VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590018

A PROJECT REPORT **(15CSP85)** ON

"IOT Based Smart Luggage System"

Submitted in Partial fulfillment of the Requirements for the Degree of

Bachelor of Engineering in Computer Science & Engineering

By

KEERTHI PRIYA S V (1CR16CS070)

LIKITHA D (1CR16CS075)

NAVYA M (1CR16CS102)

R MANJULA SRIDEVI (1CR16CS129)

Under the Guidance of,

SHERLY NOEL

Assistant Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **"IOT Based Smart Luggage System"** carried out by **Ms. KEERTHI PRIYA S V**, USN **1CR15CS070**, **Ms. LIKITHA D**, USN **1CR15CS075**, **Ms. NAVYA M**, USN **1CR15CS102, Ms. R MANJULA SRIDEVI**, USN **1CR15CS129,** bonafide students of CMR Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering** in Computer Science and Engineering of the Visveswaraiah Technological University, Belgaum during the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

.

Sherly Noel	**Dr. Prem Kumar Ramesh**	**Dr. Sanjay Jain**
Assistant Professor	**Professor & Head**	**Principal**
Dept. of CSE, CMRIT	**Dept. of CSE, CMRIT**	**CMRIT**

External Viva

Name of the examiners — Signature with date

1. _____

2. _____

DECLARATION

We, the students of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the work entitled "**IOT Based Smart Luggage System**" has been successfully completed under the guidance of Assistant Prof. Sherly Noel, Computer Science and Engineering Department, CMR Institute of technology, Bangalore. This dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2019 - 2020. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place: Bangalore

Date: 10/06/2020

Team members:

KEERTHI PRIYA S V (1CR16CS070) _____

LIKITHA D (1CR16CS075) _____

NAVYA M (1CR16CS102) _____

R MANJULA SRIDEVI (1CR16CS129) _____

ABSTRACT

The Internet of Things is a concept that not only has the potential to impact how we live but also how we work. Objective of proposal is to provide a system for permanent luggage labelling through face recognition to identifying the owner of luggage as well as tracking luggage throughout the airport, taking into account security requirement, including privacy, authentication, and data integrity. The idea of the project is to propose an efficient way to automate the way baggage is handled in airports/other places. The proposed architecture focuses more on the face detection using machine learning along QR code based tracking of the passenger baggage. In this project IoT components are being used like Raspberry pi and a QR code. User can track their luggage by using QR code scanning along with face recognition.

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out my final year project

I have a great pleasure in expressing my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

I would like to thank **Dr. Prem Kumar Ramesh,** Professor and Head, Department of Computer Science and Engineering, CMRIT, Bangalore, who has been a constant support and encouragement throughout the course of this project.

I consider it a privilege and honor to express my sincere gratitude to my guide **Sherly Noel, Assistant Professor,** Department of Computer Science and Engineering, for the valuable guidance throughout the tenure of this review.

I also extend my thanks to all the faculty of Computer Science and Engineering who directly or indirectly encouraged me.

Finally, I would like to thank my parents and friends for all their moral support they have given me during the completion of this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The Internet of things (IoT) is the internetworking of physical devices, vehicles (also referred to as "connected devices" and "smart devices"), buildings, and other items—embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data.

Misplacement of luggage can happen with anyone irrespective of the circumstances and conditions. It is very necessary to track down the bags in case of loss and theft. The Internet of Things (IoT) is a network of objects like buildings, vehicles, etc. which is embedded with sensors, electronics and other networks related things which helps these objects to collect and exchange information. IOT allows the objects to be sensed and controlled from a remote access point, which does the integration of computers with the physical world which improves financial benefit, accuracy and efficiency. When IoT is connected with sensors, it encompasses technologies such as smart homes, vehicle tracking, monitoring and controlling home appliances and ultimately smart cities. Everything is unique and identifiable and moreover becomes easy accessible when it is augmented with the Internet and computer infrastructure.

1.1 Relevance of the Project

- The luggage Tracking System works on an alarm basis with a GPS module.
- A map is created through which we can track the location of the bag as it moves.
- An S3 luggage bag is application specific design that can be useful for everyone.
- Eigen Face was used as the feature extraction, while Principal Component Analysis (PCA) was used as the classifier.
- The design of baggage tracing and handling systems using Radio Frequency Identification (RFID) & IoT.

1.2 Proposed System

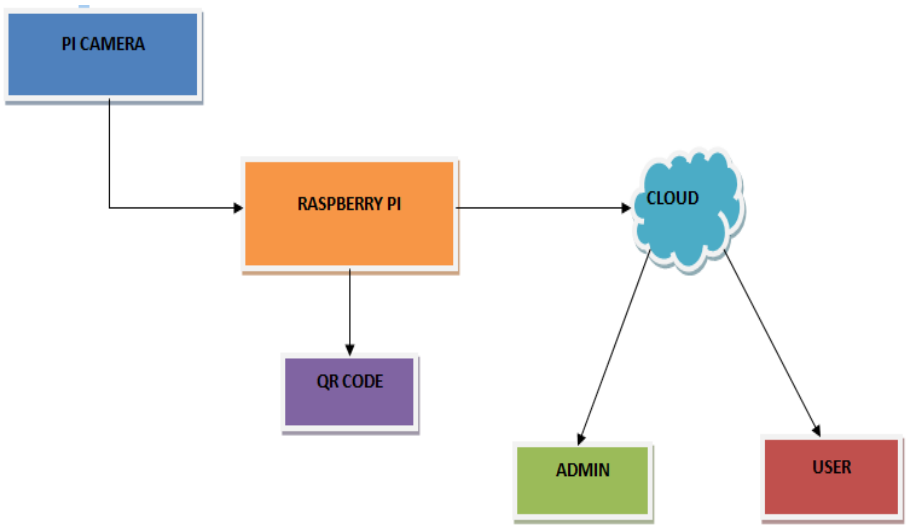

Fig1.1: Block Diagram of Tracking System

1.3 Agile Methodology

Story ID	Requirements Description	User Stories/Task	Description
1	Collection of Faces and Training the faces	Faces Collection	.py format and .jsp format
2	Face recognition Analysis	Pre-processing	Feature extraction
3	Algorithm	Model Creation	Face Detection and Face Recognition.
4	QR Code Authentication	Model creation	QR code matching
5	Training the model and Testing	Training and Testing	Training faces required
6	Improvements if Required	Bugs	

Table 1.1: Agile Methodology

1.4 Objectives

- The main objective of our proposal is to provide a system for permanent luggage labelling through face recognition to identify the owner of luggage as

well as tracking luggage throughout the airport, taking into account security requirements, including privacy, authentication, and data integrity.

- To recognize the face through scanning the person's face, first we need to detect whether the face is scanned using QR code.

1.5 Scope of the Project

- The main scope of the Luggage tracking System using IoT Project is to develop a luggage that could be user-friendly.
- The project is more efficient to use the face recognition using machine learning technique along with QR code.
- The system provides significant improvement in communication through cloud between admin and user.
- This will improve passenger security and satisfaction as well as reducing delays insights caused by mishandled luggage.

1.6 Methodology

- Users can track their luggage by using QR code scanning along with face recognition.
- The face recognition system is capable of identifying or verifying a person and allows them to scan QR code scanning.
- To recognize the face through scanning the person's face, first we need to detect whether the face is scanned using QR code.
- Raspberry pi camera is used for capturing the face image.
- Haar cascade algorithm is used for face detection using QR code.
- Raspberry pi is used as a controller and temporary server before sending information to the cloud.
- Cloud is a server used to store face related information permanently.

1.7 Chapter Wise Summary

Here is a chapter wise summary of this report.

- **Chapter 2: Literature Survey**

  We reviewed 4 research papers and compared them to find the best approach for going about our project

- **Chapter 3: System Requirements Specification**

  This chapter contains all the hardware and software requirements

- **Chapter 4: System Analysis and Design**

  In this chapter we described both the approaches for preparing our model i.e. the Haar cascade Algorithm and LBPH face recognition.

- **Chapter 5: Implementation**

  This chapter contains how the detailed implementation and phases.

- **Chapter 6: Results and Discussion**

  This chapter contains the results we observed and obtained from the project.

- **Chapter 7: Testing**

  This chapter contains the various testing methods used to test the project.

- **Chapter 8: Conclusion**

  This chapter contains the various conclusions drawn from our work

CHAPTER 2

LITERATURE SURVEY

There are various papers published on prediction of breast cancer using various methods. We have reviewed four papers and have drawn the following conclusions.

[1] Towards Smart Wearable Real-time Airport Luggage Tracking

In this paper, we propose a smart system for real-time tracking of airport luggage mobile applications and smart watches. Communication in airports for tracking the luggage is commonly used are Radio Frequency Identification based technology. Bluetooth can also be used to track the luggage .The less power consuming Bluetooth device is called ibeacon. Smart system with Kalman-filtered Wi-Fi fingerprinting for real-time tracking of luggage so that we can scan QR codes to get details of flight and to associate with each luggage with the smart tag. The owner will receive the notifications regarding the status of luggage and also the tag to recharge in case of low battery in both mobile applications and smart watches. Finite State machines are smart tags and the server. The smart tag sends the server its current GPS location for matching the final destination. The power management scheme which prevents the smart tag from depleting its battery before arrival to destination. The flight information is given to the owner for his reference to track the luggage. The modes of the operations are Flight, Landed, Arrival and Deep Sleep modes. The figure (Fig2.1) given below illustrate the flow chart of this paper.

Features of the paper:

- Finite State Machine
- Smart tag
- Smart mobile/watches applications
- Power management

Advantage: - The luggage includes a GPS module or chip and which can be tracked anywhere in the world. Even the misplaced luggage can be easily tracked. The low powered Wi-Fi modules, a GPS module and sensors are used in the system.

Disadvantage - Bluetooth typically only works within the certain range from your phone.

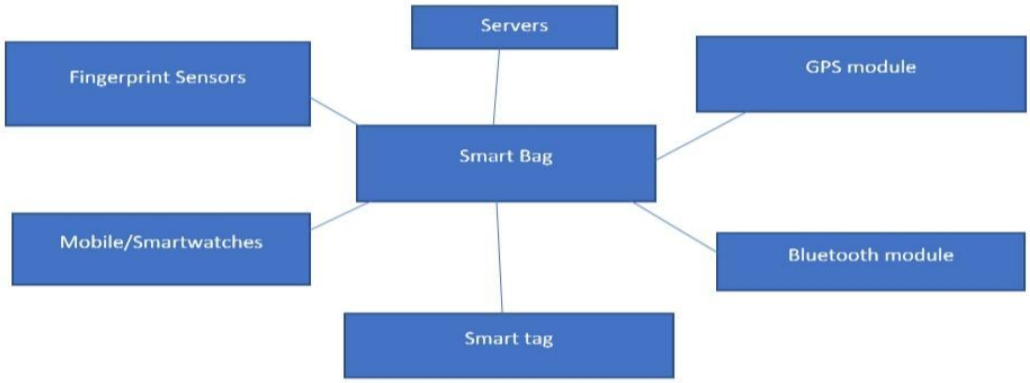

Fig2.1: Flow Chart

[2] Real Time Airport Luggage Tracking System

This paper is associated with real time baggage tracking using microcontroller, GSM and GPS. GSM module- GSM sim 900 is used to send text messages to the user and it is also used to communicate with the user's mobile. GPS module- GPS module continuously sends data as a string of characters which should be decoded to the standard format. GPS module gives coordinates of the luggage. Microcontroller-microcontroller receives a string of coded data from the GPS module, which is decoded to the standard format within the microcontroller. The microcontroller then sends the required data along with instructions to the GSM. User accesses the data through the user's mobile using an application to gather data and coordinates of their luggage from the server. Data from the GSM is stored in a database in the server. The figure (Fig2.2) given below illustrate the use case diagram of the proposed system of this paper.

Features of this paper are:-

- ARDUINO Controller UNO
- GSM Module
- GPS Module
- PC side server
- Smart application for user

Advantage through this system is to minimize time utilized and to give access to people to navigate their baggage efficiently through the user's mobile or smart wearables with an application.

Drawbacks-The RFID scanner technology used in the project works with low frequency range and limiting the range and type of the ID's on the luggage.

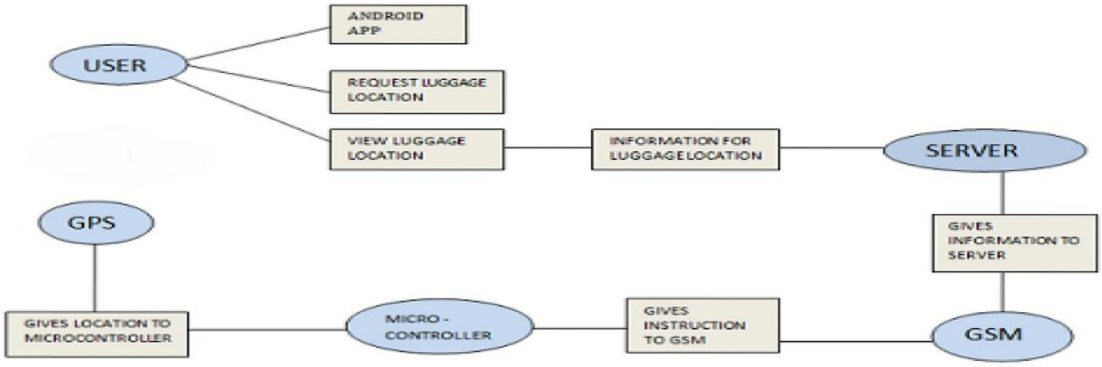

Fig2.2: Use Case Diagram

[3] A Smart System to Minimize Mishandled Luggage at airports

The system developed in this paper is to link the luggage to the passenger using RFID. Passenger luggage will be linked with the flight tickets using the RFID tags. RFID tags can be monitored using RFID readers. The system developed will check, compare and save the data that is present in the tag into the database each time the passenger passes through the exit gate. It will recognize whether the passengers are carrying their own associated luggage. The exit gate will allow the passengers to exit only if they have the luggage that is associated with them. The figure (Fig2.3) given below illustrate the system block diagram of this paper.

Design and implementation of this system include:

- Generating RFID tag
- Stamping RFID tags
- Smart exit gate
- Mobile application features

Advantage

The system developed is smart, low cost and secure to handle and minimize the rate of loss in the airports. It will reduce the number of luggage misplacements and reduces the time of passengers to find their luggage.

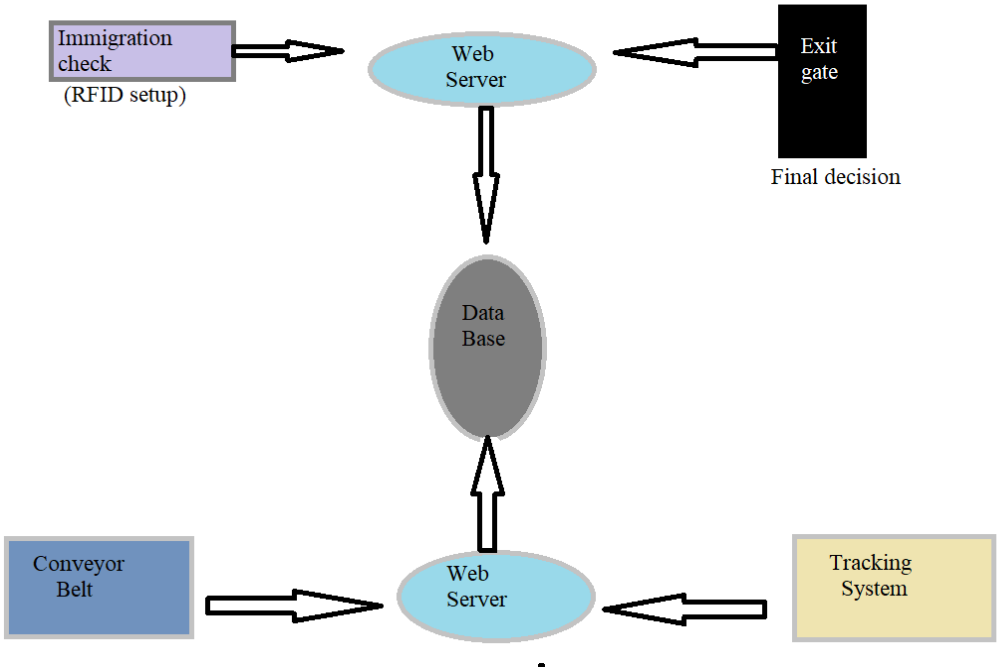

Fig2.3: System Block Diagram

[4] Smart Luggage

The main idea of the project is to develop a luggage that could be user friendly. The project is more of a luggage and less of a robot. There are a lot of applications to the luggage but all of them are not controlled from the luggage, instead the commands are sent from the mobile phone to the luggage via machine to machine communication. The mobile phone has pre pre-installed Application with pre pre-installed set of instructions. They wait for the user to send the commands. After the microcontroller embedded inside the luggage receives instruction from the user it acts accordingly. This can either be for tracking its location and sending it to the user or sending the luggage weight. A GPS module is used to track the location of the luggage. The figure (Fig2.4)

given below illustrate the basic flow of this paper

Advantages:-

- Ensure safety and build security to the user
- Easy to implement

Disadvantage:-

If developed under AI, the application of the luggage will expand and it will search for new horizons to explore.

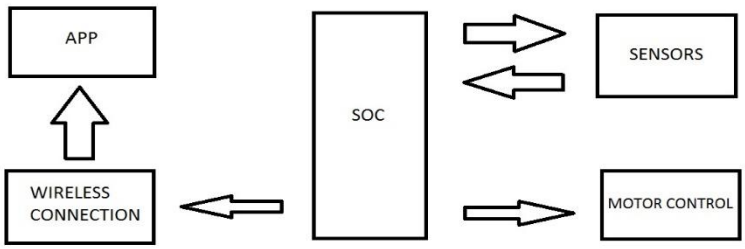

Fig2.4: Basic Block Diagram

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

The chapter describes about the requirements. It specifies the hardware and software requirements that are in order to run the application properly. The Software Requirement Specification (SRS) is explained in detail, which includes the overview of functional and non-functional requirements.

An SRS document describes the functional and behavioural requirements of the software under development. SRS is a fundamental document which forms the foundation of the software development process. It is the complete description of the behaviour of a system to be developed it is not only the list of requirements of a system but also has a description of its major features. Requirement Analysis in system engineering and software engineering encompasses those tasks that go into determining the need or conditions to meet for a new or altered product.

The SRS functions as a blueprint for completing a project. It is important to note that an SRS contains functional and non- functional requirements.

Thus, the goal of preparing the SRS document is to

• To firm foundation for the design phase

• Support system testing facilities

• Support project management and control

• Controlling the Evolution of system

3.1 Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs/or conditions. These may include calculations, data manipulation and processing and other specific functionality. In these systems following are the functional requirements

• Input test case must not have compilation and runtime errors.

• The application must not stop working when kept running for even a long time.

• The application should generate the output for a given input test case.

3.2 Non-Functional Requirements

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours.

Given below are the non-functional requirements:

• Product requirements

• Organizational requirements

• Basic operational requirements

3.3 Hardware Requirements

- Pi Camera
- Raspberry pi board

3.4 Software Requirements

- Raspbian OS
- Windows OS
- Apache tomcat server

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 DESIGN MODULES

1. PI CAMERA

The pi camera is used to capture the images. In this project first we will create the database and store the user face images in the database. Once the database is created, then we run the project and pi camera turns on. Pi camera start capturing the images and compare the image with the database images and if the image match is found, the output window display the user id number. Here image matching is done in 2 steps: Face detection: It can be done with Haar cascade, Face recognition: This can be done using LBPH recognizer. The Raspberry Pi Camera v2 is the new official camera board released by the Raspberry Pi Foundation. The Raspberry Pi Camera Module v2 is a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for the Raspberry Pi, featuring a fixed focus lens. The Raspberry Pi Zero now comes complete with a camera port! Using the new Raspberry Pi Zero Camera Adapter, you can now use a Raspberry Pi camera with your Zero. It's capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video. It attaches to the Raspberry Pi via one of the small sockets on the board's upper surface and uses the dedicated CSi interface, designed especially for interfacing to cameras.

Features: Fixed focus lens on-board, 8 megapixel native resolution sensor-capable of 3280 x 2464 pixel static images, Supports 1080p30, 720p60 and 640x480p90 video, Size 25mm x 23mm x 9mm, Weight just over 3g, Connects to the Raspberry Pi board via a short ribbon cable (supplied), Camera v2 is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system

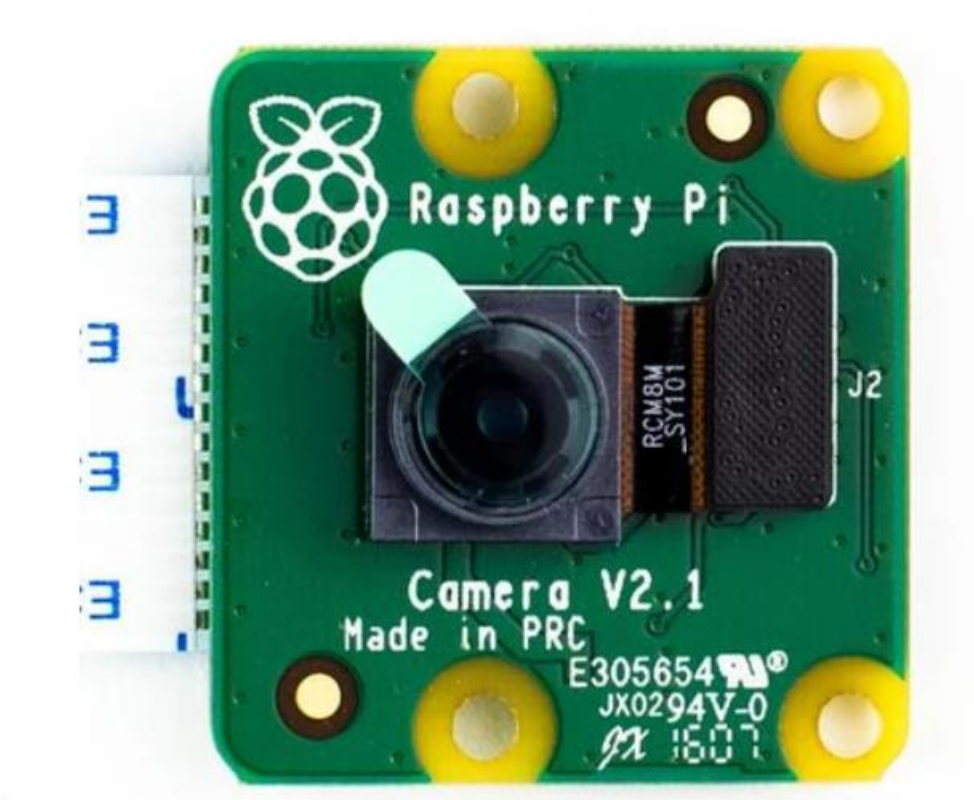

Fig4.1: Pi Camera

2. RASPBERRY PI

A face recognition security system has been proposed using raspberry pi which can connect to a pi camera. Raspberry pi controls and monitors the entire system. User can track the luggage by using QR code scanning along with face recognition. The face recognition system is capable of identifying or verifying a person and then allows scan QR code scanning. The user and admin information is stored in the cloud. The Raspberry pi is a single computer board with credit card size that can be used for many tasks that your computer does, like games, word processing, spreadsheets and also to play HD video. It was established by the Raspberry pi foundation from the UK. It has been ready for public consumption since 2012 with the idea of making a low-cost educational microcomputer for students and children. The main purpose of designing the raspberry pi board is, to encourage learning, experimentation and innovation for school level students. The raspberry pi board is a portable and low cost. Maximum of the raspberry pi computers is used in mobile phones. In the 20th century, the growth of mobile computing technologies is very high, a huge segment of this being driven by

the mobile industries. The 98% of the mobile phones were using ARM technology. The Raspberry Pi is a Broadcom BCM2835 SOC (system on chip board). It comes equipped with a 700 MHz, 512 MB of SDRAM and ARM1176JZF-S core CPU. The USB 2.0 port of the raspberry pi boars uses only external data connectivity options. The Ethernet in the raspberry pi is the main gateway to interconnect with other devices and the internet in model B. This draws its power from a micro USB adapter, with a minimum range of 2.5 watts (500 MA). The graphics, specialized chip is designed to speed up the manipulation of image calculations. This is in built with Broadcom video core IV cable that is useful if you want to run a game and video through your raspberry pi. Features of Raspberry PI Model B

- 512 MB SDRAM memory
- Broadcom BCM2835 SoC full high definition multimedia processor
- Dual Core Video Core IV Multimedia coprocessor
- Single 2.0 USB connector
- HDMI (rev 1.3 and 1.4) Composite RCA (PAL & NTSC) Video Out
- 3.5 MM Jack, HDMI Audio Out
- MMC, SD, SDIO Card slot on board storage
- Linux Operating system
- Dimensions are 8.6cm*5.4cm*1.7cm
- On board 10/100 Ethernet RJ45 jack

Fig4.2: Raspberry pi Board

3. QR Code

The QR code is a two-dimensional version of the barcode. QR stands for “Quick Response”, which refers to the immediate access to the information hidden in the code. QR codes are gaining popularity because of the open source i.e. easily available for everyone. Advantages of the QR code is large data capacity and high fault tolerance. Personalized QR codes can be generated using the QR code generator on the web.

Fig4.3: QR code

4.2 FLOW CHART

There will be two systems used in the project, one for each Raspberry Pi Board and the server. The two systems are connected through URL and IP address. The user at the boarding point will be asked for their face recognition and the system will store the data along with the generated QR code. The generated QR code will be sent to the user. The generated QR code will contain the essential information of the user. While at the receiving point, the user will be asked to scan the user's face for the face detection by comparing data stored in the server. Face detection is a two layered security process where it uses Haar Cascade and LBPH.

Haar Cascade algorithm- The input image is processed in two ways : ‘+ image’(containing image region with face only) and ‘- image’(containing image region without face).

LBPH- LBPH algorithm is a part of opencv. It is based on a local binary operator, where the facial image is stored in a binary pattern as a graphical representation called histogram. When the user’s face image is matched, he will be asked to produce the QR code received by the user. Once the QR code is matched the server

sends a binary output saying 1 or 0. The system will finally display all the stored details of the user and the baggage.

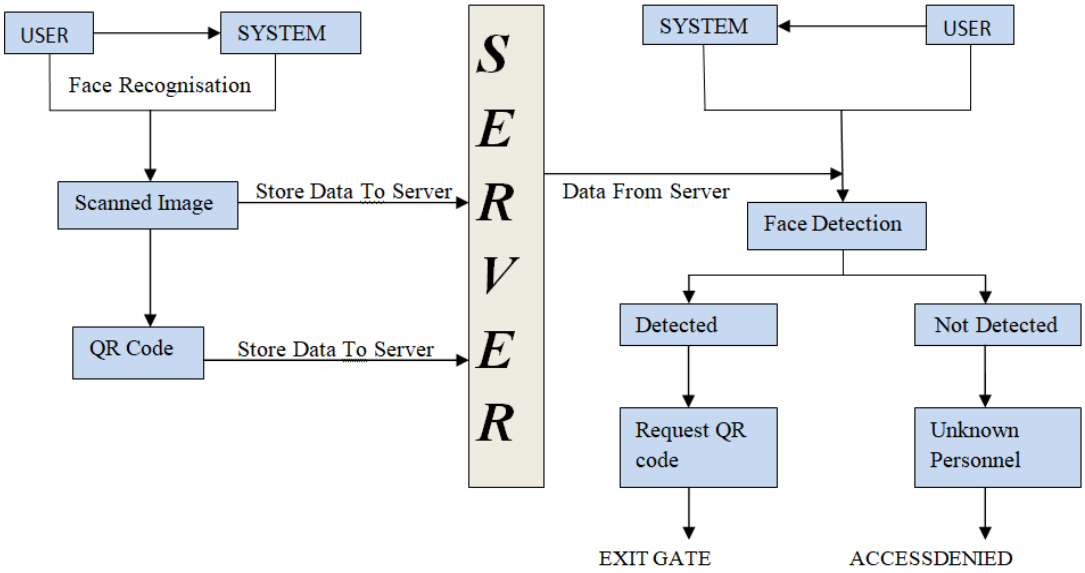

Fig4.4: System Architecture

4.3 ALGORITHM

Haar Cascade Algorithm

Haar Cascade is a machine learning object detection algorithm used to identify objects in an image or video. Object detection using Haar feature-based cascade classifier is an effective object detection method proposed. It is a machine learning based approach where a cascade function is trained from a lot of positive images (images of face) and negative images (images without face) to train the classifier. Then features are extracted from it. For this, Haar features shown below images are used. Each feature is a single value obtained by subtracting the sum of pixels under white rectangle from the sum of pixels under black rectangle.

Haar Cascade detection in OpenCV

- OpenCV comes with both a trainer as well as a detector.
- If you want to train your own classifier for any objects like cars, planes, etc. You can use OpenCV to create one.
- Here we deal with face detection.

- OpenCV already contains many pre-trained classifiers for face, eye, smile, etc. Those XML files are stored in opencv/data/haarcascades/ folder.

Algorithm

- Pick f (maximum acceptable false positive rate per layer) and d (minimum acceptable detection rate per layer)
- Let Ftarget is target overall false positive rate
- Let P is a set of positive examples
- Let N is a set of negative examples
- Let Fo =1, Do=1, and i=0 (Fo: overall false positive rate at layer 0, Do: acceptable detection rate at layer 0, and i: is the current layer)
- While Fi>Ftarget (Fi: overall false positive rate at layer i):
    - i++ (layer increasing by 1)
    - ni = 0; Fi = Fi-1 (ni :negative examples i):
    - While Fi >f*Fi-1 :
        - Ni++ (check a next negative example)
        - Use P and N to train with AdaBoost to make a xml (classifier)
        - Check the result of a new classifier for Fi and Do
        - Decrease threshold for new classifier to adjust detection rate r>=d*Fi-1
    - N = empty
    - If Fi >Ftarget , use the current classifier and false detection to set N

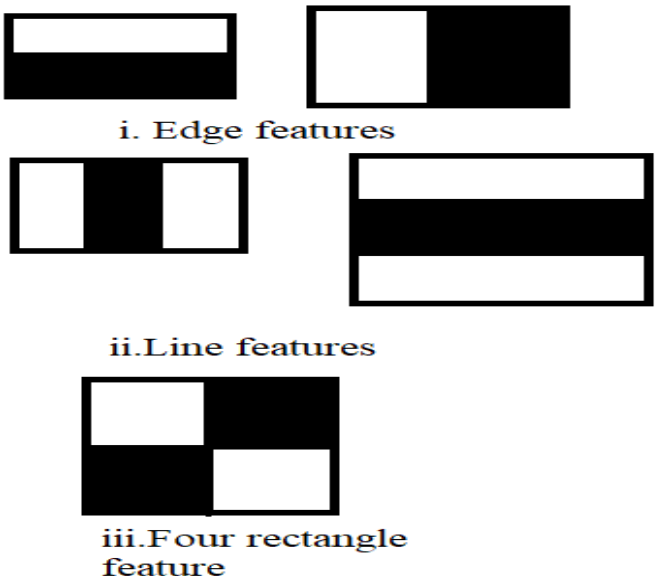

Fig4.5: Haar Cascade Classifier

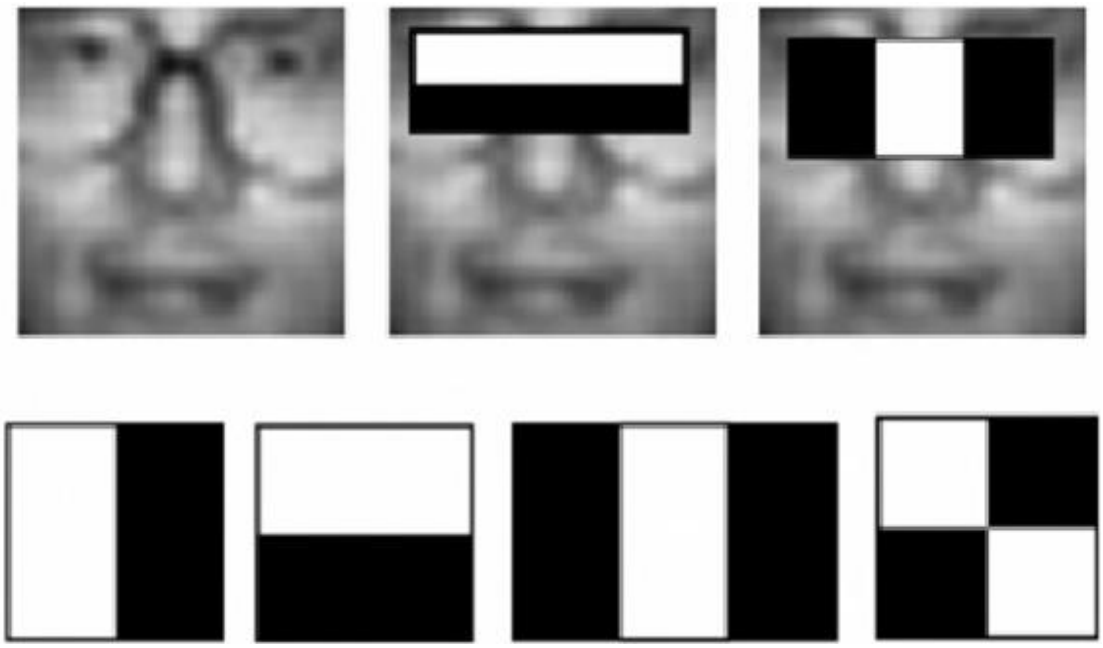

Fig4.6: Object detection using Haar Cascade

Training the Algorithm: First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. We need to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output. Images of the same person must have the same ID. With the training set already constructed, let's see the LBPH computational steps.

Applying the LBP operation: The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters **radius** and **neighbours**.The below image shows the procedure

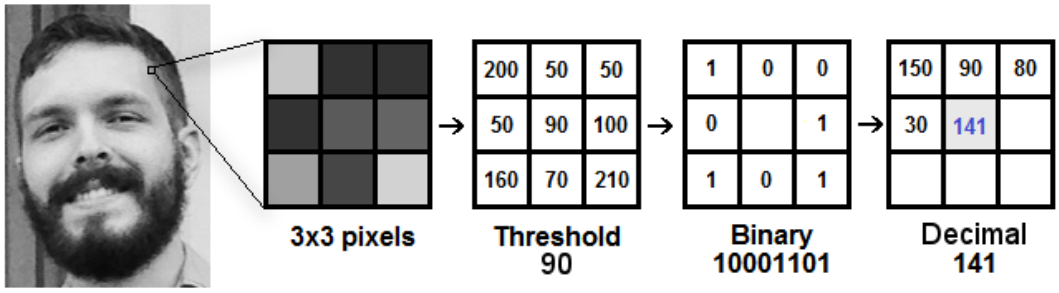

Fig4.7: Procedure Representation

CHAPTER 5

IMPLEMENTATION

5.1 Code

To capture User image and to store in Database

```
from picamera import PiCamera
import time
camera=PiCamera()
print('Camera Start to Capture....')
v=1
for i in range(10):
time.sleep(0.5)
camera.capture('/home/pi/Desktop/luggage/userdb/user%s.jpg' % v)
v=v+1
print('Scan Complete')
```

To capture ADMIN image and to store in Database

```
from picamera import PiCamera
 import time
camera=PiCamera()
print('Camera Start to Capture....')
v=1
for i in range(10):
time.sleep(0.5)
camera.capture('/home/pi/Desktop/luggage/admindb/admin%s.jpg%v)
v=v+1
print('Scan Complete')
```

Code for face detection and face recognition and QR code generation

```
import cv2, os
import picamera
import numpy as np
from PIL import Image
from time import sleep
from PIL import Image, ImageEnhance
import RPi.GPIO as GPIO
import requests
import time
import serial
import string
import urllib2
from sys import argv
import zbar
import time
GPIO.SETMODE(GPIO.BOARD)
GPIO.SETWARNINGS(FALSE)

M1=37
GPIO.SETUP(M1,GPIO.OUT)
GPIO.OUTPUT(M1,FALSE)

SER = SERIAL.SERIAL(
PORT='/DEV/TTYAMA0',
BAUDRATE = 9600,
PARITY=SERIAL.PARITY_NONE,
STOPBITS=SERIAL.STOPBITS_ONE,
BYTESIZE=SERIAL.EIGHTBITS,
TIMEOUT=1
 )
COUNTER=0
```

```
        COUNT=0
        B=0
        L=1
       # FOR FACE DETECTION WE WILL USE THE HAAR CASCADE
         PROVIDED BY OPENCV.
         CASCADEPATH                                                    =
"HAARCASCADE_FRONTALFACE_DEFAULT.XML"
         FACECASCADE = CV2.CASCADECLASSIFIER(CASCADEPATH)
        # FOR FACE RECOGNITION WE WILL THE THE LBPH FACE
         RECOGNIZER
         RECOGNIZER
```

```
    FOR FACE DETECTION WE WILL USE THE HAAR CASCADE PROVIDED
BY OPENCV.
CASCADEPATH = "HAARCASCADE_FRONTALFACE_DEFAULT.XML"
FACECASCADE = CV2.CASCADECLASSIFIER(CASCADEPATH)
# FOR FACE RECOGNITION WE WILL THE THE LBPH FACE RECOGNIZER
RECOGNIZER = CV2.CREATELBPHFACERECOGNIZER()
################################################################
PROC = ZBAR.PROCESSOR()
PROC.PARSE_CONFIG('ENABLE')
DEVICE = '/DEV/VIDEO0'
IF LEN(ARGV) > 1:
   DEVICE = ARGV[1]
PROC.INIT(DEVICE)
DEF QR():
   PROC.VISIBLE = TRUE
   PROC.PROCESS_ONE()
   PROC.VISIBLE = FALSE
   FOR SYMBOL IN PROC.RESULTS:
     #PRINT 'DECODED', SYMBOL.TYPE, 'SYMBOL', '"%S"' %
SYMBOL.DATA,
```

```
        X=SYMBOL.DATA
        U=X[1:20]
        PRINT 'LUGGAGE:'+U
        RETURN U
DEF ID1():
   IF(PREDICTED>=1 AND PREDICTED<=10):
            PRINT('ID1')
            PRINT ('SHOW U LUGGAGE QR CODE')
            TIME.SLEEP(5)
            U=QR()
            P1=1
            TIME.SLEEP(1)
            PRINT P1
            RETURN P1
DEF ID2():
   IF(PREDICTED>=11 AND PREDICTED<=20):
            PRINT('ID2 ')
            PRINT ('SHOW U LUGGAGE QR CODE')
            TIME.SLEEP(5)
            U=QR()
            P2=1
            TIME.SLEEP(1)
            PRINT P2
            RETURN P2
DEF ID3():
   IF(PREDICTED>=21 AND PREDICTED<=30):
            PRINT('ID3')
            PRINT ('SHOW U LUGGAGE QR CODE')
            TIME.SLEEP(5)
            U=QR()
            P3=1
            TIME.SLEEP(1)
```

```
        PRINT P3
        DEF GET_IMAGES_AND_LABELS(PATH):
IMAGE_PATHS = [OS.PATH.JOIN(PATH, F) FOR F IN OS.LISTDIR(PATH) IF NOT F.ENDSWITH('.')]
IMAGES = []
 LABELS = []
  VAR = 0
  I=1
  FOR IMAGE_PATH IN IMAGE_PATHS:
    # READ THE IMAGE AND CONVERT TO GRAYSCALE
MAGE_PATH=('/HOME/PI/DESKTOP/SMART LUGGAGE/DB/STUDENT%S.JPG' % I)
     I=I+1
     PRINT(IMAGE_PATH)
     IMAGE_PIL = CV2.IMREAD(IMAGE_PATH)
     GRAY = CV2.CVTCOLOR(IMAGE_PIL,CV2.COLOR_BGR2GRAY)
     CV2.CVTCOLOR(PREDICT_IMAGE_PIL,CV2.COLOR_BGR2GRAY)
     PREDICT_IMAGE = NP.ARRAY(GRAY, 'UINT8')
     #PREDICT_IMAGE = NP.ARRAY(PREDICT_IMAGE_PIL)
     PREDICT_IMAGE =CV2.RESIZE(PREDICT_IMAGE,(128,128))
     # PREDICT_IMAGE = IMAGEENHANCE.SHARPNESS(PREDICT_IMAGE)
     FACES = FACECASCADE.DETECTMULTISCALE(PREDICT_IMAGE)
     FOR (X, Y, W, H) IN FACES:
        PREDICTED,CONF= RECOGNIZER.PREDICT(PREDICT_IMAGE[Y:Y+H,X:X+W])
        CV2.IMSHOW("ADDING FACES TO TRANING SET...", PREDICT_IMAGE[Y:Y+H,X:X+W])
        CV2.WAITKEY(50)
     #PRINT(PREDICTED, CONF)
     P1=ID1()
     P2=ID2()
    IF(U=="12345" AND P1==1):
```

```
    PRINT('UPLOAD TO SERVER')
IF(U=="23451" AND P2==1):
    PRINT('UPLOAD TO SERVER')
IF(U=="34512" AND P3==1):
    PRINT('UPLOAD TO SERVER')
ELSE:
    PRINT("UNKNOWN PERSON")
```

CHAPTER 6

RESULTS AND DISCUSSION

1. This is the home page that is displayed when the server side of the project is executed. Using this page admin and the user can login to their respective pages.

Fig6.1: Home Page

2. This is the page that shall be displayed when the admin logs into his page. Only new user request will be sent to the admin's user requests. In this page admin can control the user's login by accepting or rejecting the request by clicking on the user request that is been displayed on this screen

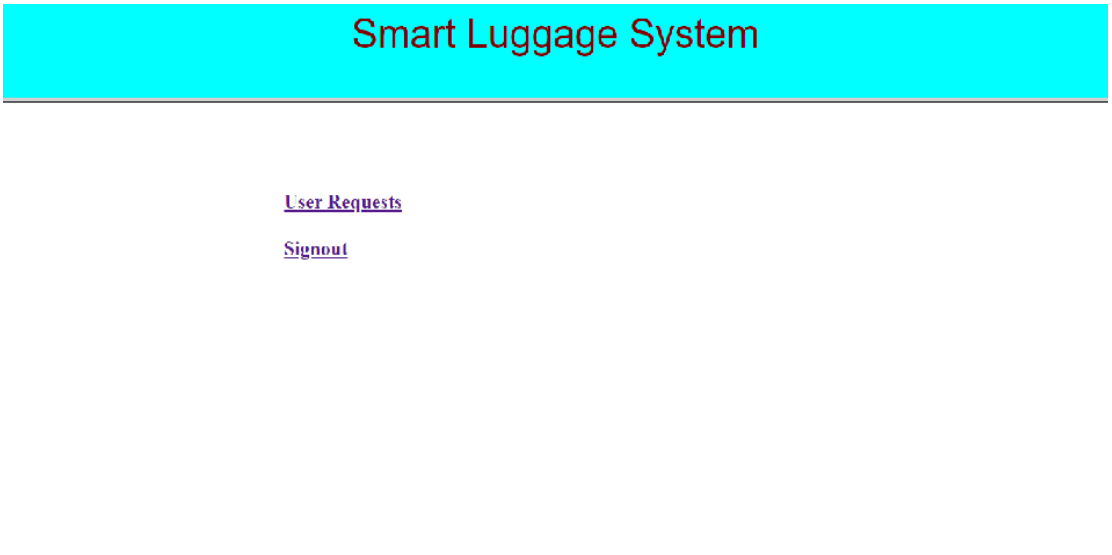

Fig6.2: Admin Page

3. This page will be displayed when the user logs into his respective account. The user can view and update his personal information by clicking on view profile. The information that is stored in the users QR code is transmitted to the view luggage information through IP address and this information can be viewed by the user.

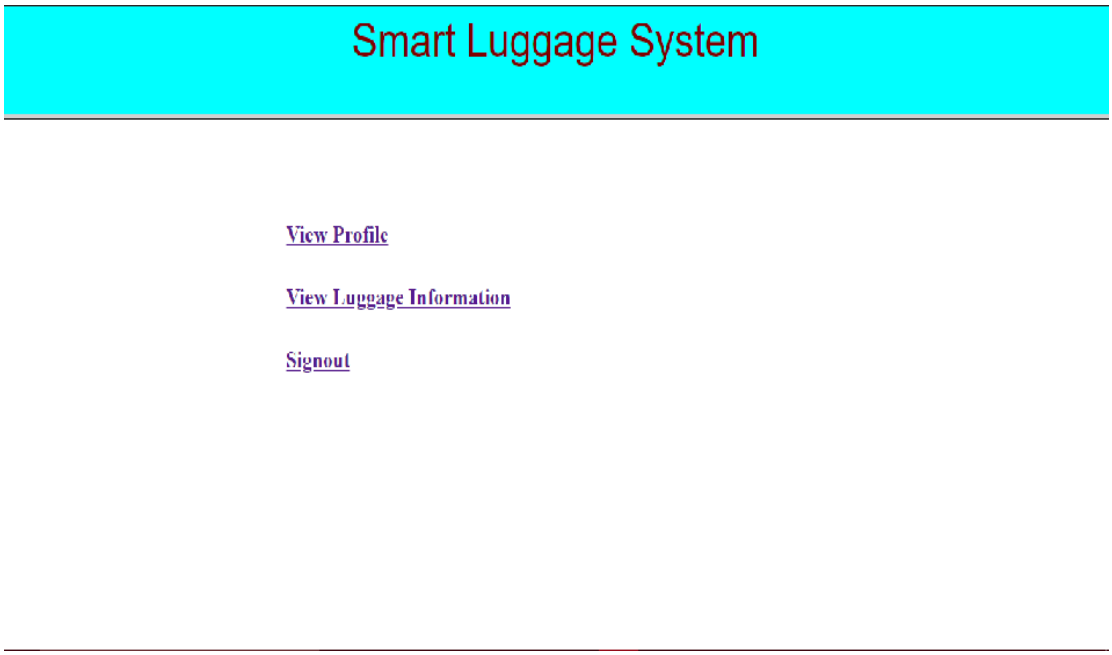

Fig6.3: User Page

4. This is the format in which the information is stored inside view luggage information though IP address sent by the hardware component raspberry pi system.

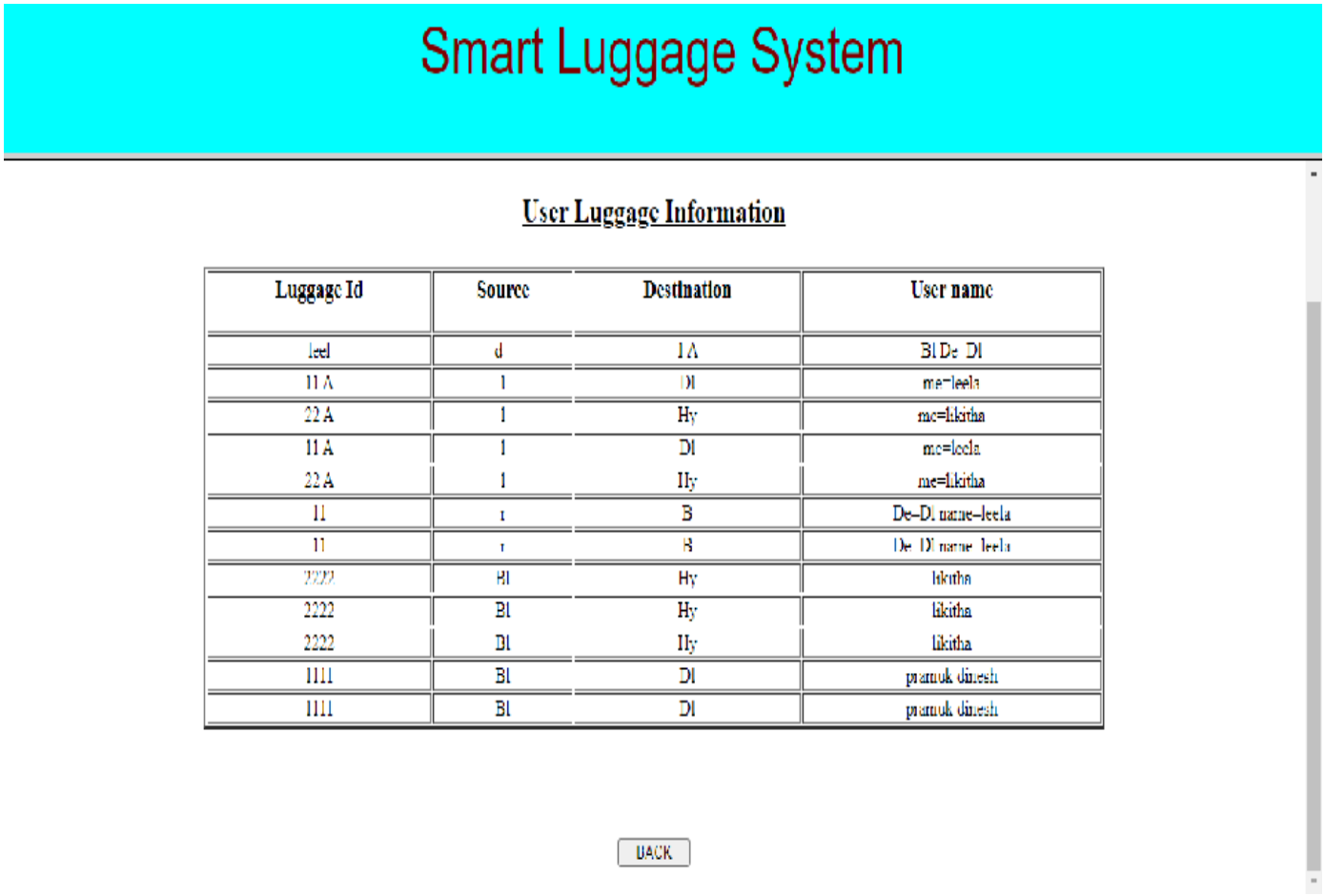

Fig6.4: Information Stored using URL

5. This is the dataset of the user face images that is captured by the Pi camera. Pi camera captures 10 images of each user at a time and stores it in user's data base. These images are used for face detection and recognition.

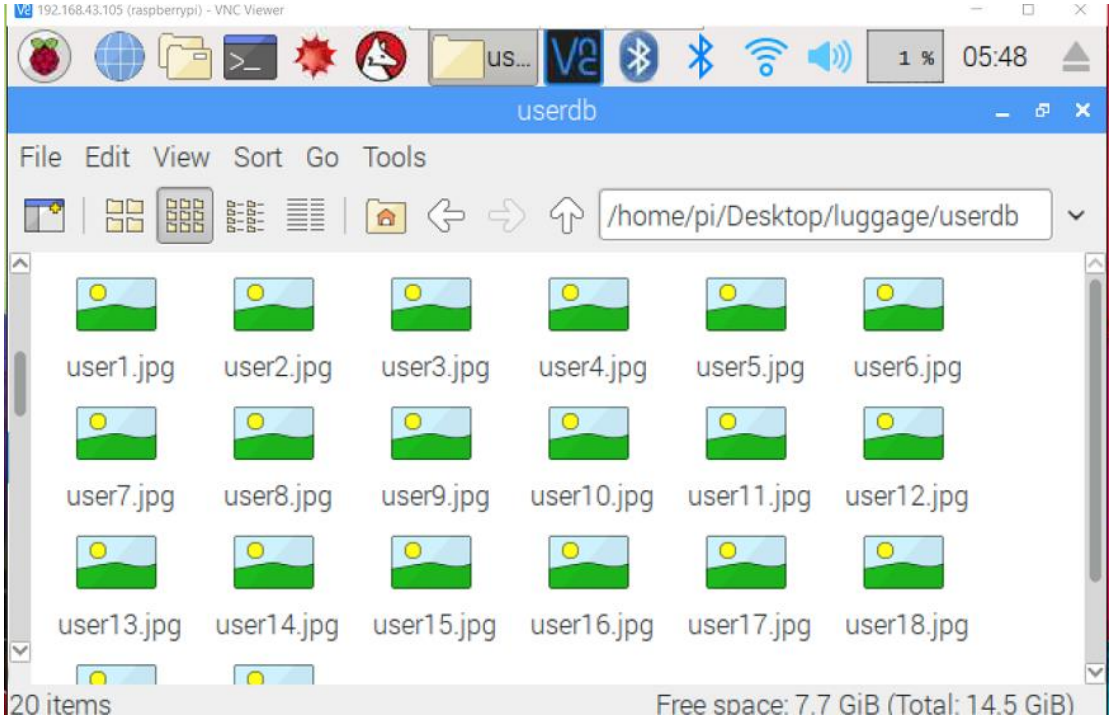

Fig6.5: User Image Captured and Stored in Database

6. These are the images captured by the pi camera and stored in admin database that are used to recognise the admin's face. It also captures 10 images of each admin

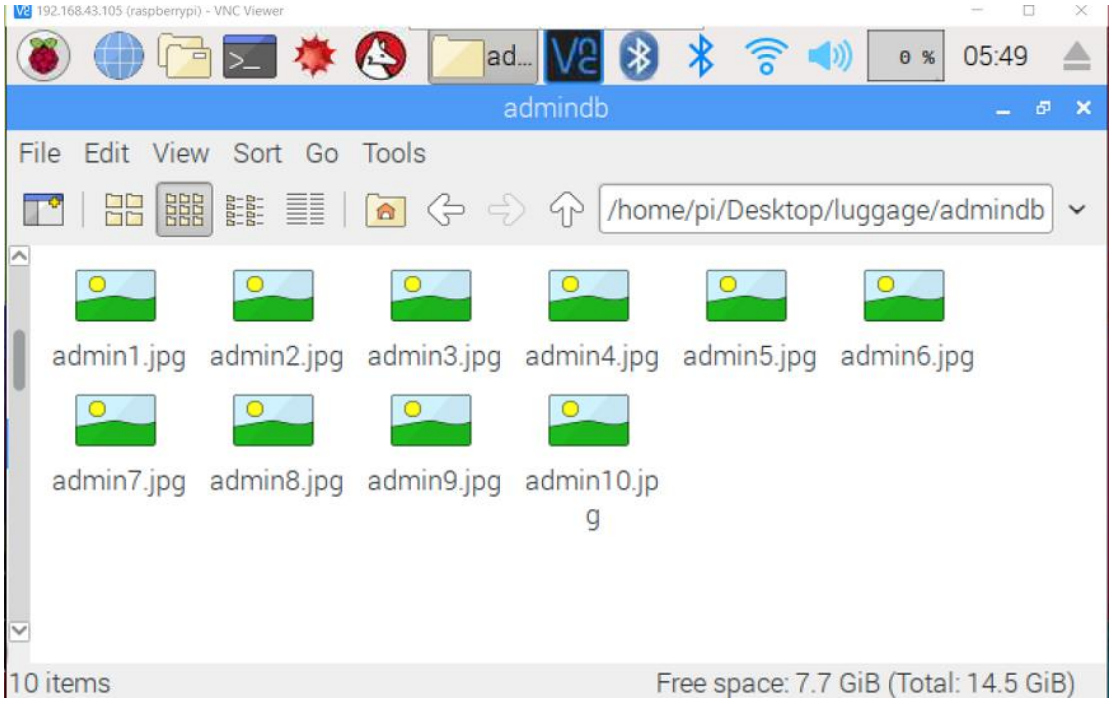

Fig6.6: Admin Image Captured and Stored in Database

7. First the user's database is loaded here and the camera starts to detect the users face. Using haar cascade algorithm the images are detected. After detecting, faces are compared with the user's database and using LBPH face is recognised. Once the face is recognised the QR code is generated and is mailed to the proper email id of the user. If the face is not recognised, it generates a message saying image is not present.

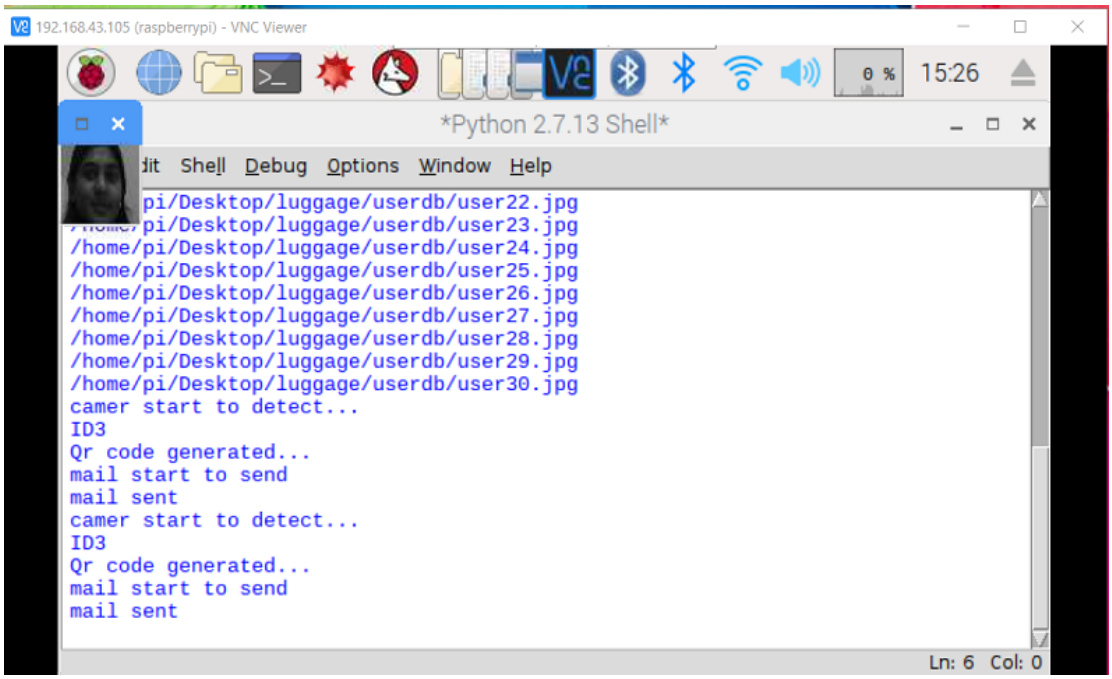

Fig6.7: For face detection and face recognition and to generate QR code.

8. In this page, first the admin database is loaded and camera starts to recognise whether the detected face is admin or not. If the admin's face is recognised video starts to stream to scan the QR code that is sent to the user through e-mail. Once the QR code is scanned, the user's information stored in it will be displayed. This information will be updated to the server. This information can be used by the admin, to trace back the luggage in case of misplacement.

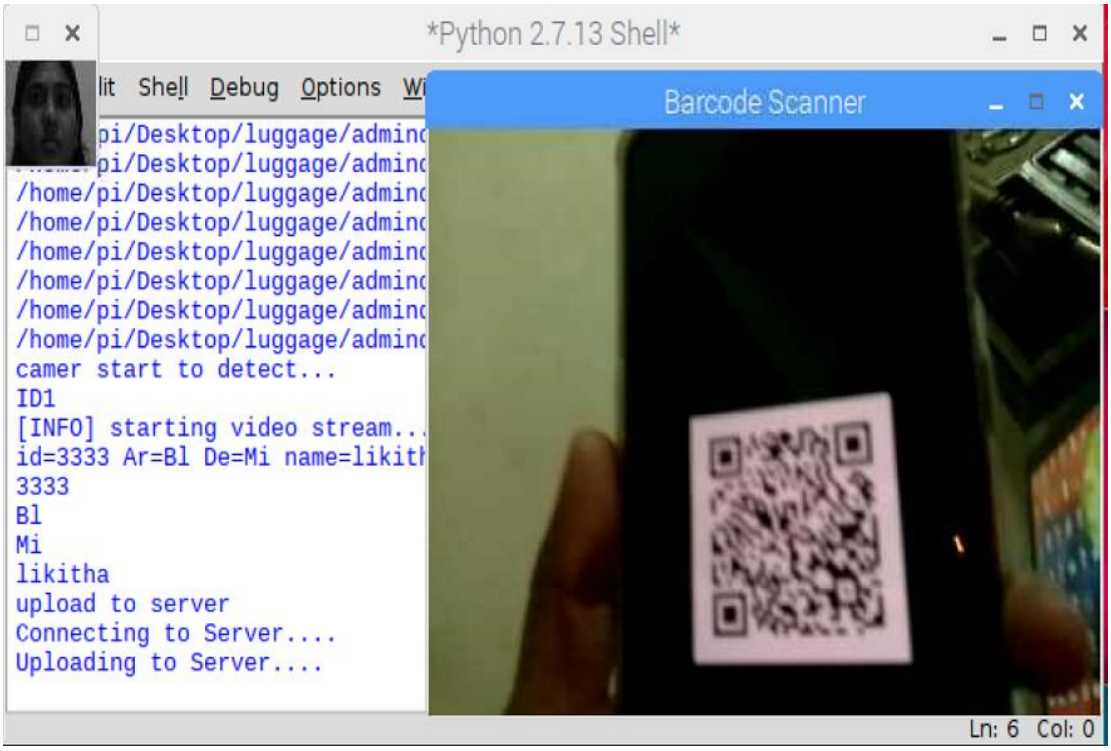

Fig6.8: To scan QR code and store the information present in QR code to server.

CHAPTER 7

TESTING

Software testing is a process used to help identify the correctness, completeness and quality of developed computer software. Software testing is the process used to measure the quality of developed software. Testing is the process of executing a program with the intent of finding errors. Software testing is often referred to as verification and validation.

7.1 Functional Testing

The goal of utilizing numerous testing methodologies in your development process is to make sure your software can successfully operate in multiple environments and across different platforms. These can typically be broken down between functional and non-functional testing. Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using uses cases provided by the design team or business analyst. These testing methods are usually conducted in order and include:

- Unit testing
- Integration testing
- System testing

7.2 Unit Testing

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing can be conducted manually, but automating the process will speed up delivery cycles and expand test coverage. Unit testing will also make debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process. Test Left is a

tool that allows advanced testers and developers to shift left with the fastest test automation tool embedded in any IDE. In this project we have four units such as to capture the image, to store it in database, for face detection and face recognition and to send QR code to user's email. Each every module is tested and executed as required.

7.3 Integration Testing

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected (i.e., the interactions between units are seamless). These tests are often framed by user scenarios, such as logging into an application or opening files. Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests. In this project we have integrated two components and one is used for capturing image and other component is used for storing into the database in raspberry pi board.

7.4 System Testing

System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of the software is tested from end-to-end and is typically conducted by a separate testing team than the development team before the product is pushed into production. To display the users luggage information in the server side using IP address and to capture user face. Displaying the users information when face is recognised and to send the QR code to user mail.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

An efficient based luggage tracking system using raspberry pi was proposed. A number of technologies have been implemented to speed these processes but one technology that has the potential to revolutionize luggage handling technique is based on Iot technique. This system helps us to find the luggage with the help of user face recognition using QR code scanning

8.2 Future Scope

In future, we are planning to include some features such as automatic object avoidance, stair case climbing, automatic zip opening on fingerprint and these extra features make the luggage more safety and user friendly. The application should be more dynamic and it should show the live feed of the movement of the luggage which updates every time the luggage moves. The tracking could be taken online using the cloud technology

REFERENCES

In IEEE style https:/www.sciencedirect.com/science/article/pii/s2001037014000464 papers:

[1] Mohammed Ghazal, Samr Ali, Fasila Haneefa, and Ahmed Sweleh, "Towards Smart Wearable Real-time Airport Luggage Tracking", 978-1-5090-3484-0/16 ©2016 IEEE

[2] Rishabh Chaturvedi, Krishna Darji, Atul Mahajan, Maya Varghese, "Real Time Airport Luggage Tracking System", 746-2-4221-0110-9/16 ©2016 ICCSII

[3] Wessam Shehieb, Hafez A1 Sayed, Mohamad mazen Akil, M. Anas sarraj, Mustahsan Mir, "A Smart System to Minimize Mishandled Luggage at airports", 978-1-5090-3484-0/16 ©2016 IEEE

[4] P Sai Vamsi , V Madava Sarma, S.V.Y.S Samraj, S.R Deepika, N.Neha , K Prabhakara Rao, "Smart Luggage", 2017 ICEI