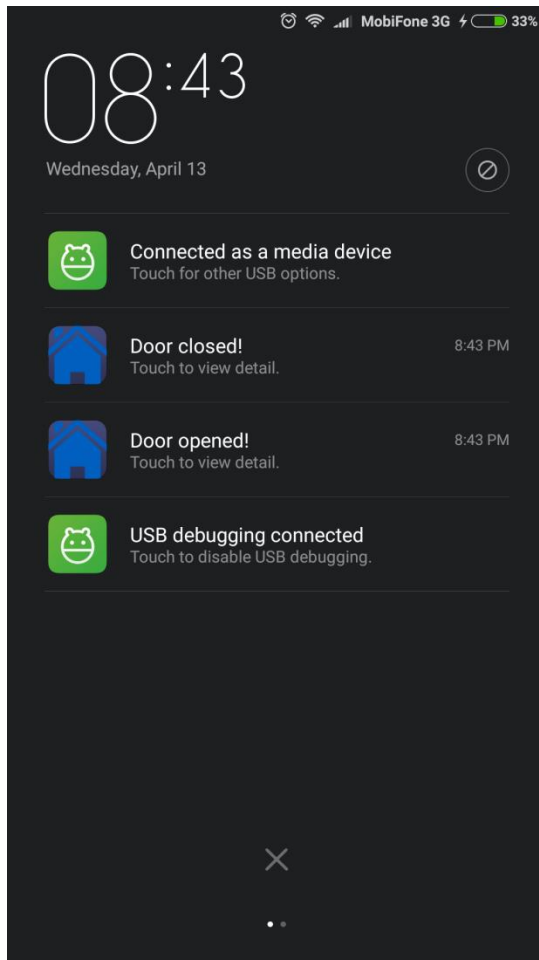
# SOFTWARE DESIGN (CONT.)

## ❑ Pushing Notification: Triggering Event

Sensor	Trigger	Action	Message
Button	Press or hold once	Push to HSS-App	Someone press the doorbell
Door	Change state from closing to open	Push to HSS-App	Door has been opened
	Change state from open to close	Push to HSS-App	Door has been closed
PIR	Detecting motion	Push to HSS-App once every 5s	Motion detected
	Motion ends	Push to HSS-App after 5s without any motion detected	Motion ended

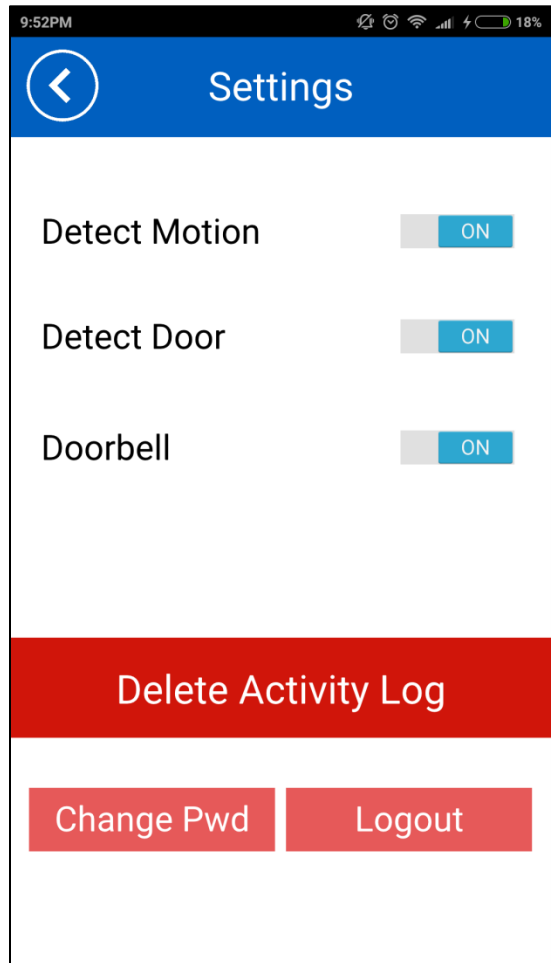
# SOFTWARE DESIGN (CONT.)

## ❑ Pushing Notification: Sample Screen



# SOFTWARE DESIGN (CONT.)

## ❑ Control Pushing Notification



Detect Motion

ON

Detect Door

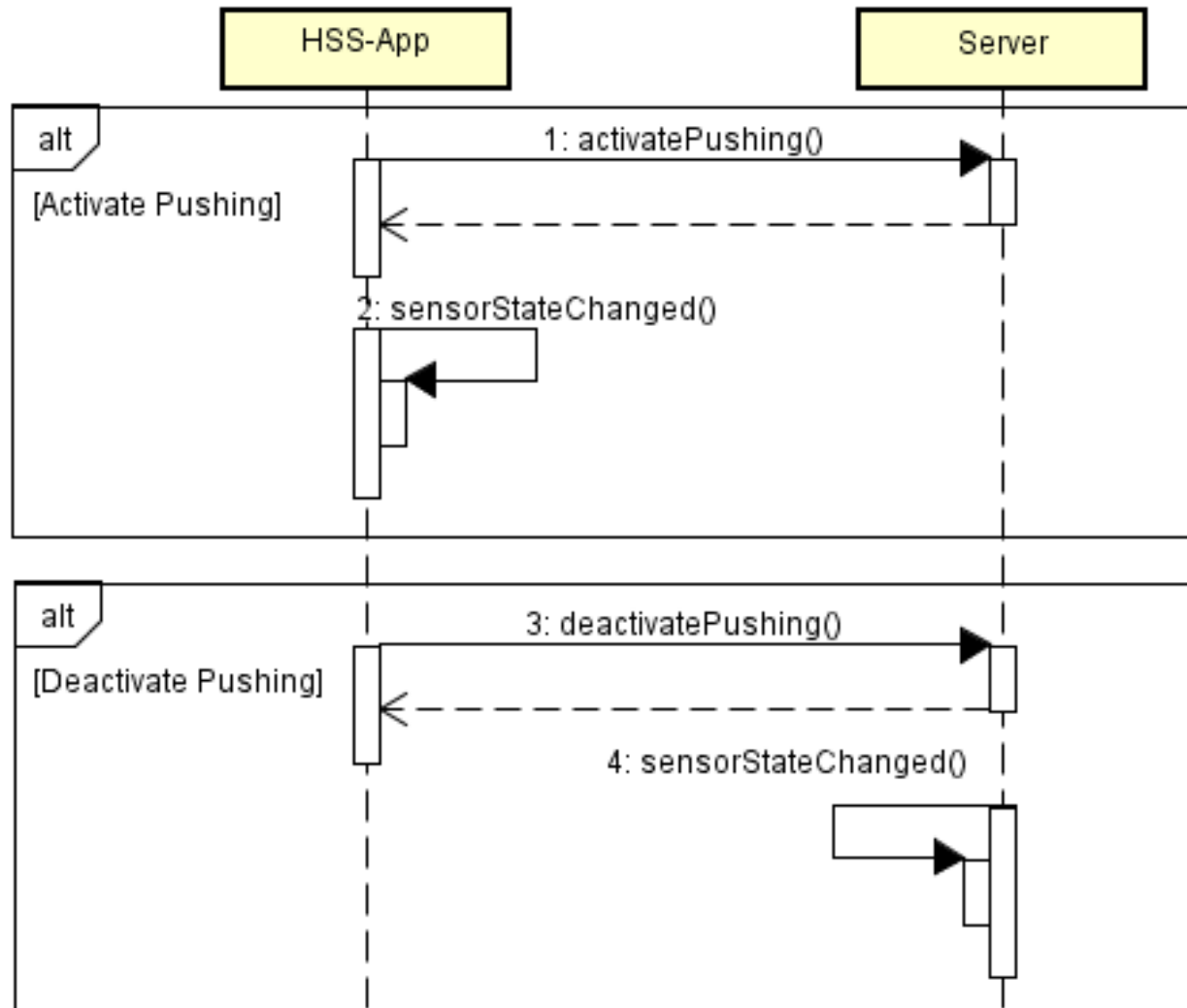
ON

Doorbell

ON

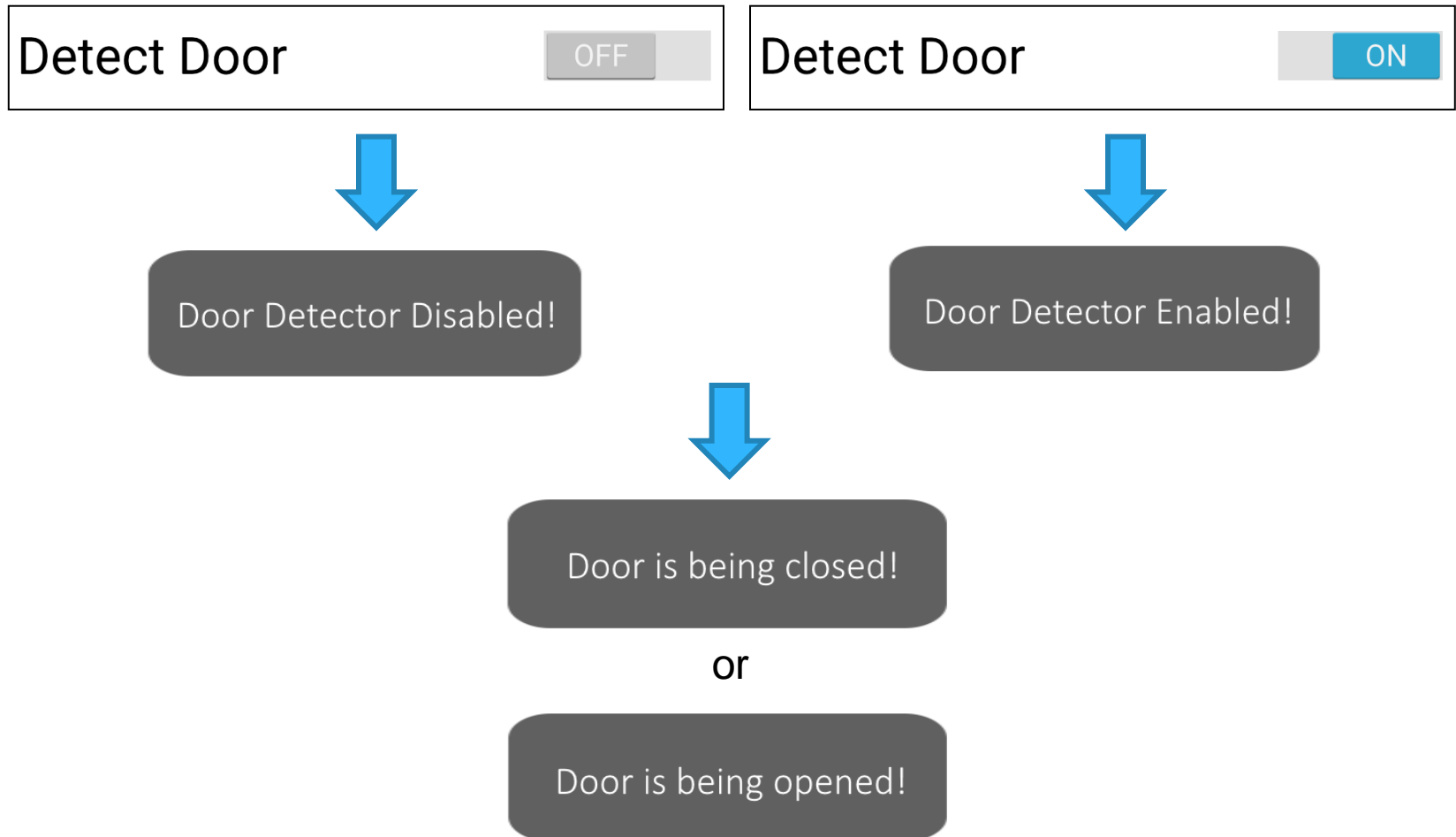
# SOFTWARE DESIGN (CONT.)

## ❑ Control Pushing Notification



# SOFTWARE DESIGN (CONT.)

## ❑ Check current door stte



A close-up photograph of a Raspberry Pi 4 Model B computer board. The board is populated with several peripherals: a silver M.2 NVMe SSD is connected to the M.2 slot; a black USB-A to USB-C adapter is plugged into a USB-A port; a black USB-C to Ethernet adapter is plugged into a USB-C port; and a black USB-A to FireWire adapter is plugged into another USB-A port. The Raspberry Pi logo is visible on the board. The image is overlaid with a semi-transparent dark blue filter.

# CALLING DOORBELL

# SOFTWARE DESIGN (CONT.)

## ❑ Calling Doorbell

### ▪ **Voice over IP Technology (VoIP)**

- Delivery of voice communications and multimedia sessions over IP network
- Providers usually offer lower rates than traditional phone

### ▪ **SIP (Session Initiation Protocol)**

Signaling and controlling communication sessions

### ▪ **STUN (Session Traversal Utilities for NAT)**

Assisting devices behind NAT with their packet routing

### ▪ **Linphone SIP Server**

Free SIP service based on Flexisip proxy server

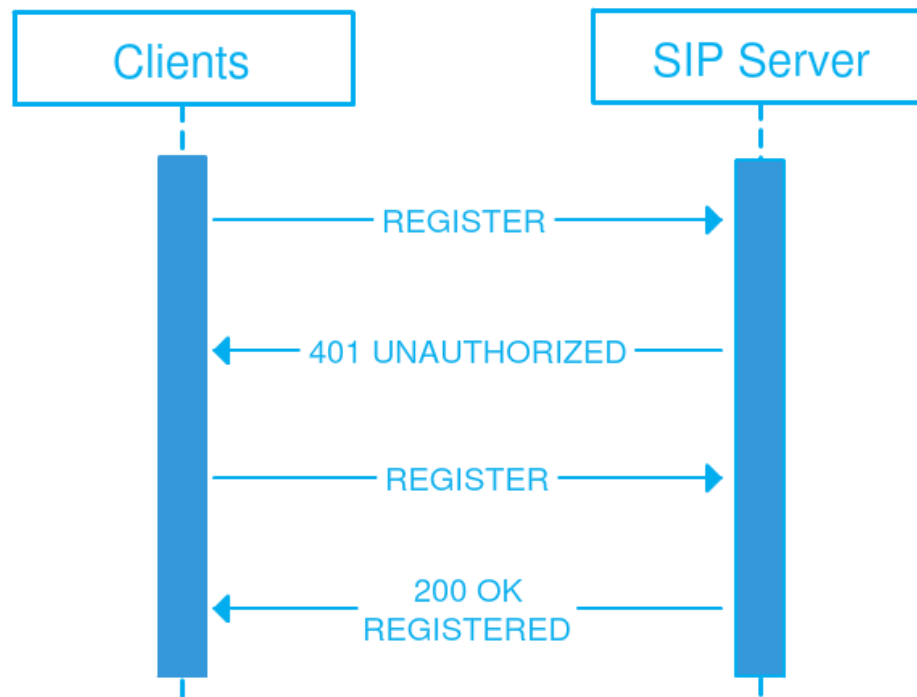


# SOFTWARE DESIGN (CONT.)

## ❑ Calling Doorbell

- **SIP Server in HSS**

Responsible for keep registering with clients in both 2 sides: Raspberry Pi and HSS-App; then routing the connection when initiating a VoIP call (Calling to Doorbell)



# SOFTWARE DESIGN (CONT.)

## ❑ Calling Doorbell

- On Raspberry Pi
  - Install Linphone module package for Python
  - Setting in code:

Parameter	Value
SIP Server	sip.linphone.org
STUN Server	stun.linphone.org
Auto Answer Call	Yes
Sound Input	ALSA: USB PnP Sound Device
Sound Output	Default

# SOFTWARE DESIGN (CONT.)

## ❑ Calling Doorbell

- On HSS-App, Linphone Library and API were used for developing VoIP feature

### LIBLINPHONE

Cross-platform SDK  
for SIP communication and media processing

#### Easy to use API to:

- place and receive calls
- manage proxies
- manage presence
- configure codecs

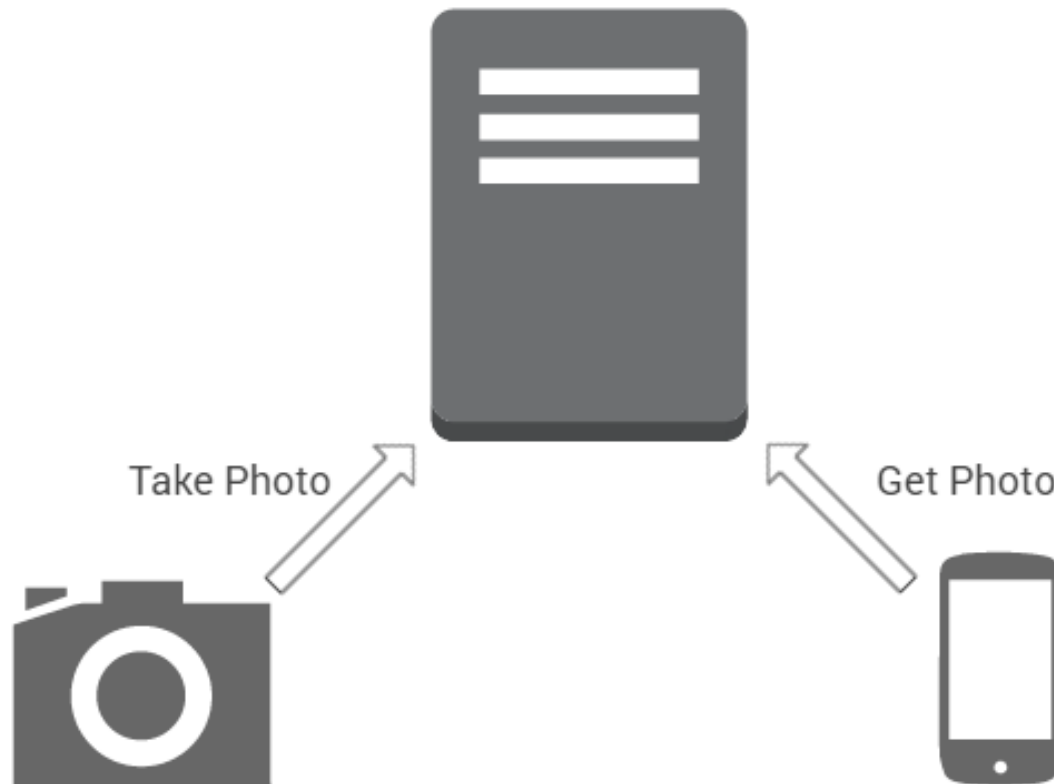
#### Available for:

- iOS
- Android
- Windows Phone 8
- BlackBerry 10
- Windows Desktop
- Mac OSX
- Linux

# SOFTWARE DESIGN (CONT.)

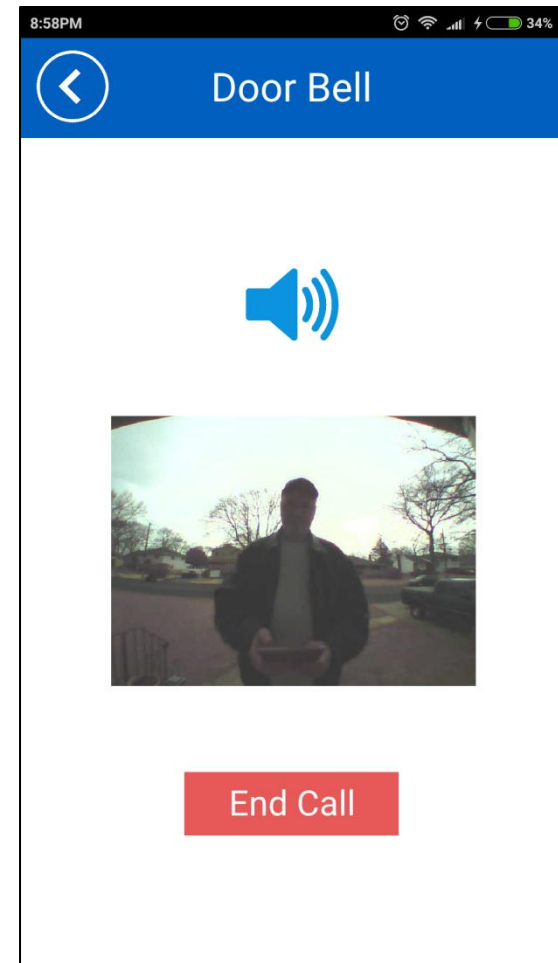
## ❑ Calling Doorbell

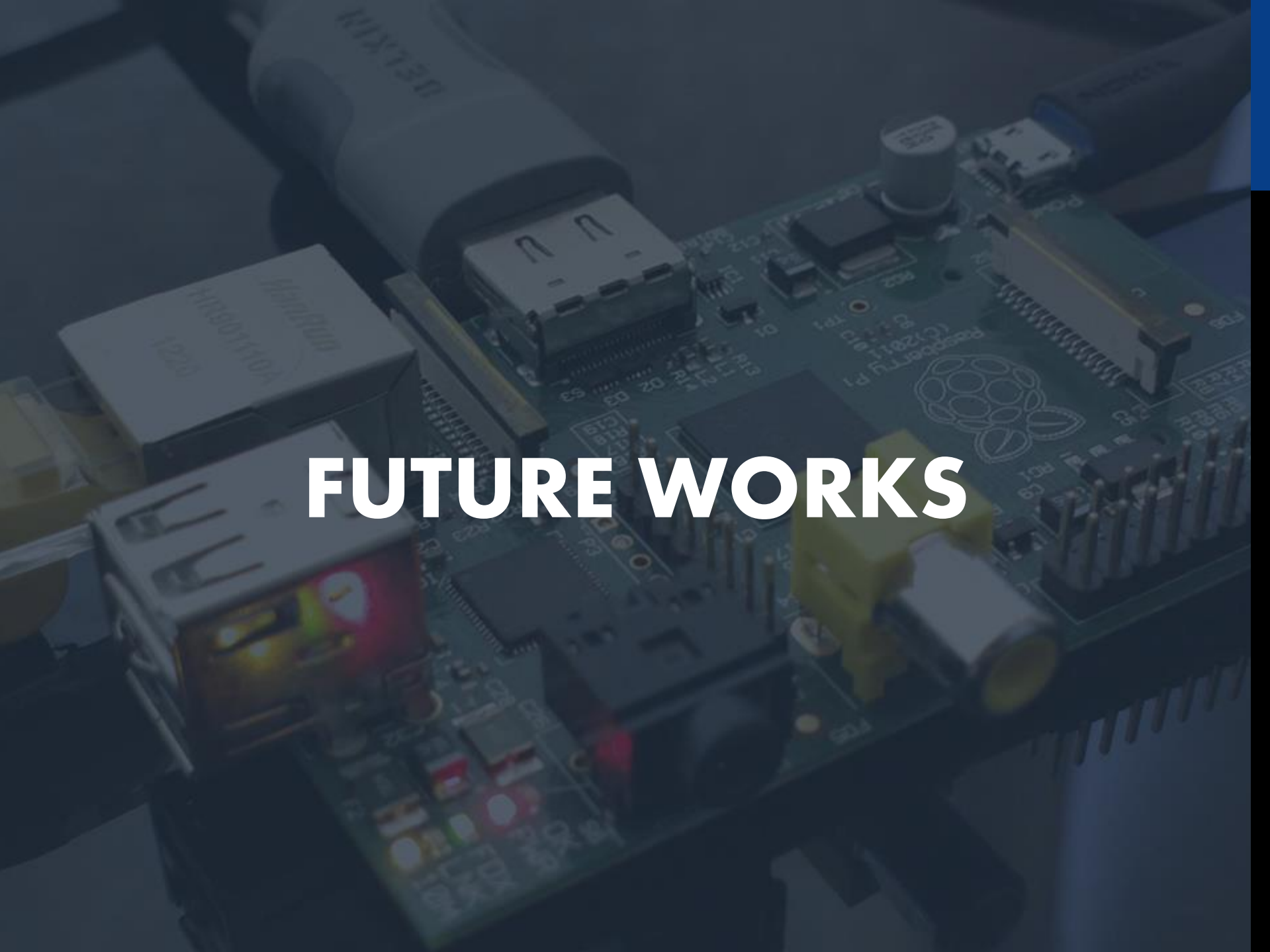
- Take photo when someone press the doorbell
  - fswebcam
  - Apache HTTP Server



# SOFTWARE DESIGN (CONT.)

## ❑ Calling Doorbell





# FUTURE WORKS

## **FUTURE WORKS**

- **Add more sensors**  
CO detector, glass break sensor, etc...
- **Support video for calling doorbell**  
Track your visitor in real-time, not only static image
- **Self-setup at first**  
Fully automatic synchronization at first setup
- **Self-manage existing Camera IP**  
Full of features to manage viewing every camera IP

A person in a white shirt and tie is holding a small electronic device with wires. The device has a red light and some text on it. In the foreground, there is a table with a remote control and a computer mouse. The word "DEMO" is overlaid in the center.

# DEMO



A group of people, mostly men, are seated at a long table covered with a white cloth. They appear to be in a classroom or a meeting room. Some are looking at laptops, others at papers. The room has a whiteboard in the background. The text 'QUESTION & ANSWER' is overlaid in large white letters with a blue underline. The background image is dimmed.

# QUESTION & ANSWER

A group of people, mostly men, are seated at a long table in a meeting room. They are looking towards the left side of the frame, presumably towards a speaker. The room has a whiteboard in the background. The text 'THANK YOU' is overlaid in the center of the image in a large, white, sans-serif font. A small blue horizontal line is positioned below the text.

**THANK YOU**