

San Jose State University
Software Engineering Department

Spring 2016



CMPE 296A-IoT

Class Project 2
Project Report

Submitted to:

Prof. Sang Shim

Submitted by:

Group 4

Dhruv Kalaria
Edward Hwang
James Cho
Naghma Anwar
Vivek Maheshwari

Table of Contents

- 1. Motivation**
- 2. Introduction**
- 3. Business Plan**
- 4. High Level Design Architecture**
- 5. High-level Specification**
- 6. Technology Stack**
- 7. Pictures of Device Connection/Setup**
- 8. Instructions for Circuit Connections**
- 9. API Design**
- 10. How to Setup and execute the code**
- 11. Screenshots of Android App**
- 12. Lessons Learned and Possible Future Work**

1. Motivation

Our mission is to provide an automated and remote control of a building or residential door entry. We want to be able to provide a means for a guest or a trusted person to access entry to the building without the need for a key or another person to be physically present. This will create greater convenience and accessibility for entry into an unoccupied building. With our system the owner of the building can allow entry access to visitors while not being around. This makes it possible to allow guest into the home without compromising security for unwanted visitors. Having to leave a hidden key or an access code just for the visitor to use to enter can be a considerable and serious security liability should an unauthorized person come in possession of it.

Not only that but we also wish to create an automated system that will allow an authorized individual such as the owner the ability to access the door automatically without a key or input. This forgoes the need of having a key within possession at all times. We will make the unfortunate event of being accidentally locked out a thing of the past. No longer will one lose entrance to their place simply because they have lost or forgotten their key. Removing this hassle will greatly ensure a greater quality of life by no longer having to remember or account for a key to open the door. Waiting for a locksmith and having to pay for their services will no longer be a factor in play.

The ability to accommodate multiple people without the need of creating more keys to enter makes it all the more secure and accessible. We no longer need to have multiple keys for an entire family of a residential home, we can simply utilize this system so all of this becomes unnecessary. Having numerous keys increases the chance of risk when keys become lost or stolen. This does away with an inconvenience with having to carry and account for the keys needed to enter a home.

Last but not least, the ability to identify your visitor without coming to the door reduces risk and exposure from potentially dangerous and unsavory intruders. With Smart Door, one can simply see who is at the door without having to come to the door. This minimize risk of forceful intrusions and home invasions.

2. Introduction

Smart Door – ‘See who’s there, from anywhere’.

The door to today’s home isn’t just about keeping the bad guys out. It’s also about letting the right people in—your family, friends, and even favorite service providers. With the Smart Door, you are always in control of your front door, no matter where you are, right from your smartphone or computer. You can be anywhere away from home and never miss a door visit from a guest ever again. With Smart Door, you will know who came to visit with the use of our camera system. With just the press of a button you can allow or deny the entrance of the visitor who came to your doorstep. You can allow anyone you trust the access to your home without the need of giving them any keys or door codes that can be stolen or misused.

In addition, Smart Door comes with its own automatic authorized entry system. Smart Door recognizes any pre-authorized individual and unlocks the door for them to enter without any manual or controlled interaction. Just show your face and Smart Door will open the way. No more

remembering to carry a key, which can be lost or stolen. Smart Door let you in whenever you please. Entrance is just a door bell press away.

Unwanted Guests? Go Away! With Smart Door, boring and annoying door to door salesmen and solicitors will be a thing of the past. With Smart Door you can see who is at the door without having to stop what you're doing to answer the door. Just Ignore and deny access to the unwanted visitor and continue to go about whatever task you were doing. With Smart Door your Home is your Castle, with just a movement of your hand you will decide who comes in or stays out.

How it works - With the use of a doorbell button which will activate the peephole camera, one could see/monitor who's there at the front door and unlock the door remotely. So, the camera will take snapshots, send it to the cloud and prompt the user to either unlock the door or deny it. The system will send the prompt to the user's mobile application or web interface. An image of the visitor and a prompt asking to allow or deny entry will show up. The user can then press either the option to allow or deny entry. If the user denies, then the door will remain locked and no entry will be permitted. If the user allows, then the door will be unlocked for the visitor to enter inside.

An automated system for easy entry for an authorized user is also in place. By using Facial Recognition, the system is made to recognize the face of the authorized user. When an authorized user rings the doorbell, the system takes the snapshot and see if it matches anyone who is authorized in the database. If there is a match, the system will bypass the need for a user prompt and automatically unlock the door to allow entrance.

3. Business Plan

Founders and the Vision:

This product will be built by Group 4 (Dhruv Kalaria, Vivek Maheshwari, Naghma Anwar, Edward Hwang, James Cho) of CMPE 296A Internet of Things. The product uses the technology learned in the class and the use of raspberry pi and supported camera board. The final product will enable homeowners to possess a door automation system. The owner will be able to control and see who is at the front door and can allow access. Additionally, the product will be able to recognize the face of the owner and automatically open the door.

Market Opportunities:

Home automation is become a much larger part of the market to homeowners. The global home automation and control market was worth \$5.77 billion in 2013 and is predicted to reach \$12.81 million by 2020. The market is projected to grow at an estimated compound annual growth rate of 11.36% between 2014 and 2020. North America is expected to retain its position as the largest market. The increasing awareness about wireless technologies and various developments related to the integration of wired and wireless technologies will progress the home automation market and control by a great amount.

Finances/Material List:

Material	Financial Listing
Raspberry Pi 2 Model B	Included from class
Raspberry Pi Camera Module	\$19.95(Amazon Listing)
Jumper Wires	Included
Button	Included
Breadboard	Included
USB WiFi Dongle	Included

Table 1. Finances/Material List**Market Size and Growth Rate:**

Our product competes in more than one market. It can be a useful feature for a smart door. It can also provide a system integration application for home security systems. It can also be used for hotels and apartments where residents can register their faces and have the system automatically unlock the doors without need of a key.

Target Market:

Our target customer could be any homeowner or apartment owner who wants to remotely view who's at the main door and be able to remotely control the door.

Competition:

There are several products in the market which allow the homeowner to remotely view the person at the door. However, our product allows the user to also access the door. The user can thus remotely access the door and allow the visitor to enter, if he wants. The homeowner will have the advantage of viewing and controlling the door access on the same mobile app.

4. High Level Design Architecture

The figure below (Figure 1) depicts the typical project architecture and flow of the Smart Door.

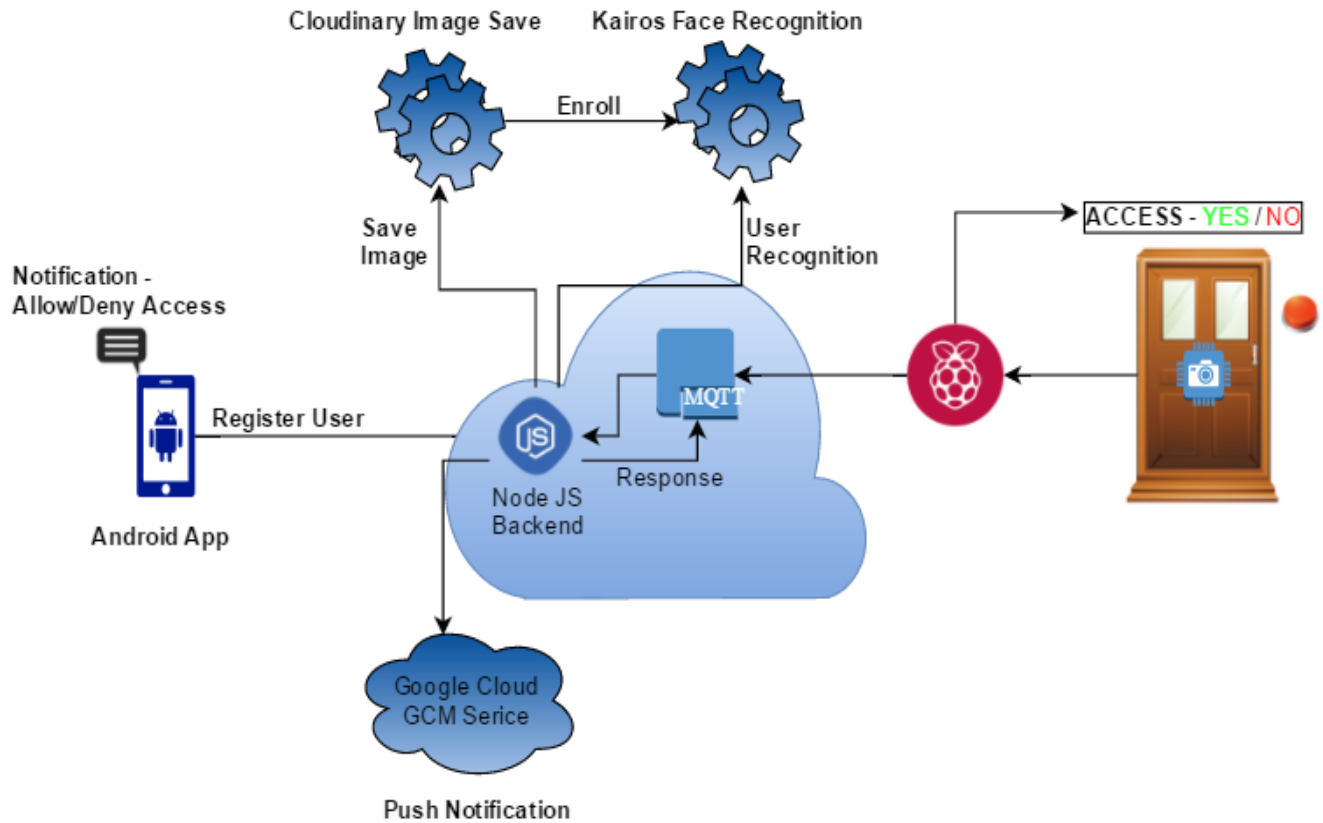


Figure 1. Smart Door Architecture

The above figure represents the architecture and data flow diagram for our application. The Cloud infrastructure consists of NodeJS service for recognizing the user who is at the door. Further, there is an enrollment service which will enroll the user from the android application. Finally, the response from the NodeJS server or the Android app by the user is used at the device to either unlock the door or deny the request.

5. High-level Specification

The high level architecture depicted in the above figure (Figure 1) can be described as -

➤ **Android App:**

The app has following features:

1. User Enrollment

The user who owns the home needs to enroll first via a mobile app. He needs to take 5-6 pictures of his face in order make the facial recognition work. Once enrolled, every time the home owner comes to the door and presses the door bell, his picture will be clicked, and send to Kairos service and perform facial recognition. If the match is a success, the door will be unlocked.

2. Notification:

If someone else other than the people who are enrolled comes to the home and presses the door bell, then the image will be captured and send to cloud for facial recognition. Here, there will be mismatch and this will trigger a notification with the image of the person to the actual registered user. Then based on the image of the person the actual owner can make a decision to unlock the door or not. He can push the unlock button on his app and the door will be unlocked. If the user denies then the door will remain locked.

➤ **Device/Raspberry Pi:**

Raspberry Pi captures the image whenever the user presses the doorbell. The Raspberry Pi Cam is installed in the peep hole of the door. It saves the image and publishes it to Mosquitto broker on cloud.

➤ **Cloud:**

There are 3 important modules deployed on the cloud:

a) Mosquitto - MQTT Broker: The broker acts as a bridge between the NodeJS recognition service and mosquito. It pushes the image received from raspberry pi to recognition service. When the user is not authenticated using the facial recognition, the request goes to Android App and the user decision is conveyed to the raspberry pi through Mosquitto broker

b) Node JS Server: It has 2 service, User Enrollment and User Recognition service. The Enrollment service enrolls the user via the pictures clicked through the Android App. The User Recognition service identifies if the user is enrolled or not. We use the Cloudinary service to store the images of the enrolled users on the cloud, while Kairos service is used to enroll the user and do the facial recognition.

c) 3rd party services: We use Cloudinary Service API to store the image of the user during the enrollment phase. Kairos Service API takes up the images from Cloudinary API and does the user enrollment on a gallery. During the recognition phase, the images from the raspberry pi camera are stored in Cloudinary API and Kairos again uses it for recognizing.

6. Technology Stack

➤ Hardware -

- ✓ Raspberry Pi
- ✓ Raspberry Pi Camera Module
- ✓ Push button
- ✓ Arduino with LCD Display (with RFs)

➤ Software/Technology -

- ✓ Mosquitto Broker
- ✓ Python – Pi Code
- ✓ NodeJS – App Server Code, with following API Services -
 - Cloudinary – Cloud Image Service
 - Kairos – Face Recognition and Analysis
 - GCM – to send notifications to Android App
- ✓ Android Application

7. Pictures of Device Connection/Setup

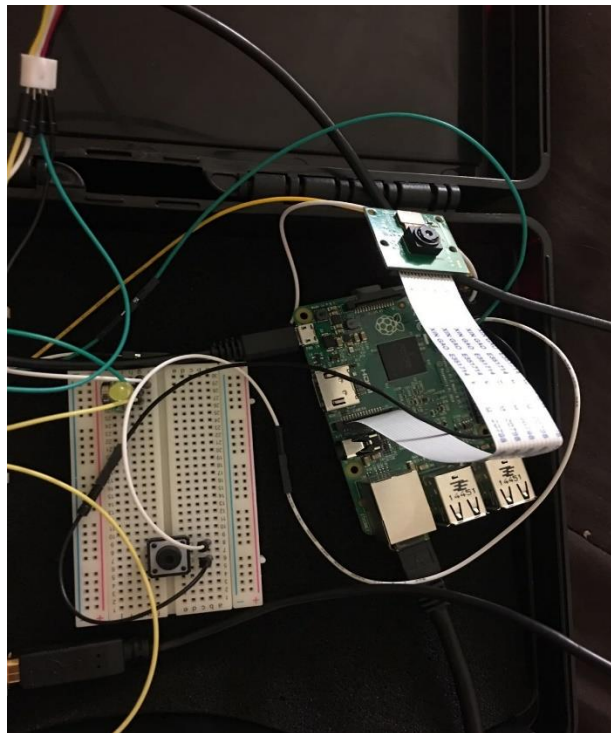


Figure 2. Raspberry Pi - Device

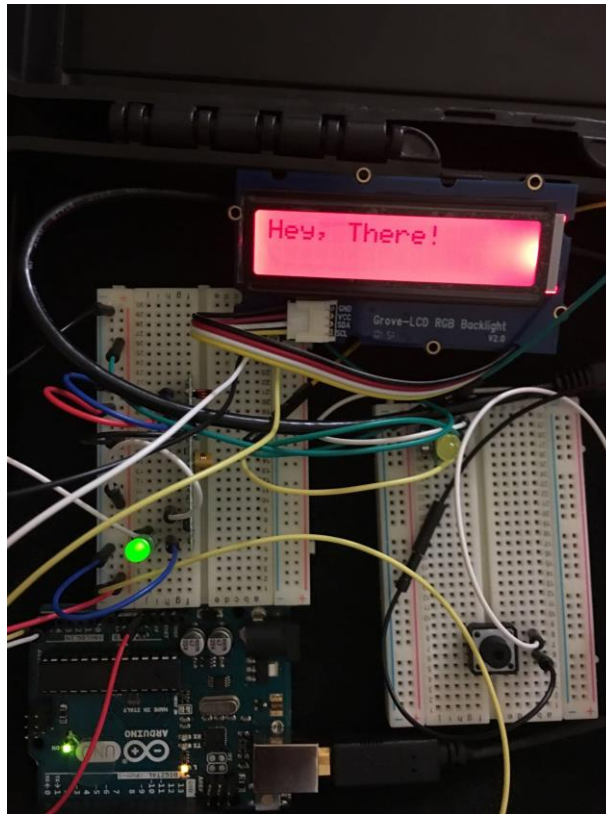


Figure 3. Arduino – LCD Display

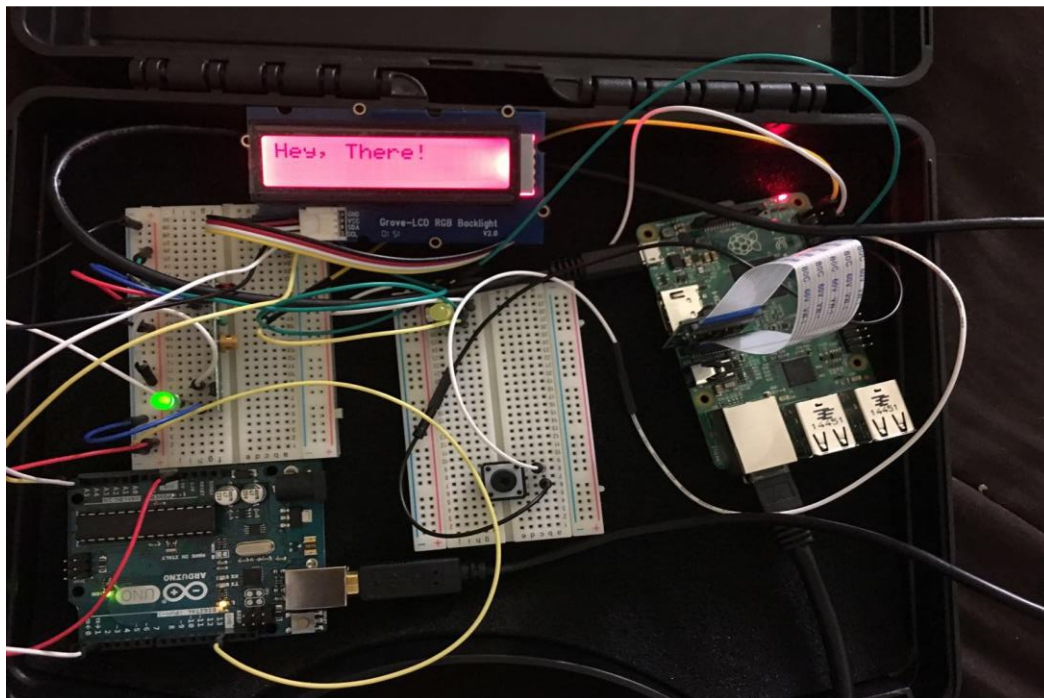
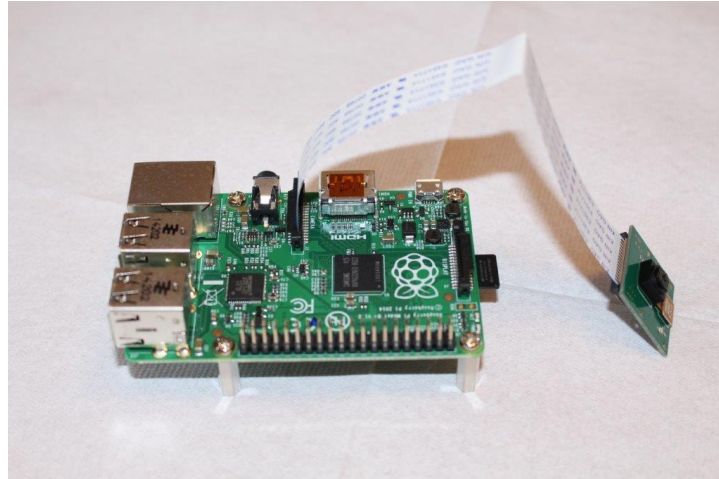


Figure 4. Entire Setup (with LCD)

8. Instructions for Circuit Connections

Below are the steps for the circuit connections -

- ✓ Attach the Raspberry Pi Camera Module as shown in the figure below -



- ✓ Add a push button (acts as doorbell) on GPIO Pin 18.
- ✓ Connect the LCD display as shown in the images above.

9. API Design

- ✓ Since we have a IoT device involved here, we often use the publish-subscribe messaging using Mosquitto message broker via the MQTT protocol.
- ✓ The Application server does have an API, to register/add a user to the owner's list i.e. -
 - **POST /addOwner** – This API is responsible for the registering or adding up of the user to the Home Smart Door access list, so that it can recognize the user the next time.
params = image, userName
where, image is in base64 and userName is String.
- ✓ The Application Server further uses following **Kairos API's** –
 - **POST/enroll** – This API is responsible for enrolling the image onto the Kairos gallery and subject id.
 - **POST/recognize** – This API is responsible for recognizing the image received from the device. It further returns a status, stating success or failure and the confidence percentage.
- ✓ The Application Server also uses the following **Cloudinary API** to upload an image -
 - **POST/upload** – This API is responsible for uploading the image onto the cloud and returns a url for that image.

10. How to Setup and execute the code

➤ Raspberry Pi - Gateway

1. Connect the circuit as described in the Instructions section and as shown in the setup pictures.
2. Connect to Raspberry Pi via SSH.
3. Install the paho mqtt dependency using the command, 'pip install paho-mqtt' and also install the picamera dependency.
4. The Raspberry Pi code has a file, camera_mqtt.py which takes the pictures and pushes the image to our app server. Run the file using the command, 'sudo python camera_mqtt.py'.
5. Make sure the above file is running.

➤ Server -

1. Install the npm module.
2. Install the following dependencies using the npm -
 - fs
 - mqtt
 - cloundinary
 - kairos-api
 - node-gcm
 - express
 - imagemagick
 - Body-parser
3. Provide the Mosquitto broker IP:Port and the GCM server key and registration token.
4. The App server has two files, 'server.js' which recognizes the image send over by the pi camera and if failure, forwards that image to the Android app. If there's a success, it publishes back to the device to open the door.
5. The other file is 'app.js' which enrolls the images from the Android app and adds the users to the gallery to be allowed access.
6. Run the both files separately using the command, 'node server.js' and 'node app.js'.

➤ Android App -

1. Download Android IDE (can be Android Studio or Eclipse Android Plugin).
2. Import the SmartDoorbell project into Android Studio and connect android phone to the computer via USB.
3. Build project and Run 'app'.
4. A GCM token can be seen in console (use this for the server along with server API key within code) to send notifications to the phone.
5. SmartDoorbell should be installed on phone and automatically opened.

11. Screenshots of Android App

The Android application, SmartDoorbell is a simple application which has following features -

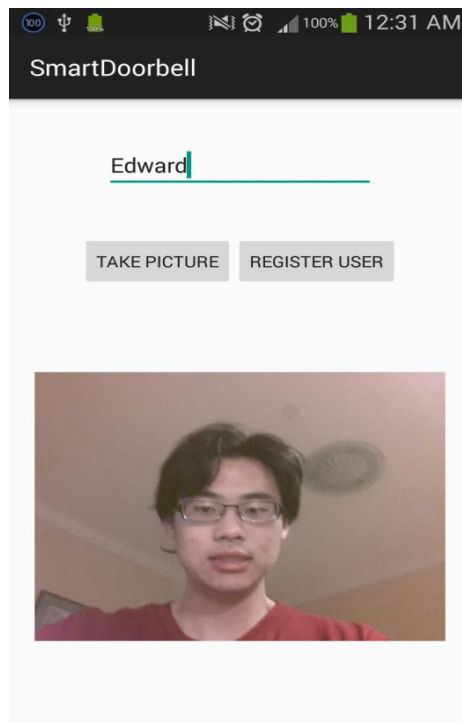


Figure 5. Register User

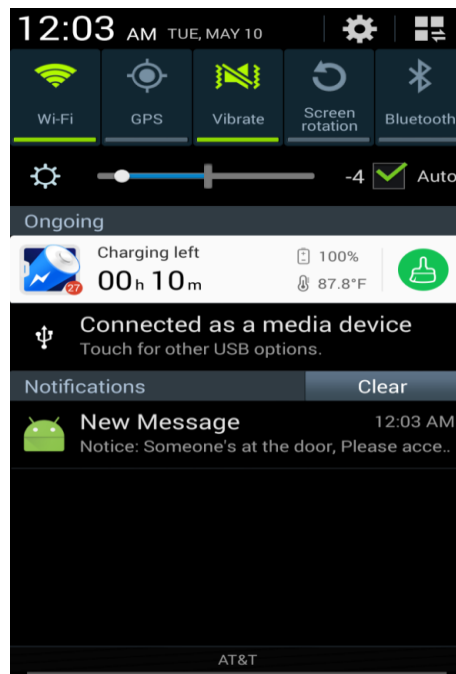


Figure 6. Notification Pop-up

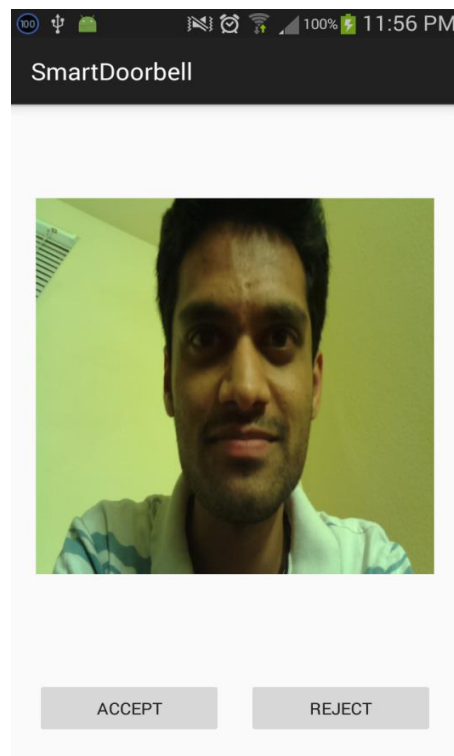


Figure 7. Notification with Image of who's at the door

12. Lessons Learned and Possible Future Work

Lessons Learned:

We implemented the concept of a Smart Door, which activates the camera module (can be mounted on peephole or doorbell) when the doorbell is pressed and sends the picture to the registered user android mobile app, from where the user can either unlock the door or deny the permission. Also, when one of the registered user is at the front door, it checks on with the App server and unlocks the door. So, the product also behaves as a keyless-lock for the registered user. We tried to utilize many of the concepts taught to us in the class by the professor.

Possible Future Work:

- ✓ Make the image recognition system better in terms of performance.
- ✓ Integrate the Doorbell cam with motion sensors, to identify people at the door, even if they don't ring the bell.
- ✓ Speak with the visitors and activate a live video feed.
- ✓ Integrate the Smart door with other IoT devices like Smart Keypad.
- ✓ Develop an iOS application for the Smart Door product.
- ✓ Use HD camera to get a great view of the visitors.
- ✓ Save the recordings of the visitor, to replay a conversation.