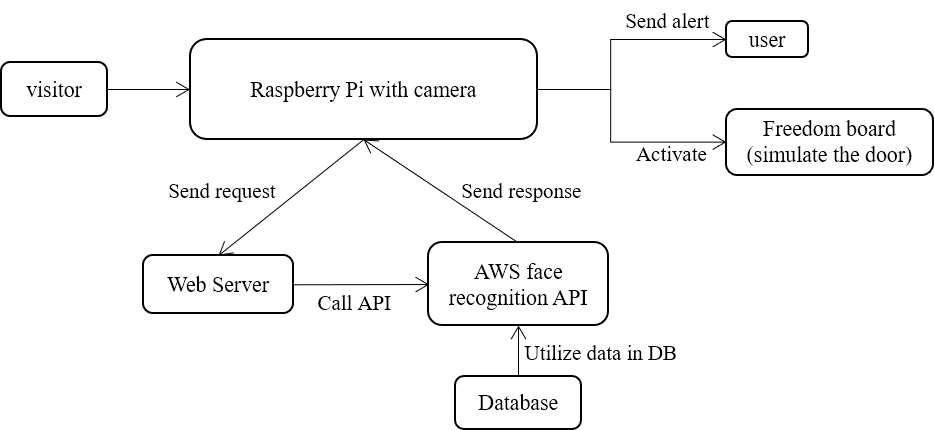
EC 544 Project

Smart Doorbell

|  |  |
| --- | --- |
| Team Member | Email Address |
| Qianhao Liulin | linqianh@bu.edu |
| Ziyu Zhao | zepher@bu.edu |

1. Introduction

We want to build a smart doorbell for modern family which is capable of automatically open the door for friends and family members and send alerts to the user if their home is visited by strangers. Our idea is that use a embedded system to capture and preprocess the photo of visitor, then send the photo to a web server which will classify whether this visitor is permitted to enter, if so, activate the door, if not, record this visit and send alert to the user, and the user could proceed with further operations such as call the police or simply keep the door closed.



2. Resources

1. Raspberry Pi board
2. camera (connect with Pi)
3. freedom board
4. Router
5. AWS face rekognition API
6. Python
7. C/C++
8. Javascript
9. HTML
10. CSS
11. Database

3. Technical Risk Areas & Risk Management

1. Image caption and processing using camera and Pi board, fail to configure camera and Pi
2. Image quality is too low for classification
3. Image classification using AWS face recognition API, fail to effectively to classify input image
4. Fail to construct webserver to handle requests and send response to fulfill the requirement
5. Fail to accomplish communication between Pi, freedom board, and webserver

4. Technical Approach

1. Capture images using camera and Pi board
2. Transmit image from pi board to web server
3. Web Server Configuration

Functions of our webserver:

1. Receive request from Pi board, which contain one image or multiple images of visitors
2. Create, read, update and delete data in database
3. Call AWS face recognition API to match image from request with images in database, analysis response from AWS API, then integrate information and send final response back to Pi, post visit record on the front-end website and send alert (email) to the user

Build web server and web application

1. Use Python3 as server-side programming language
2. Use HTML, CSS and Javascript to construct simple front-end website to post visit records for user to review
3. Activate the door or send alert to the user

5. MileStones

1. Configure the camera and Pi to capture, preprocess, and store the images
2. Construct the webserver which is capable of handling requests from the Pi and process, classify received image
3. Build web app with front-end website which s capable of recording visit history
4. Make sure the server could send the response back to the Pi board
5. Make sure the server could send alert to the user through email
6. Pi board send activate signal to the door