Astutë Bell 2.0

A PROJECT REPORT

Submitted by

Deepraj Darbar

Frank Macwan

In fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER ENGINEERING

LDRP Institute of Technology and Research, Gandhinagar

Kadi Sarva Vishwavidyalaya

April/May, 2018

LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH GANDHINAGAR

CE-IT Department

CERTIFICATE

This is to certify that the Project Work entitled **"Astutë Bell 2.0"** has been carried out by **Deepraj Darbar(1418BECE30024)** under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering (8th Semester) of KadiSarvaVishwavidyalaya University during the academic year 2017-18.

Prof. Tejasvee Gupta
Internal Guide
LDRP ITR

Dr. Hiren B. Patel
Head of the Department
LDRP ITR

LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH GANDHINAGAR

CE-IT Department

CERTIFICATE

This is to certify that the Project Work entitled **"Astutë Bell 2.0"** has been carried out by **Frank Macwan(1418BECE30030)** under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering (8th Semester) of KadiSarvaVishwavidyalaya University during the academic year 2017-18.

Prof. Tejasvee Gupta	Dr. Hiren B. Patel
Internal Guide	**Head of the Department**
LDRP ITR	**LDRP ITR**

ACKNOWLEDGEMENT

I take this opportunity to humbly express our thankfulness to all those concerned with my project. First of all, I am thankful to **LDRP-Institute of Technology and Research** for undertaking this project. I am sincerely indebted to **Prof. Tejasvee Gupta** for giving me the opportunity to work on this project. His continuous guidance and help have proved to be a key to my success in overcoming the challenges that I faced during my project work. His support made the project a pleasantly memorable one, without his help at all stages in spite of his own lectures and work load; the completion of the project would not have been possible. I express my sincere gratitude to **Dr. Hiren B. Patel** for his valuable guidance and positive feedback. There are so many people without whose help I would never have conceived and learnt, to whom I would like to express my gratitude – my friends, colleagues, and of course Computer Engineering& Information Technology Department of LDRP – ITR.

Deepraj Darbar
(1418BECE30024)

Frank Macwan
(1418BECE30030)

ABSTRACT

This System provides the user with the facility of home security using the latest technology called IoT (Internet of Things), hereby we present to you the **AstuteBell 2.0** that is the Smart Doorbell with the facility of knowing whether or not there are any malicious activities going on the outside of your house. There will be basically two mechanisms first that is the standard doorbell alert or the second method where the word smart comes into play, i.e. if any person standing/wandering in front of the gate, his/her video will be captured and would be checked against the pre-existing database of all the known faces of the family and if there no match is found then the owner will be notified with a message and the recording that someone is outside you house and adding to it we can also unlock the door, if the person's face is recognized by the system. This technology will help the current on-the-go generation to keep them and their loved ones safe.

By providing continuous monitoring and smart notification system from anywhere and anytime. This is our one step to add to the nation moving towards Digital India we call it making it the **SecureIndia** project.

TABLE OF CONTENTS:

LIST OF FIGURES

1. Introduction

1.1 Introduction

The Astutebell that is our IoT based Smart doorbell provides home security without the need of constant or even little surveillance, i.e. the system will notify the user with image/video if there is presence of any unrecognized person(via face recognition) in front of the premises more than the time set or if he/she rings the bell, and also will upload to an online drive for backup with the new facial recognition feature added will help to identify the known people and won't notify on their arrival as well as would unlock the door for the recognized person.

1.2 Scope

The product of this project is a demo version of Astute Bell, this application is very useful for security purposes. It helps you to keep an eye on those who keep an eye on you. Besides it being user-friendly and cost efficient, it is multi-functional product.

1.3 Project Summary and Purpose

The following documentation is to brief about a prototype of AstuteBell 2.0.

1. Identify ambiguities, inconsistencies, and incompleteness from a requirements specification
2. Identify and state functional requirements
3. Identify and state non-functional requirements

The Project summary provides with all the procedure that it took to build this project and also the code part and the hardware networking and the server-side phase of the project. The product is a stand-alone application which can be used in any part of the home, office, or other buildings for security purpose. There are some products available in the market which resemble this product, but there are more features added to this such as a PIR sensor which senses the presence and Face detection that identifies registered people (and doesn't generate alert for them), capture images of the object which causes motion, saving the images to drive for further reference, accessing the system even when we are far away from it with live notifications and we do not require to monitor the system to make use of it.

1.4 Overview of the project

The next chapter of this SRS document describes the tools and technology used in building the project and also a brief on work done in past.
The Third chapter describes the System Requirements of the project that is the hardware

and software requirements of the project and the assumptions made under working circumstances.
Fourth chapter covers the system analysis phase of the project where in all the strength and weaknesses of the project are mentioned, as well as the functions of the product are mentioned also a few diagrams for understanding of it's working.
The last chapter describes the system design and the features of the project and also the code used to create the working prototype.

2.Technology and Literature Review

2.1 About Tools and Technology

Raspberry Pi3 and RaspbianOS

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside of its target market for uses such as robotics. Peripherals (including keyboards, mice and cases) are not included with the Raspberry Pi. Some accessories however have been included in several official and unofficial bundles, and here we are using the raspberry pi3 model which comes with built-in wireless lan chip which enables it to be connected to the network that allows it to send real-time notification and it has a slot for Pi camera with which we have integrated the face detection module in it.

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make your Raspberry Pi run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi.

OpenCV (Open Source Computer Vision)

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. The library is cross-platform and free for use under the open-source BSD license.

OpenCV supports the deep learning frameworks TensorFlow, Torch/PyTorch and Caffe. It is used for Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel, disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable, Advance vision-based commercial applications by making portable, performance-optimized code available for free – with a license that did not require code to be open or free itself. OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE.

OpenCV's application areas include:

2D and 3D feature toolkits
Egomotion estimation
Facial recognition system
Gesture recognition
Human–computer interaction (HCI)

Mobile robotics
Motion understanding
Object identification
Segmentation and recognition
Stereopsis stereo vision: depth perception from 2 cameras
Structure from motion (SFM)
Motion tracking
Augmented reality

Flask microFramework (API)

Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. It is BSD licensed.

The latest stable version of Flask is 0.12.2 as of May 2017. Applications that use the Flask framework include Pinterest, LinkedIn, and the community web page for Flask itself.

Flask is called a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program.

Sensors (PIR)

A **passive infrared sensor** (**PIR sensor**) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view.

An individual PIR sensor detects changes in the amount of infrared radiation impinging upon it, which varies depending on the temperature and surface characteristics of the objects in front of the sensor. When an object, such as a human, passes in front of the background, such as a wall, the temperature at that point in the sensor's field of view will rise from room temperature to body temperature, and then back again. The sensor converts the resulting change in the incoming infrared radiation into a change in the output voltage, and this triggers the detection. Objects of similar temperature but different surface characteristics may also have a different infrared emission pattern, and thus moving them with respect to the background may trigger the detector as well.

Raspberry Pi Camera

The Raspberry Pi Camera Module is a 5-megapixel custom designed add-on for Raspberry Pi, featuring a fixed focus lens. It's capable of 2592 x 1944 pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video. It attaches to Pi by way of one of the small sockets on the board upper surface and uses the dedicated CSi interface, designed especially for interfacing to cameras, which on integrating with the picamera module in the python programming language provided the means to perform face recognition to identify known people.

Relay switch

A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a separate low-power signal, or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits as amplifiers: they repeated the signal coming in from one circuit and re-transmitted it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

Smart Lock

Smart Lock is made by using a micro controller inside a mechanical lock which when the circuit gets complete on a signal HIGH then it pulls the locking mechanism and unlocks the door for the user else remains closed.

2.2 Brief History of Work Done

We built a working prototype of the Astutebell in the last semester in which it could detect presence of a person outside the premises of the owner's house i.e. in front of the door and generate a notification for the same if he/she doesn't press the doorbell within 15 seconds of their arrival and the user would get a real time notification displaying an alert that somebody is outside your door. This was achieved with the use of Raspberry Pi and PIR sensor which were used to detect the presence and a webcam that captured the image of the intruder into which we connected all the components of the system in synergy with the help of python code and further adding to it we are now moving to towards the second model of our astutebell by adding the ability of face detection which would improve its usability by ignoring the registered users and not generating a notification for them.

\

3.System Requirements Study

3.1 User Characteristics

Home automation: This feature helps in electronically managing the entry of each person who visits the house and also maintain their data. As a result, even if nobody is present in the home we can know who visited us at that time.
Security System: This feature highlights the safety purpose. The system captures an image of the object the passes by the sensor. As a result, if there is someone who is trying to keep an eye on your house will be caught without his knowledge
Face Detection: This feature helps in identifying who is there outside the door and also would be helpful in sending alerts to user only for non-registered users.

3.2 Software and Hardware Requirements

3.2.1 Software Requirements

Programming of the product is done in Python language. Hence it needs some of the libraries of Python.

• A Remote Desktop Client – Putty/Remote Desktop Connection

• Advanced IP Scanner – To view the IP address assigned to the Raspberry Pi and setting up Static IP

• Installing picamera and mime modules to use the raspberry pi camera and also to send notification via email.

• Installing OpenCV and Python libraries on the raspberry pi system.

3.2.2 Hardware Requirements

First hardware that interacts will be the sensor, which will sense the motion. If the constraints are satisfied, the system will trigger the automation of camera which will identify the person with the help of face recognition and capture the image of the object causing movement if it's not a known member. Also, after this, the system will send and store the image of an intruder in the drive automatically and notify the user via notification and an e-mail.

• A power supply (Max 5V) Recommended – Portable Battery

• A speaker for ringing the bell (Demonstrative)

• Raspberry Pi camera for the software to run for face recognition

• PIR sensor (last build for motion detection)

3.2.3 Functional Requirements

- Functional requirements of this product will be to provide a remote access interface to the user/owner anytime
- Notification to users in real-time
- Provide storage of photos/videos on cloud storage for backup and review.
- The Application should also allow the user to view current status of the premises for surveillance

3.2.4 Non-functional Requirements

Performance Requirements

The things that the system requires person's face and motion. Hence to fulfill the security aspect, movement of the object which is considered susceptible is necessary also an uncovered face to identify the unique features of the face that differentiate between two individuals.

Safety Requirements

As the product has many hardware parts, it will be advisable to keep them away from water, direct sunlight or any such thing that can damage the product.

Security Requirements

Images/videos will be stored on a drive. Hence in order to save those from being manipulated by others, it is advisable to keep the username and password of the drive being used, safe.

Software Quality Attributes

The product is a multi-purpose application. Easy to maintain and is portable. Also, it is redundant, as it gets the image only when motion is sensed, saving the rest of space stored by the picture

3.3 Constraints

3.3.1 Operating Environment

The product works on a microprocessor “Raspberry Pi” version 3.0 which has a Raspbian OS along with a PIR Sensor, Pi Camera, some jumper cables, a display screen and is linked with drive to store images/videos.

3.3.2 Design and Implementation Constraints

The system may face a constraint of image redundancy where it can get same pictures of a person using up the memory on the drive where they are stored. Also, the sensor detects motion and not human, so there are chances that some non-human moving object causes the whole process which will be in vain.

3.3.3 Parallel Operations

The project is not on basis of a multi-user environment. This is used for carrying out updating as well as entry by preventing the redundancy of the data.

3.3.4 Reliability Requirements

Reliability requirements of the system are one of the prime factors to the list. The system is needed to be highly reliable in terms of performance and capable of delivering robust performance. If the reports are generated within 5 seconds then the system is said to be reliable.

3.3.5 Criticality of the Application

The system can stop working on computers with very low internet connection. Other than that, there won’t be any issues. Apart from these the system should be able to make updates at regular time intervals.

3.3.6 Safety and Security Consideration

Safety and security too are other major concerns of any system. It is necessary to provide safety and security as the system is web application and might be intrude by security threats from the internet. Thus, the code needs to be encrypted and any transaction needs to be done securely.

3.3.7 Hardware Limitations

Hardware Limitations are other constraint of the system. Hardware Limitations should be overcome for better performance of the system. This can be achieved by using minimum and only necessary hardware.

3.3.8 Power Supply

Raspberry pi 3.0 requires supply of 2.5 A current to function properly. If the supply is not abundant, the functioning of the product might halt which yield reliability issues. Thus, the system should be connected to sufficient and unfaltering power supply.

3.3.9 Regulatory Policies

Regulatory policies about achieving organization's objectives through the use of regulations, laws, and other instruments to deliver better economic and social outcomes and thus enhance the life of business. Thus, the system should be developed by using these regulations to provide better outcome to the company.

3.4 Dependencies

The product is entirely based on the connection with the router, so if there is any problem with the router, the product may not work properly**.** Network is the key to functioning of the product.

4.System Analysis

4.1 Study of current system

Currently no System provides the exact same functionalities as ours or they have not been released / patented yet but similar products are available in local and foreign market which are listed as below along with their product operation.

1. SkyBell: -
A visitor presses the doorbell and SkyBell sends a live feed alert to your smart phone. See, hear, and speak to your visitor with free app for selected iOS and Android devices, no matter where your day takes you.

2.August Doorbell Camera: -
Get alerts every time the bell is rung and see and speak with visitors using your Smartphone from wherever you are. August Doorbell Cam detects motion from visitors, identifying people at your door, even if they don't ring the bell, and sends you an instant alert to your smartphone
.

4.2 Problem and Weaknesses of Current System

The Problem and weakness with the existing system is that either the system is not as intelligent as to detect the intruder automatically it only allows you to see/record the video of the premises with any application and a very few systems which do provide detection mechanism are way too costly for a layman to afford and contribute to safety. So, in order to overcome these difficulties, we are trying to come up with the most robust as well as affordable alternative to the present system available.

4.3 Product Perspective

The product is a stand-alone application which can be used in any part of the home, office, or other buildings for security purpose. There are some products available in the market which resemble this product, but there are more features added to this such as a PIR sensor which senses the presence and capture images of the object which causes motion, face detection which identifies who is outside your door, saving the images to drive for further reference, accessing the system even when we are far away from it.

Below are shown some of the main components used in the product.

- Pi camera
- PIR sensor
- Raspberry Pi 3
- API
- Smart lock

4.4 Product Functions

- Sense the presence
- Identify the person (face detection)
- Capture image/video
- Save it to the drive
- Notify the user/Identify User
- Unlock the door

4.5 Flow Diagram

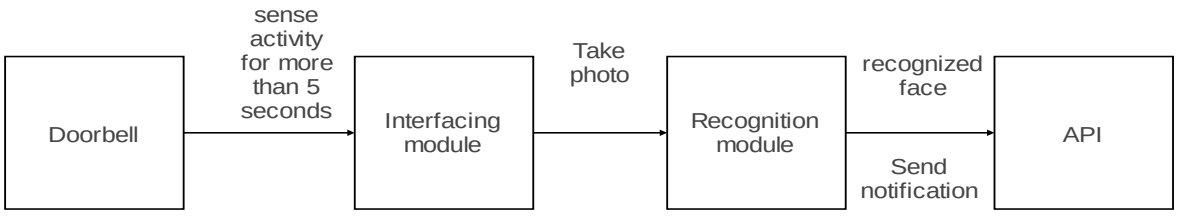


Figure 1

4.6 Feasibility Study

In the conduct of the feasibility study, we consider seven distinct, but interrelated types of feasibility. They are: -

1. Technical feasibility
2. Economic feasibility
3. Social feasibility
4. Management feasibility
5. Legal feasibility
6. Time feasibility

TECHNICAL FEASIBILITY

This is concerned with specifying and software that will successfully satisfy the user requirements. The technical needs of the system may vary considerably, but might include:

- The facility to produce outputs in given time.
- Response time under certain conditions.
- Ability to process a certain volume of tasks at a particular speed.
- Facility to communicate data to distant location is the most difficult to determine.

ECONOMIC FEASIBILITY

Economic analysis is the most frequently used technique for evaluation the effectiveness of a proposed system. More commonly known as Cost/benefits analysis: the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement the system. Otherwise further justification or alternative in the proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each of the system life cycle.

SOCIAL FEASIBILITY

Social feasibility is a determination of whether a proposed project will be acceptable to the people or not. This determination typically examines the probability of the project being accepted by the group directly affected by the proposed system change.

MANAGEMENT FEASIBILITY

It is determination of whether a proposed project will be acceptable to management. If management does not accept a project or gives a negligible support to it, the analyst will tend to view the project as a non-feasible one.

LEGAL FEASIBILITY

Legal feasible is determination of whether a proposed Project infringes on known acts statutes as well as any pending legislation. Although in some instances the project might appear. Sound, on closer investigation it may be found to infringe on several legal areas.

TIME FEASIBILITY

Time feasibility is a determination of whether a proposed Project can be implemented fully within a stipulated time frame. If a project takes too much time it is likely to be rejected.

4.7 Requirement Validation

The only functional requirement that is required from the user is to be logged in to the application as the owner or by the profile he/she has been assigned to be notified and monitor things

4.8 System Activity (Use case diagram)

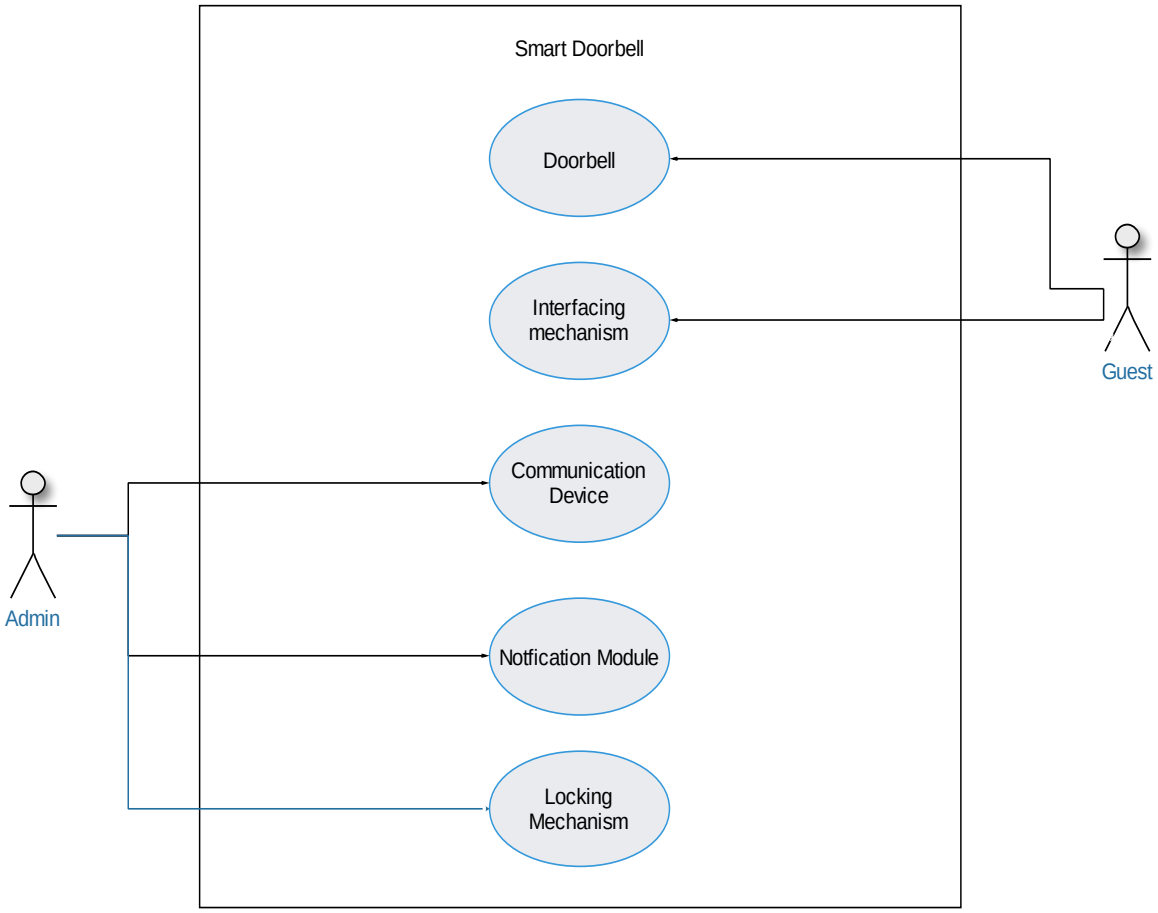


Figure 2

5.System Design

5.1 System Features

The following shows the important features of the product:

5.1.1 Automatic Photo Capture

Description and Priority

When the sensor module detects anyone standing in front of the device for more than 10 seconds, it immediately triggers a camera activation sequence. It is a high priority feature as it forms a basis for other features to work.

Stimulus/Response Sequences

As and when the camera is active, the python code to store this picture if required onto the cloud storage executes and is then stored on the cloud. The notification module then sends the required generated link to the registered mobile device.

Functional Requirements

REQ-1: All required python libraries installed
REQ-2: Predefined Resolution of images

5.1.2 Face Detection

Description and Priority

When the sensor module detects anyone standing in front of the device the camera is activated and it runs the face recognition code, it immediately triggers a sequence which runs the scan against the existing faces in the database and captures image if no match found. It is a high priority feature as it forms a basis for other features to work.

Stimulus/Response Sequences

As and when the photo is captured, the python code to store this picture onto the cloud storage executes and is then stored on the cloud. The notification module then sends the required generated link to the registered mobile device.

Functional Requirements

REQ-1: All required python libraries installed
REQ-2: Predefined Resolution of images

5.1.3 Real-Time Notifications

Description and Priority

Once stored onto the cloud, the notification module comes into picture where it uses the necessary API keys to authenticate and send the notification with the link to an image. It is also a high priority feature as this is the primary objective of this project.

Functional Requirements

REQ-1: API and Secrets Keys registered for cloud access
REQ-2: Device connected to the internet for real-time notifications

5.2 System Application Design

The following documentation represents the code/logic part of the product.

5.2.1 Pseudo Code

```
import face_recognition
import RPi.GPIO as GPIO
import time
import numpy as np
import cv2
from datetime import datetime
import shutil
import os
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email.mime.text import MIMEText
from email import encoders
from pushbullet.pushbullet import PushBullet
import picamera
print("all libraries loaded")
gmail_user = "macwanfrank1996@gmail.com" #Sender email address
gmail_pwd = ".1.2.3.7.8.9" #Sender email password
to = "macwanfrank96@gmail.com" #Receiver email address
subject = "Security Breach" #Email subject
text = "There is some activity in your home. See the attached picture." #text you want to send in email
```

```
known_image =
face_recognition.load_image_file("/home/pi/Desktop/Projectfiles/unknown.jpg"
)
sensor = 4 #pin connection of pir sensor to raspberry pi

GPIO.setmode(GPIO.BCM)
GPIO.setup(sensor, GPIO.IN, GPIO.PUD_DOWN)

previous_state = False
current_state = False
visitor=0

while True:

previous_state=current_state
current_state=GPIO.input(sensor)
if(current_state != previous_state):
      if(current_state): new_state="HIGH"
      else: new_state="LOW"
print("GPIO pin %s is %s" % (sensor,new_state))

      if(current_state):
print("A photo will be taken in 3 seconds")
time.sleep(3)
         with picamera.PiCamera() as camera:
camera.resolution = (340, 240)
time.sleep(2)
camera.capture('unknown.jpg')
unknown_image =
face_recognition.load_image_file("/home/pi/Desktop/unknown.jpg")
face_locations = face_recognition.face_locations(unknown_image)
         if(len(face_locations) == 0):
print("no face detected")
            continue
biden_encoding = face_recognition.face_encodings(known_image)[0]
unknown_encoding = face_recognition.face_encodings(unknown_image)[0]
print("encodings calculated")
         results = face_recognition.compare_faces([biden_encoding],
unknown_encoding)
         print(results)
```

```
        if results == [True]:
print("person recognised, it is frank")
            p = PushBullet("o.1SFONRUTMV1ftRffqL474fLituOMlz3m")
#secret key of pushbullet API
            devices = p.getDevices()
p.pushNote(devices[0]["iden"], 'Frank is at the door', 'open the door')
os.remove("unknown.jpg")
print("message sent")
time.sleep(10)

         else:
        os.rename("/home/pi/Desktop/unknown.jpg","/home/pi/Desktop/"+str(vis
        itor)+".jpg")
        shutil.move("/home/pi/Desktop/"+str(visitor)+".jpg","/home/pi/Desktop/
        known")
print("unknown person")
            with picamera.PiCamera() as camera:
camera.resolution = (1280, 720)
time.sleep(2)
camera.capture(str(visitor)+'.jpg')

print("Saving Photo")
picname = str(visitor) + ".jpg" #image name
            visitor=visitor+1
            #cv2.imwrite(picname,frame)
            p = PushBullet("o.1SFONRUTMV1ftRffqL474fLituOMlz3m")
#secret key of pushbullet API
            devices = p.getDevices()
p.pushFile(devices[0]["iden"], picname, "unknown person", open(picname,
"rb"))

print("Sending email")
            attach=picname #attachments to send with email
msg=MIMEMultipart()
msg['From']=gmail_user
msg['To']=to
msg['Subject']=subject

            part=MIMEBase('application','octet-stream')
```

```
part.set_payload(open(attach,'rb').read())
encoders.encode_base64(part)
            part.add_header('Contetnt-Disposition','attachment;filename="%s"'%
picname)
msg.attach(part)
mailServer = smtplib.SMTP("smtp.gmail.com",587)
mailServer.ehlo()
mailServer.starttls()
mailServer.ehlo()
mailServer.login(gmail_user, gmail_pwd)
mailServer.sendmail(gmail_user,to,msg.as_string())
mailServer.close()
print("email sent")
os.remove(picname)
time.sleep(5)
```

5.3 Activity Diagram

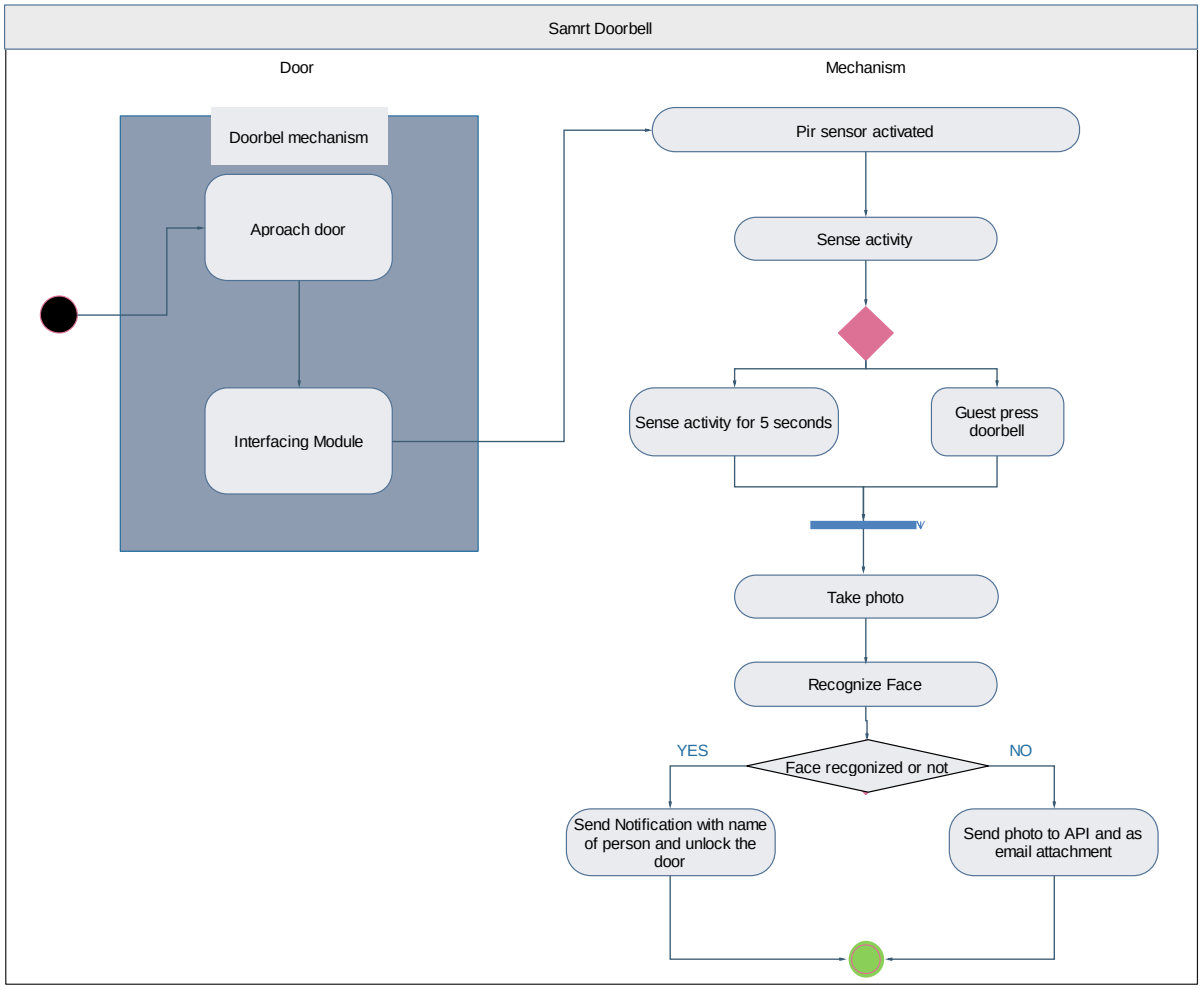


Figure 3

5.4 Sequence Diagram

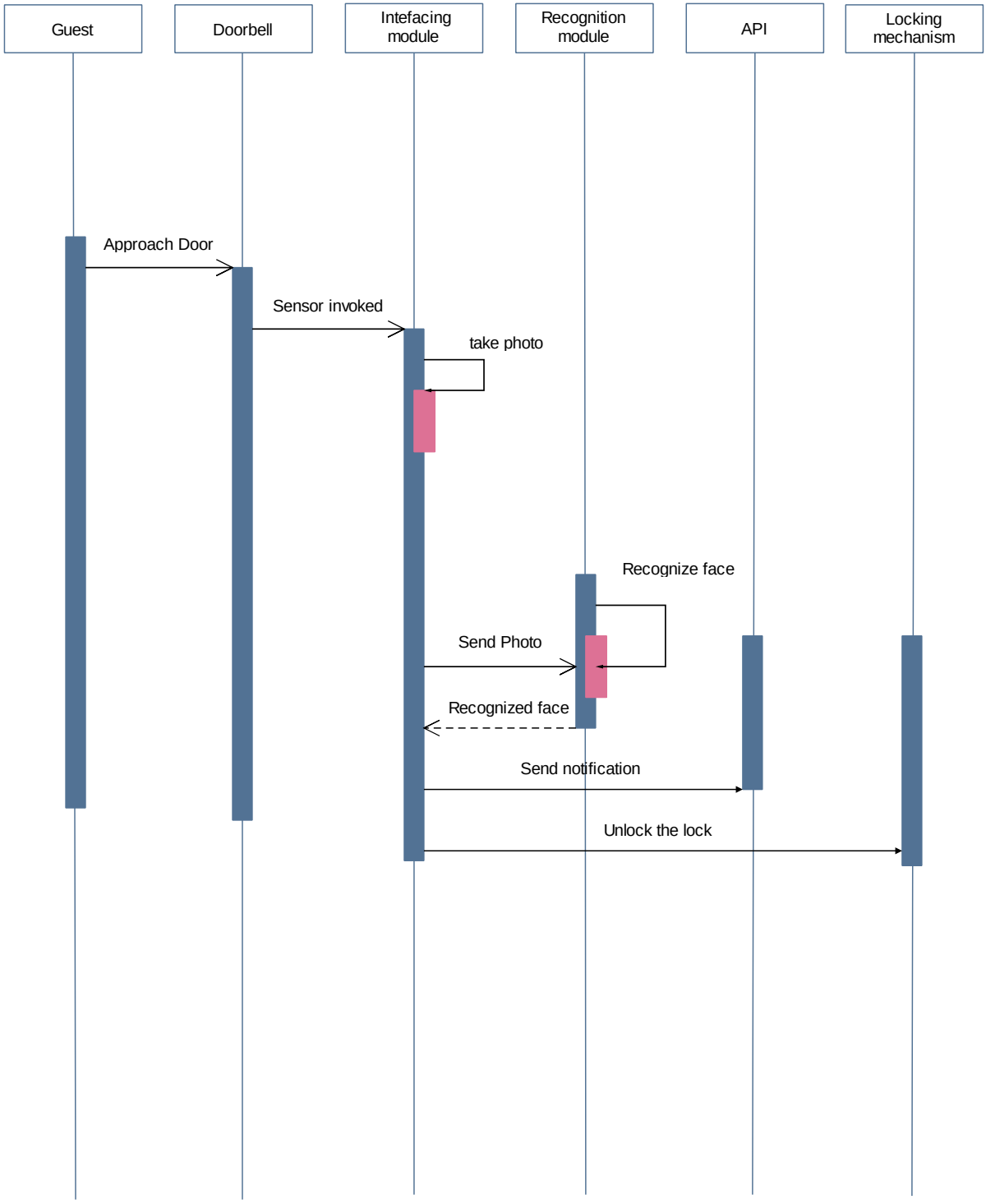
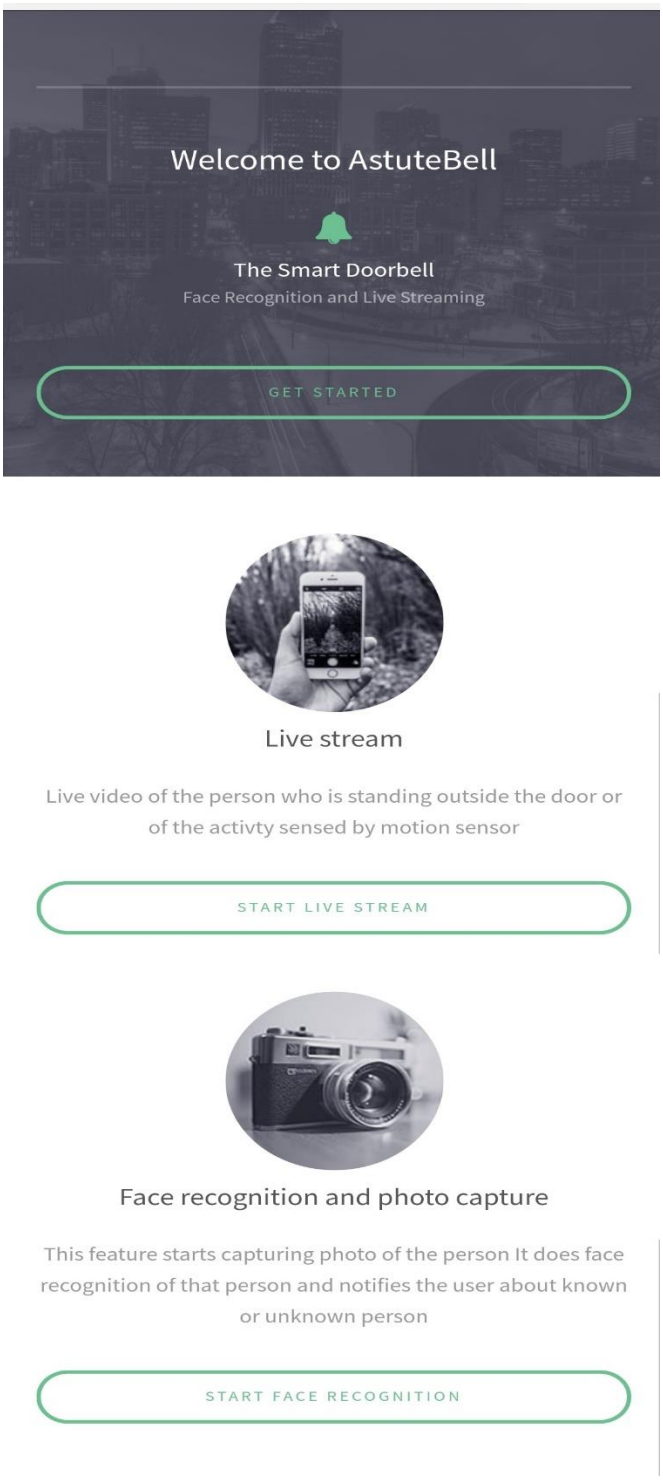
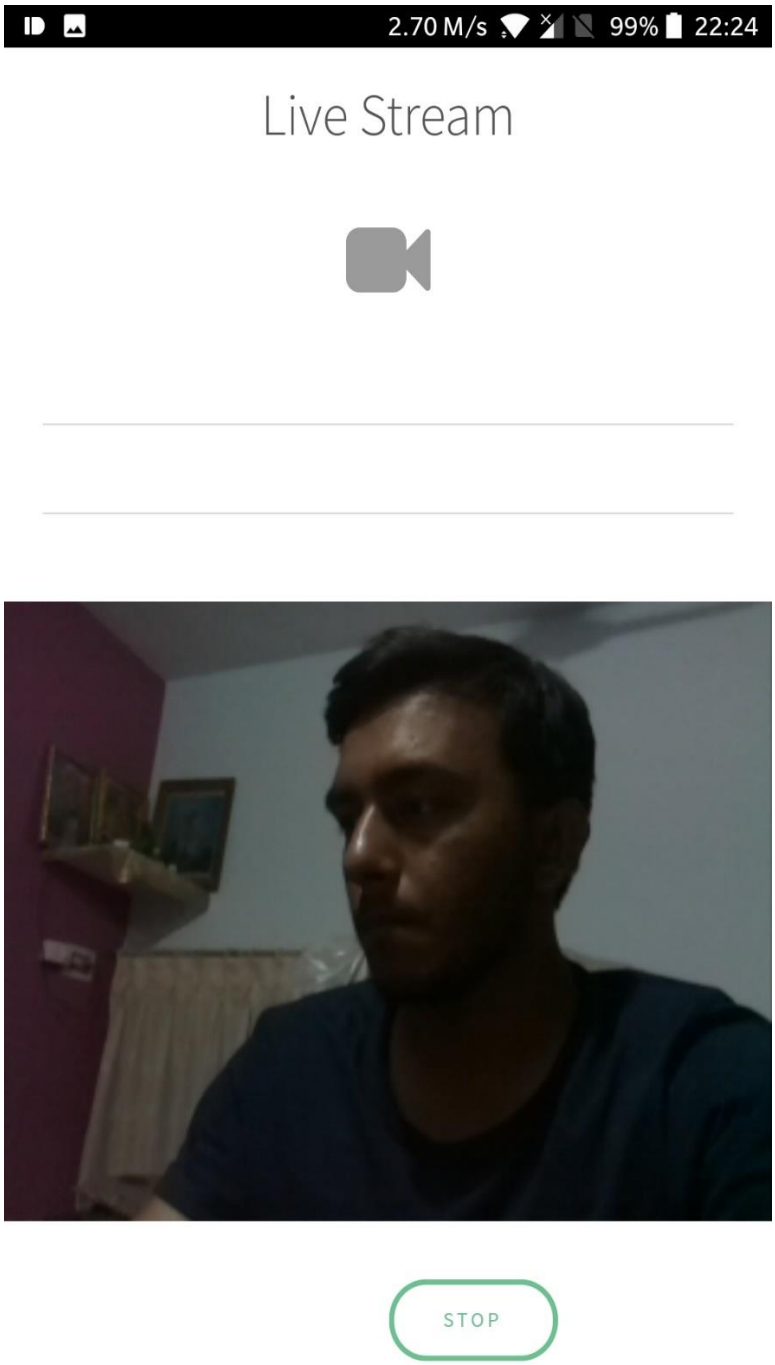


Figure 4

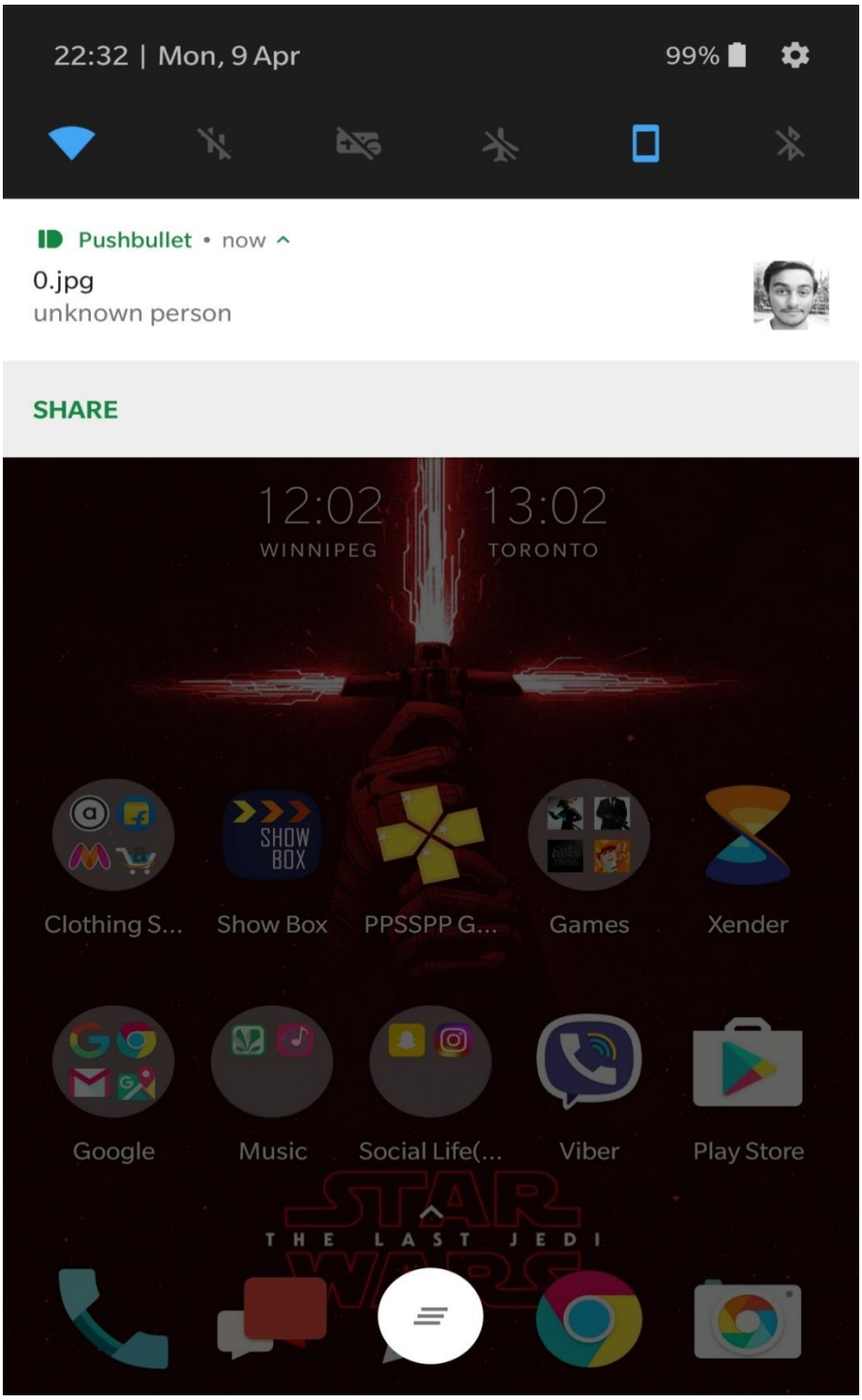
6. Screenshots



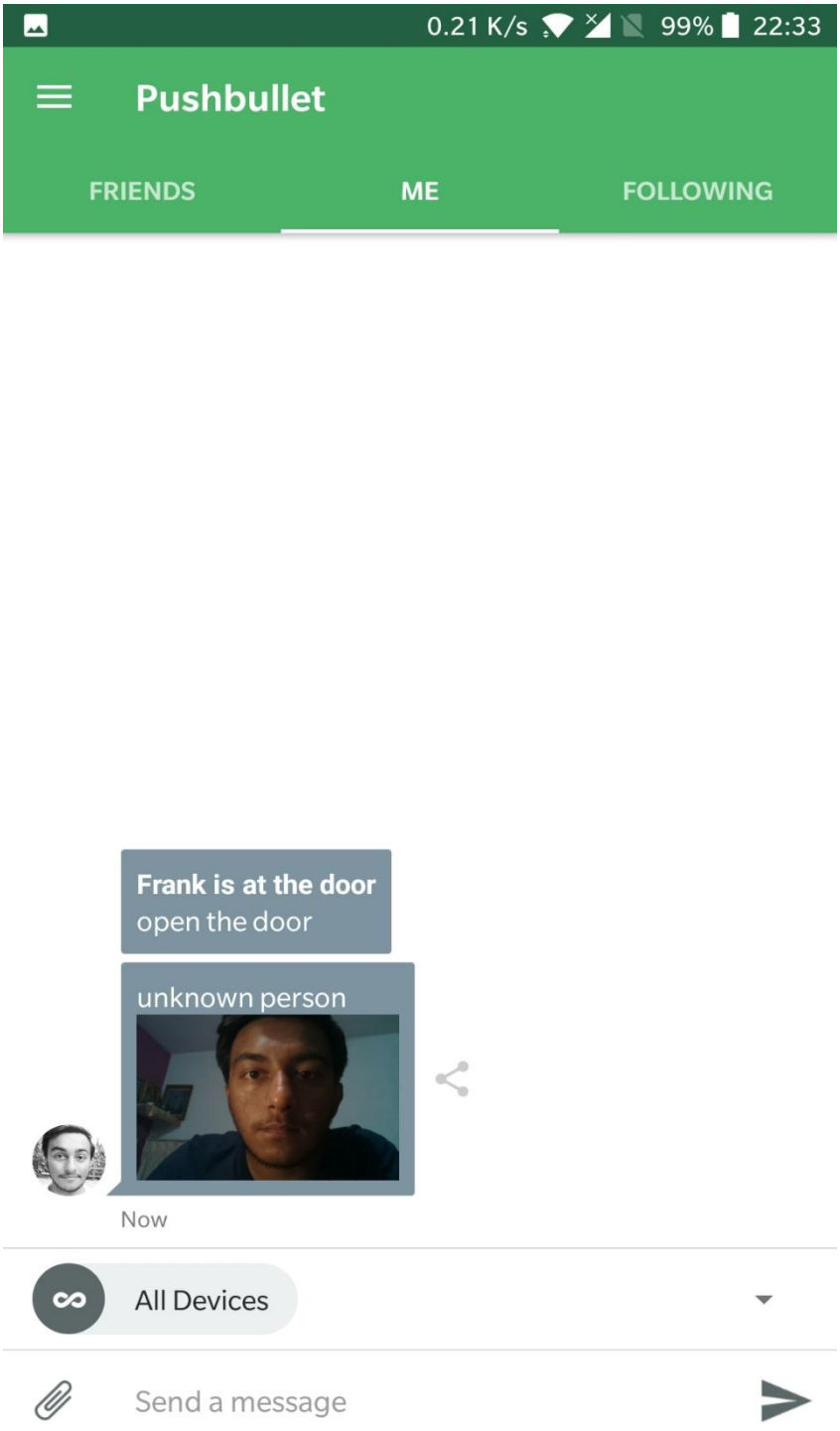
6.1 Webpage



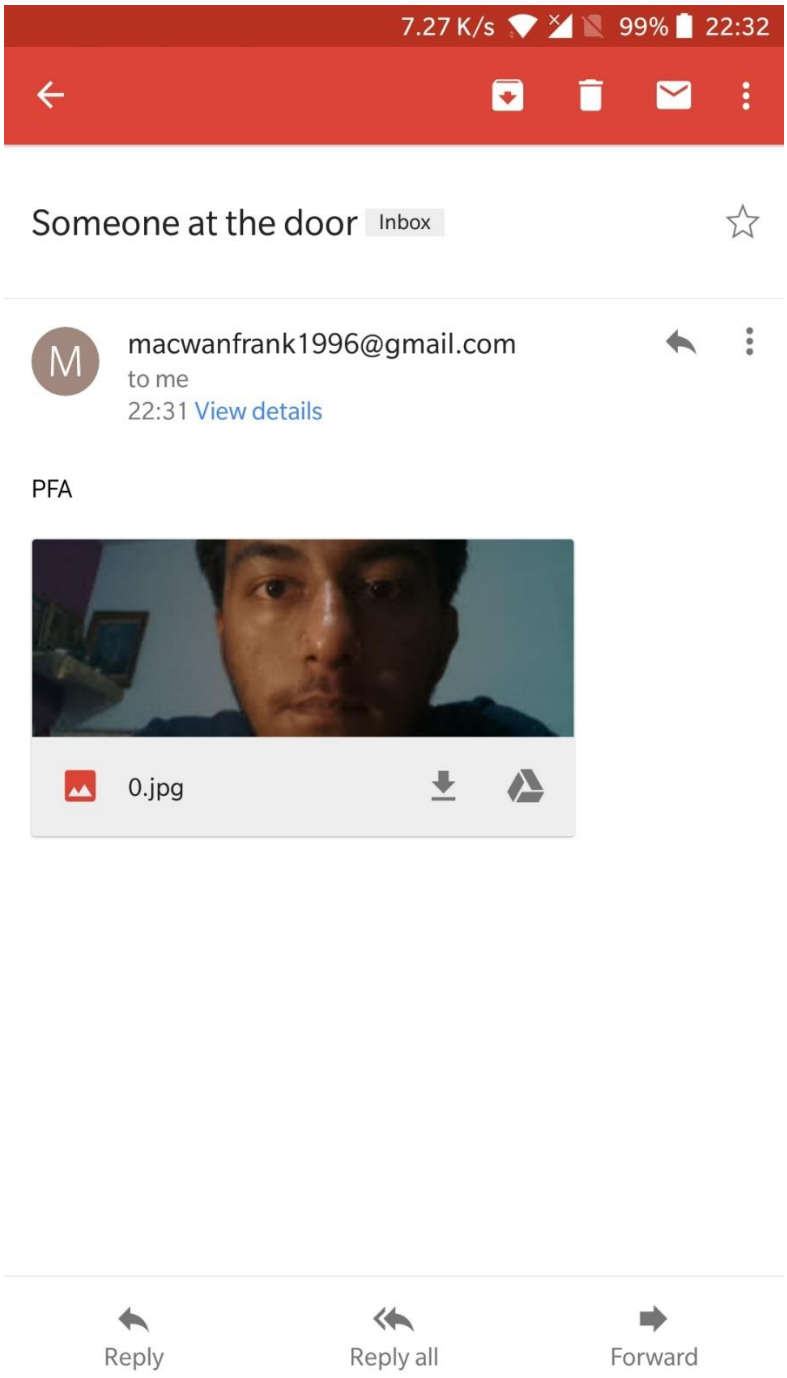
6.2 Live Stream



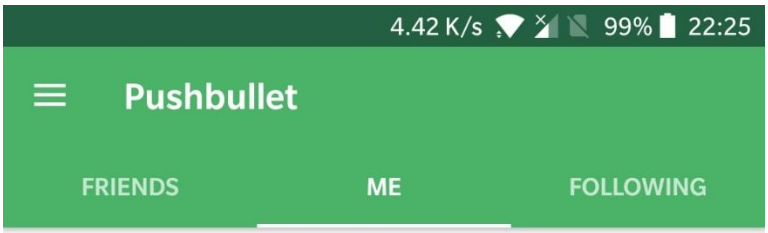
6.3 Unknow person notification

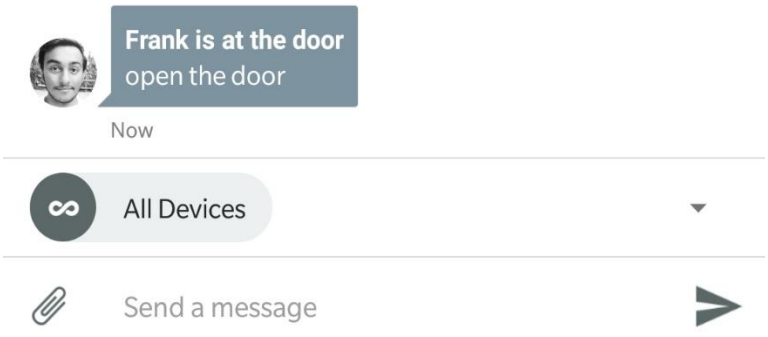


6.4 Unknown person image in API



6.5 Email with photo of unknown person





6.6 Known person API notification

7. Conclusion

The system will offer better functionalities than the manual system, where the users of the system will have to be computer/mobile literate so as to use the system effective
during the period of project, I came to know the practical aspects of Internet of Things development of that sound our knowledge in Python Programming, API interfaces and OpenCV.
Python, OpenCV and Flask are brand new technology and I will improve my skill in these technology fields.

8. Bibliography

Web Resources

www.stackoverflow.com

www.raspberrypi.org/docs

www.w3schools.com

www.hackster.io

www.instructables.com

msdn2.microsoft.com

http://www.instructables.com/id/PIR-Motion-Sensor-Tutorial/

https://www.hackster.io/pidoorbell-team/pidoorbell-7ef917

https://www.raspberrypi.org/learning/parent-detector/worksheet/

Books

“Internet Of Things : A Hands of Approach” – Vijay Madisetti, ArshdeepBagha

“Fundamentals of Wireless Networks:Theory and Practice”-WaltenegeusDargie, Christian Poellabauer

9. Experience, Limitations and Future Enhancement

9.1 Experience during project development

During the project development, we learn some new technologies and also explored variety of things and also learn various aspects of software development.

9.2 Limitations

Following are the limitations of IoT platform: -

- There needs to be an internet connectivity to access the application without internet connectivity the application cannot be accessed.
- In order to recognize face, system needs be to set properly so face can be captured, and there should be ambient illumination so that face could be recognized.

9.3 Future Enhancement

The future enhancements would be that there, could be multiple users accessing the same user's notification panel, The User would be able to also watch a live stream of the surroundings outside his door. Furthermore, adding to the same we can also provide image search on global database where in not only we can click the picture of any unknown person but would be able to search it over the internet for the same, and also modify the code to recognize multiple persons at a time.