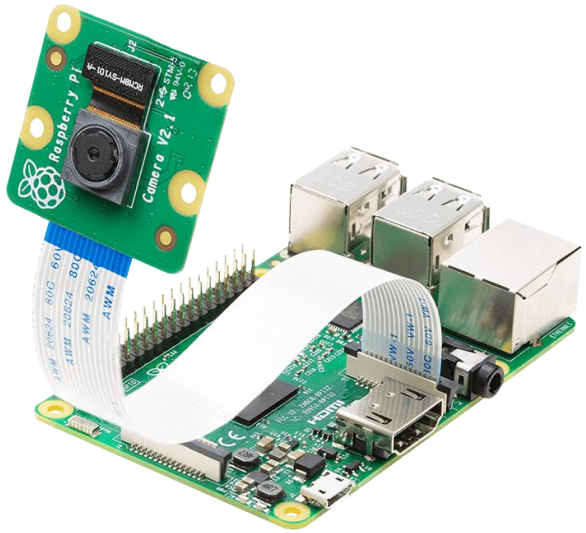
DEVELOPMENT OF IOT-BASED SOLUTION FOR CHECKING STUDENT ATTENDANCE



Student 1: Nguyen Truong Thanh (1652557)

Student 2: Nguyen Anh Quan (1552310)

Supervise: PhD. Pham Hoang Anh

January 25th, 2021

CONTENTS

INTRODUCTION

- Attendance marking
- Traditionally marking
- Fingerprint
- RFID Card
- Face Recognition

LITERATURE REVIEW

Existing Literature	Limitations	Solution
Fingerprint	Employees have to wait on queues	Marking attendance automatically and quickly without waiting in long queues
Facial Recognition	Faces will change over time and it is difficult to recognize faces if the face is too far from the camera	Load student photos with relevant ID numbers on RFID tags and update student face on the database every 2 weeks
RFID Technology	Can make fake attendance	Verify faces by checking faces with facial recognition after being identified by RFID
The student will hear beep sound after recognition	Students do not receive any notices after marking attendance	Send email notifications to students after marking his/her attendance

RESEARCH OBJECTIVES

- Identify students using RFID
- Identify students using face recognition
- Register courses and subject on the system
- Mark attendance on the database
- Check whether the student attended to correct session
- Send notification emails to students
- Generate attendance reports

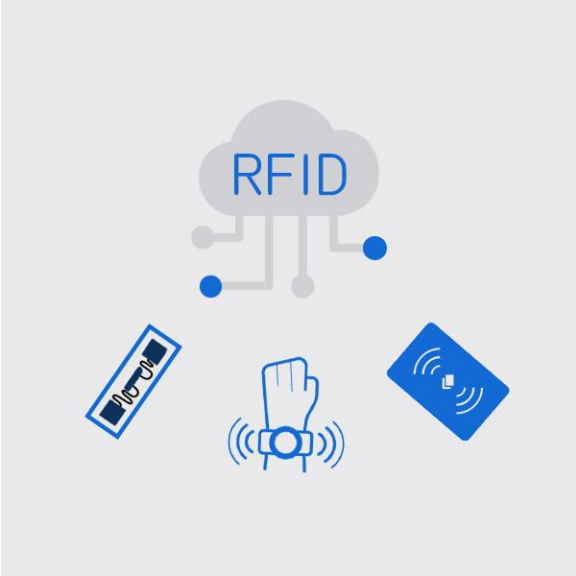
THEORETICAL BASIS

- Radio Frequency Identification
- Face Recognition
- Facial Lankmark

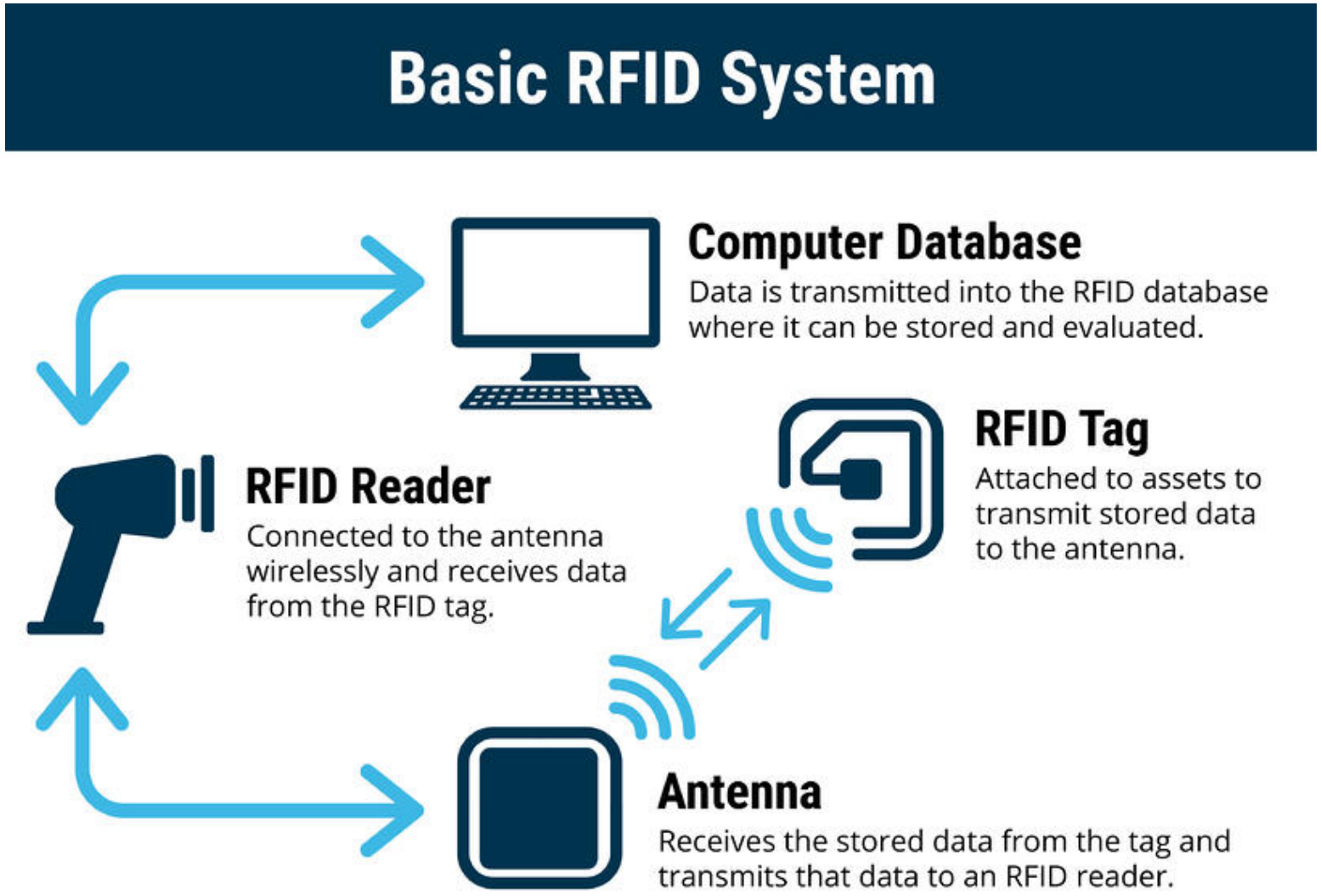
RADIO FREQUENCY IDENTIFICATION

INTRODUCTION

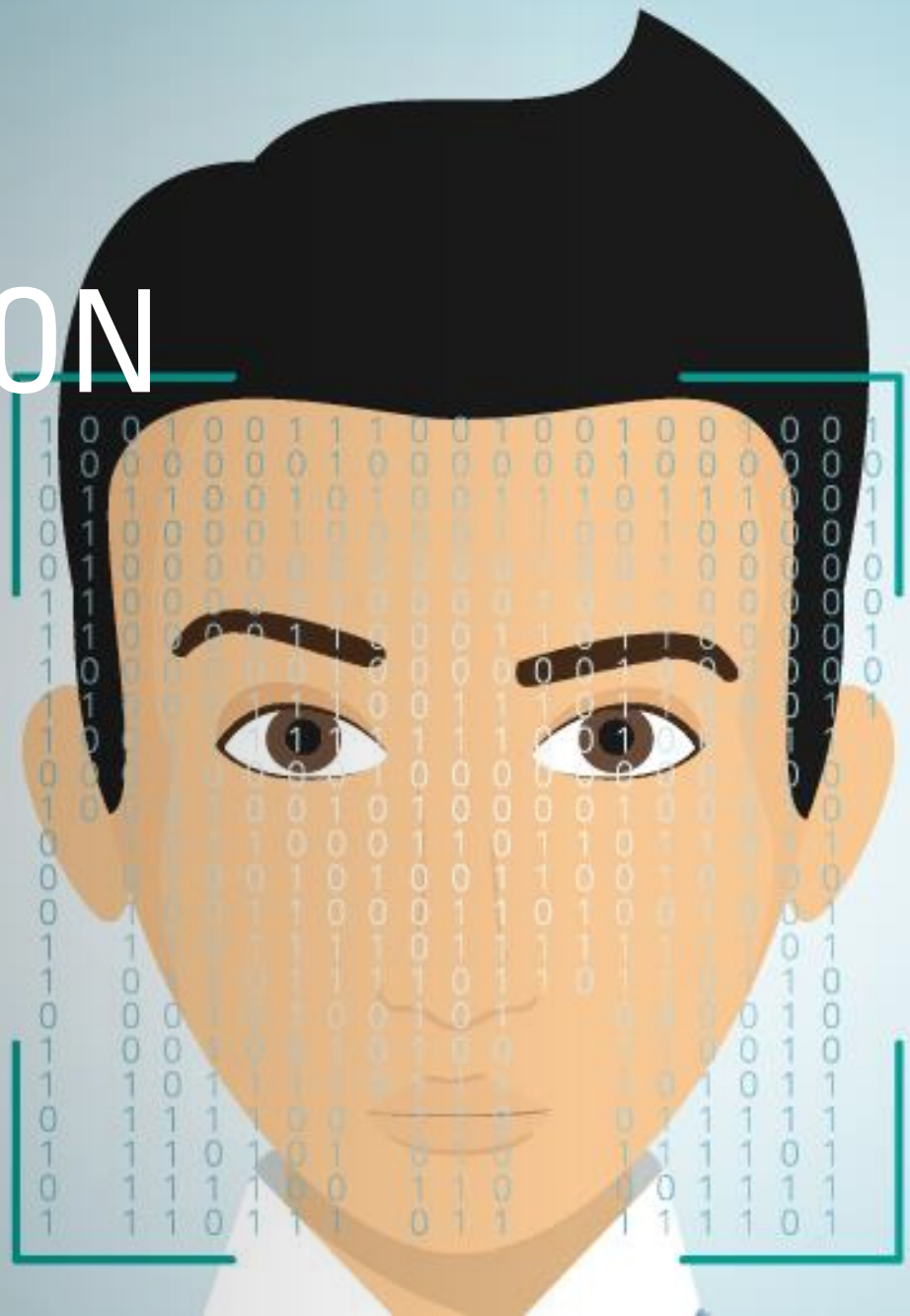
- Has a long history and is a part of both present andpast technological development.
- Application:
  - Inventory management
  - Asset tracking
  - Personnel tracking
  - Controlling access to restricted areas
  - ID Badging
  - Supply chain management



RFID SYSTEM COMPONENTS



FACE RECOGNITION



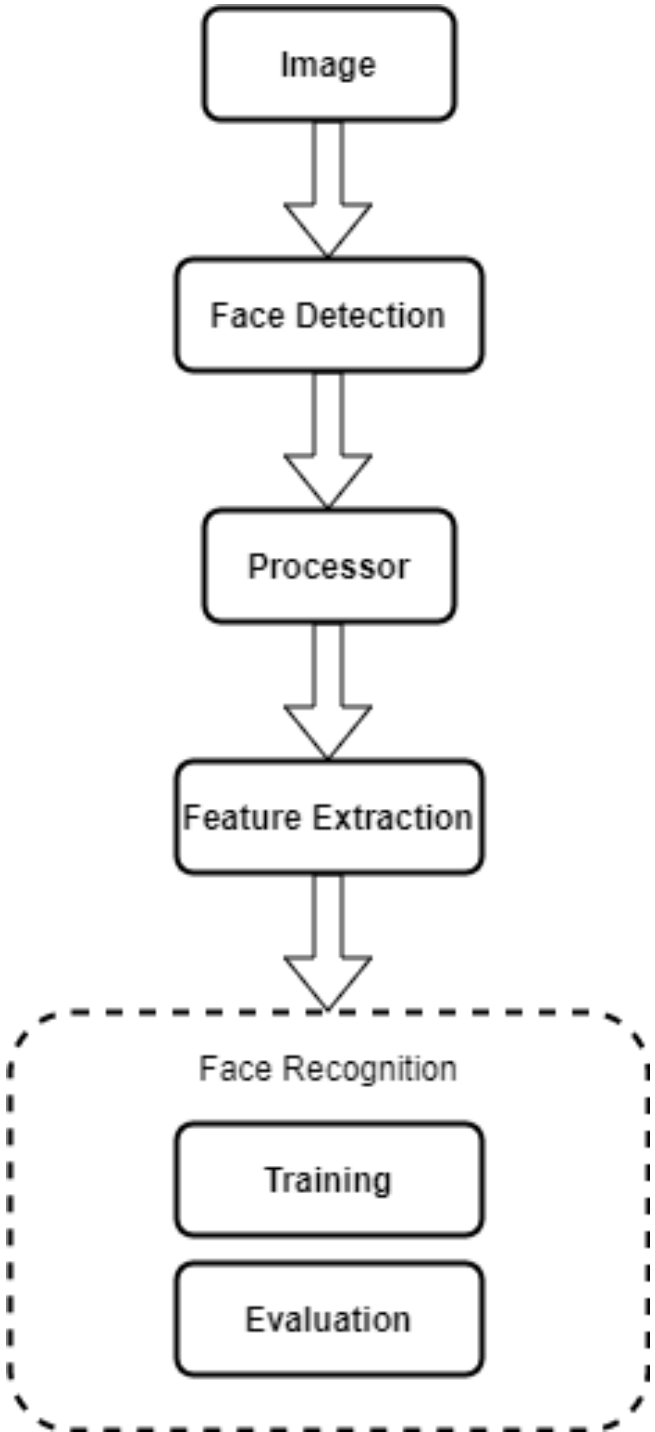
FACE DETECTION

- A computer technology being used in a variety of applications that identifies human faces in digital images.
- Face detection can be regarded as a specific case of object-class detection.
- Face-detection algorithms focus on the detection of frontal human faces.

FACE RECOGNITION

- Face recognition systems can be used to identify people in photos, video, or in real-time.
- The **face detection** process is an essential step as it detects and locates human faces in images and videos.
- The **face capture** process transforms analog information (a face) into a set of digital information (data) based on the person's facial features.
- The **face match** process verifies if two faces belong to the same person.

FACE RECOGNITION

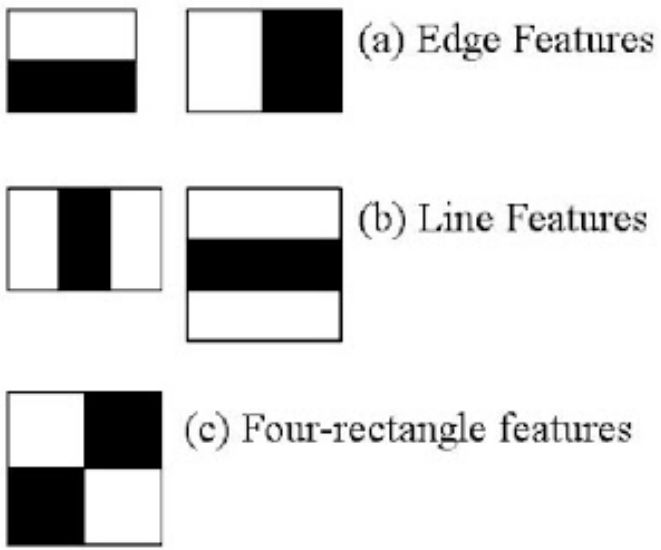


HAAR CASCADE CLASSIFIER

A machine learning based approach where a cascade function is trained from a lot of positive and negative images.

The algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier.

HAAR CASCADE CLASSIFIER

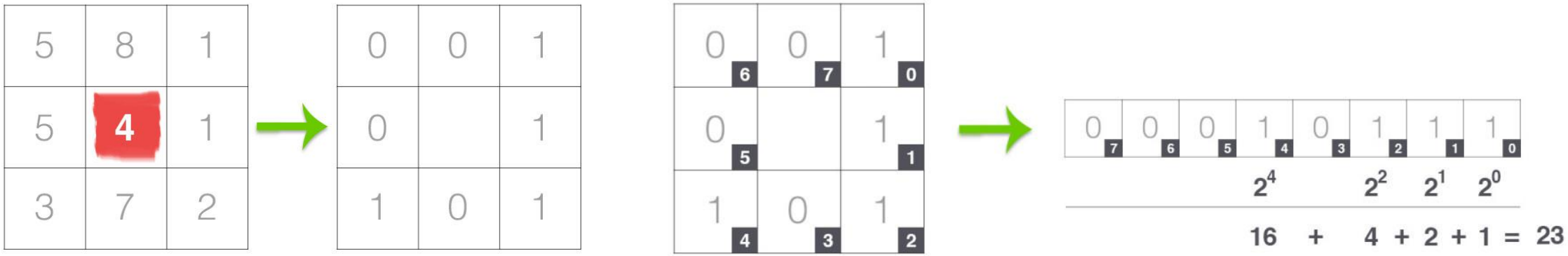


- Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.

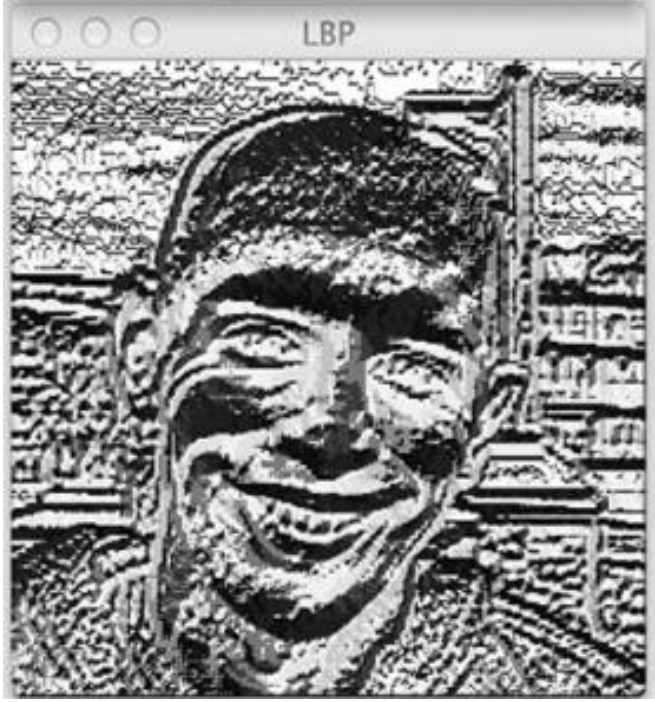
LBP (LOCAL BINARY PATTERNS) CASCADE CLASSIFIER

- To convert the image to grayscale.
- For each pixel in the grayscale image, we select a neighborhood of size r surrounding the center pixel.
- A LBP value is then calculated for this center pixel and stored in the output 2D array with the same width and height as the input image.

LBP (LOCAL BINARY PATTERNS) CASCADE CLASSIFIER




Image


LBP

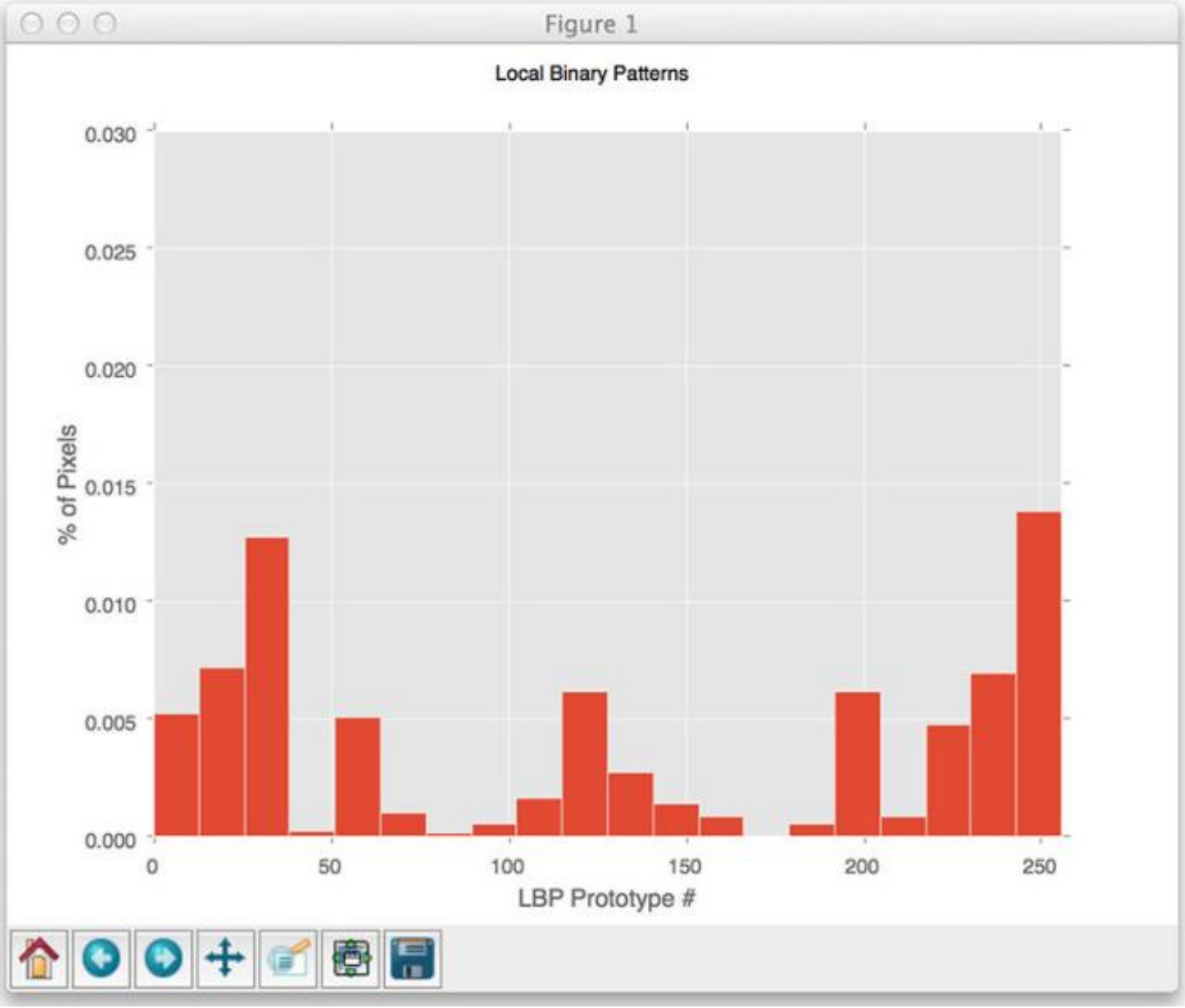
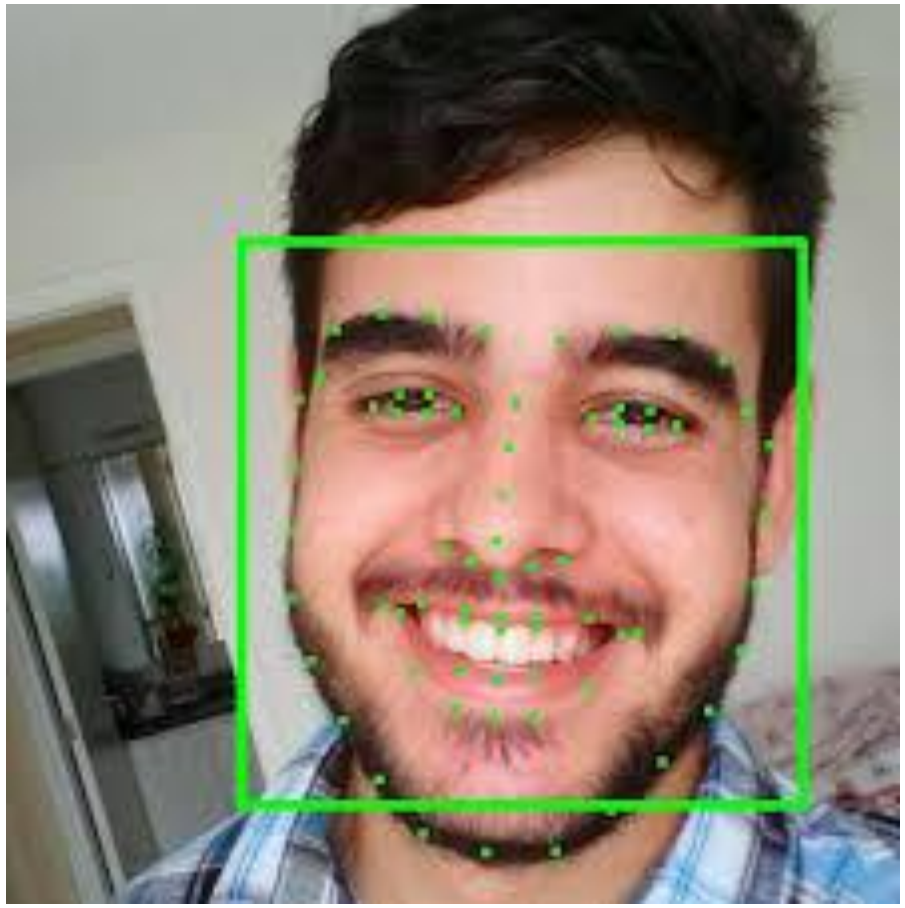


Figure 1
Local Binary Patterns
% of Pixels
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
50
100
150
200
250
LBP Prototype #

OPENCV

- OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library.
- Detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, etc.
- Supports Windows, Linux, Android and Mac OS.

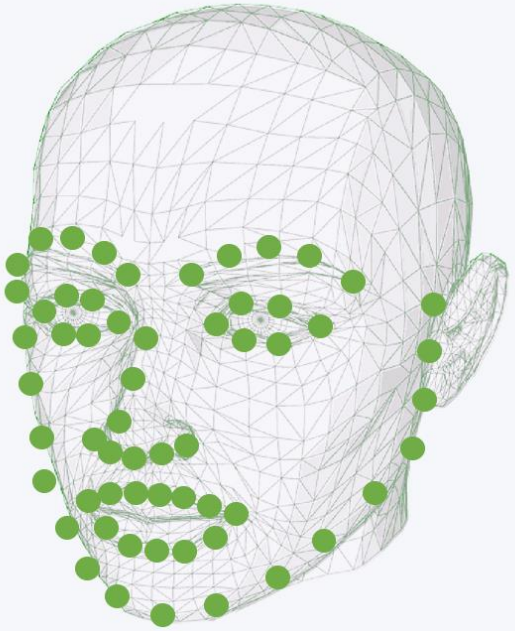
FACIAL LANDMARKS



FACIAL LANDMARKS

- Facial landmarks are used to localize and represent salient regions of the face, such as: (eyes, eyebrows, nose, mouth, jawline)
- Face alignment, head pose estimation, face swapping, blinl detection, etc.

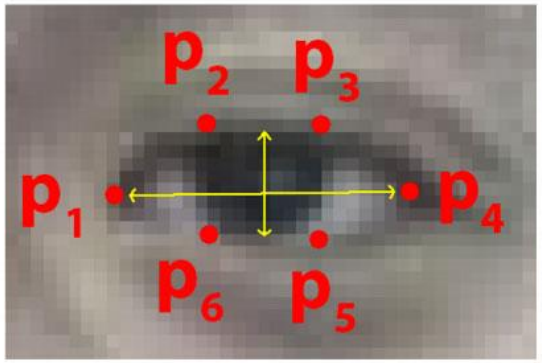
FACIAL LANDMARK POINTS



Feature	Point range
Left jaw line	0-7
Chin	8
Right jaw line	9-16
Left eyebrow	17-21
Right eyebrow	22-26
Bridge of nose	27-30
Bottom of nose	31-35
Left eye	36-41
Right eye	42-47
Outer edge of lips	48-59
Inner edge of lips	60-67

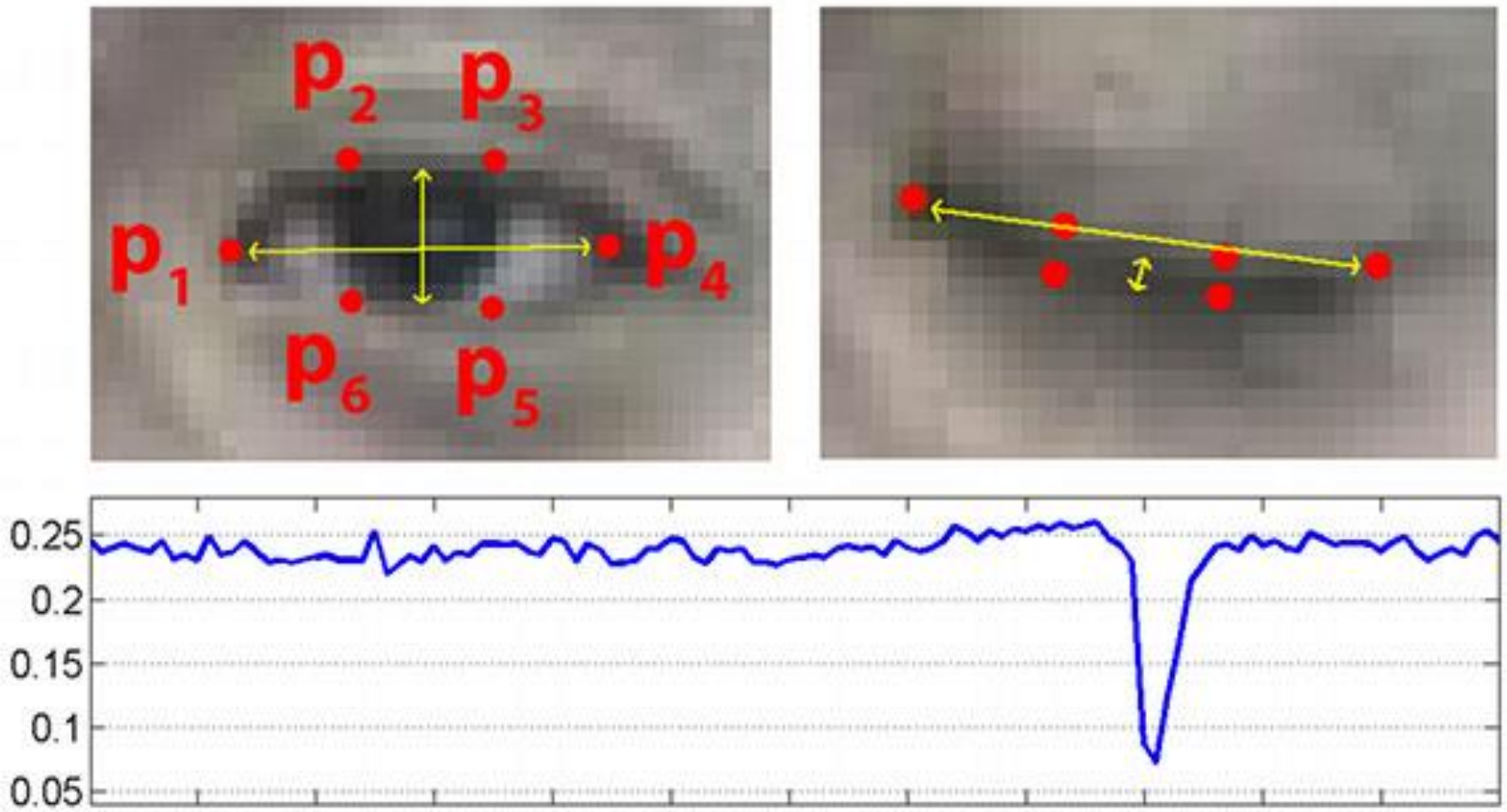
EYE ASPECT RATIO (EAR)

- Each eye is represented by 6 (x, y)-coordinates

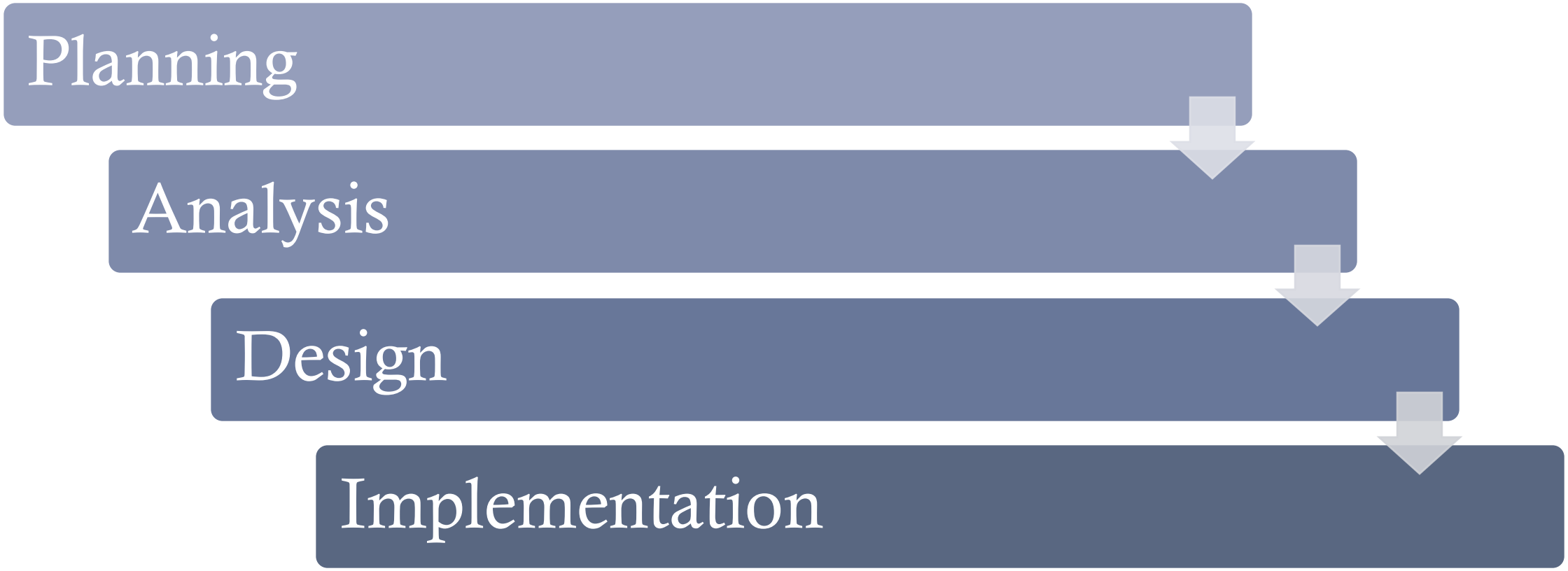


- There is a relation between the width and the height of these coordinates

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$


p2
p3
p1
p4
p6
p5
0.25
0.2
0.15
0.1
0.05

METHODOLOGY

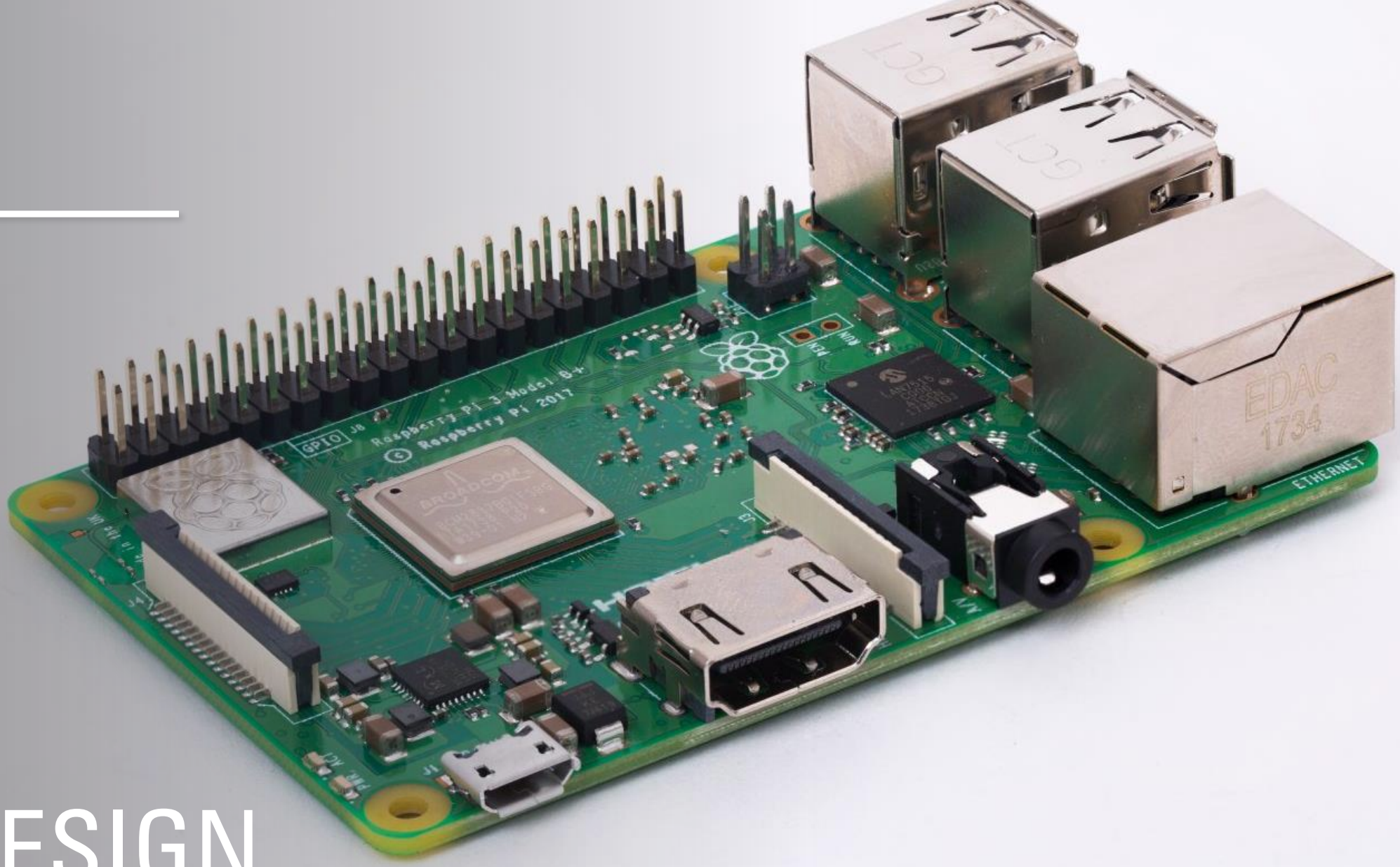


PLANNING

- Problems
- Traditional
- RFID
- Face Recognition
- Fingerprint
- Number of students

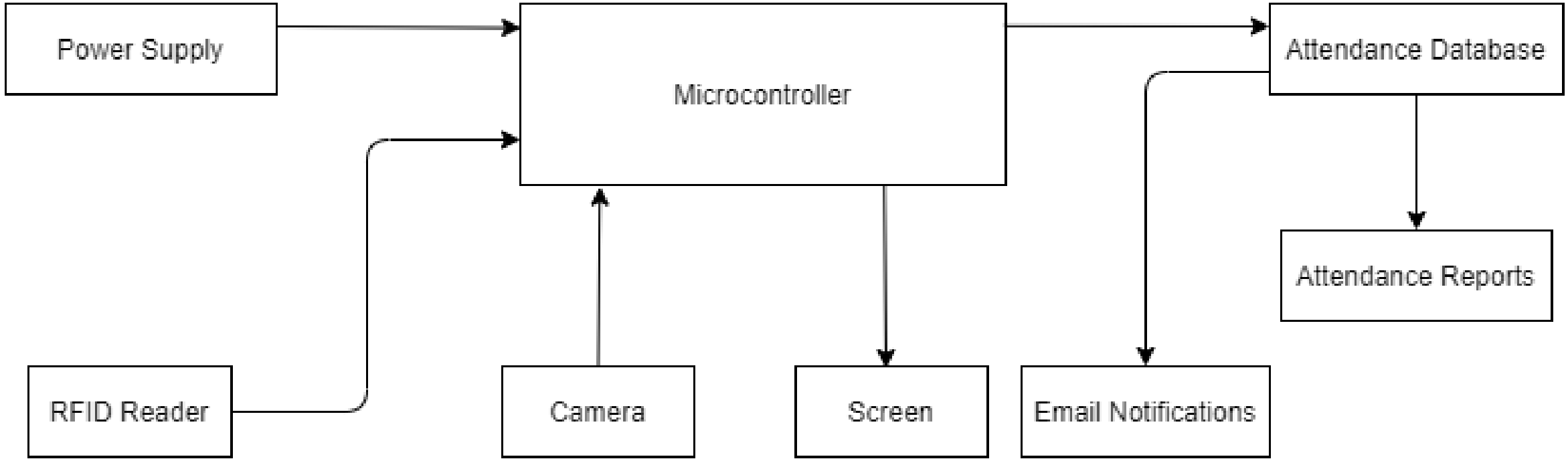
ANALYSIS

RFID + Facial Recognition

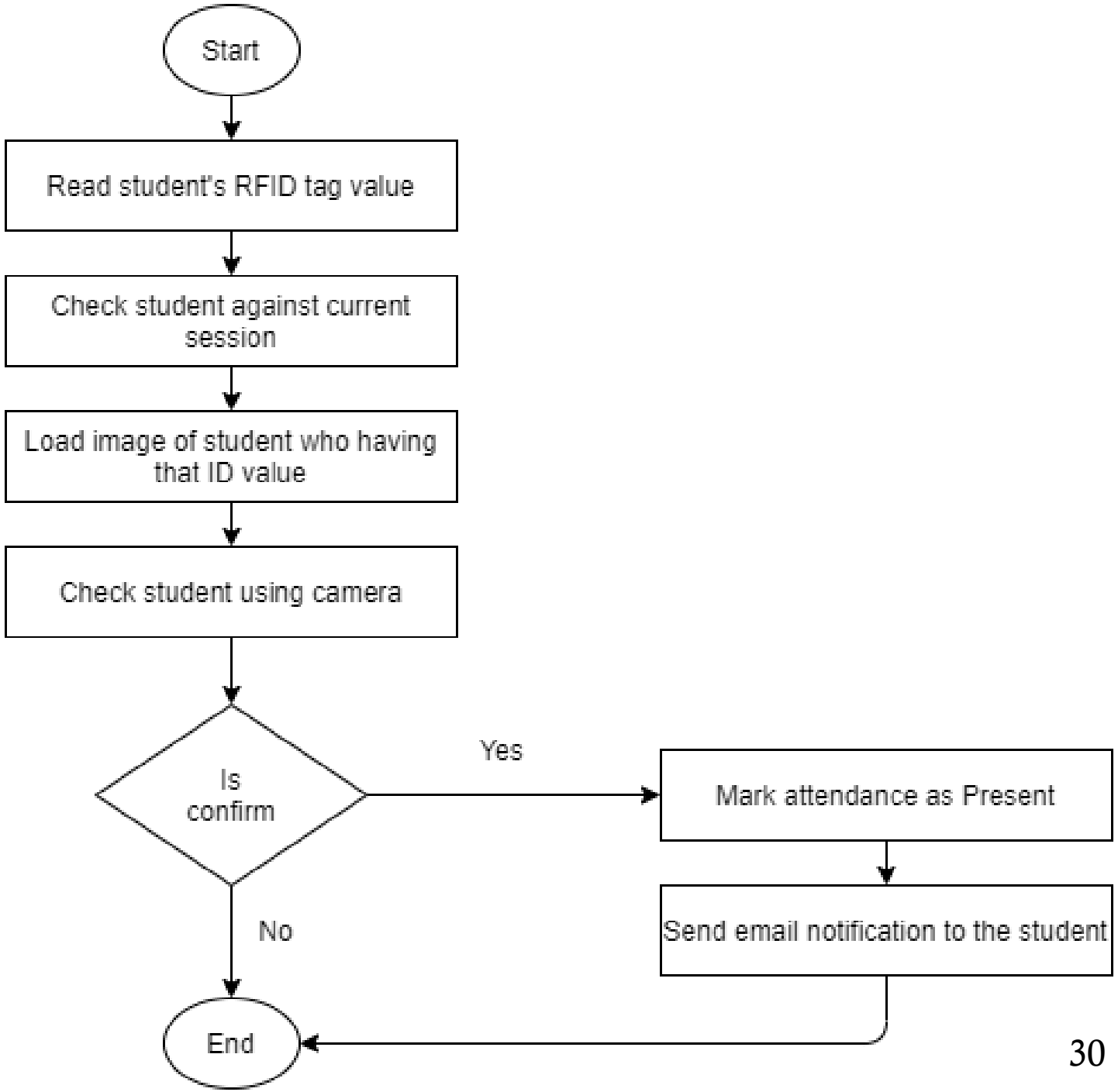

GPIO
Raspberry Pi 3 Model B+
© Raspberry Pi 2017
EDAC
1734
ETHERNET

DESIGN

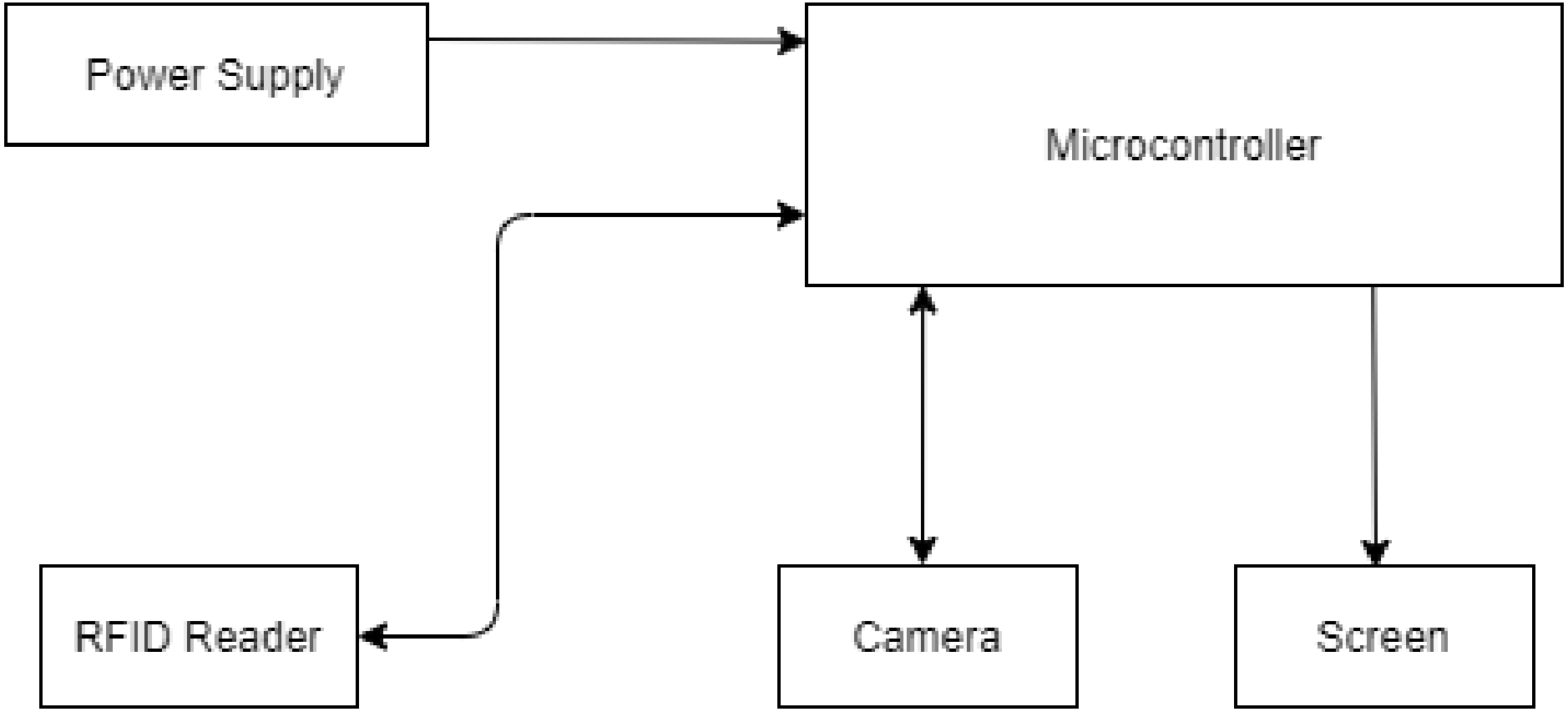
SYSTEM OVERVIEW

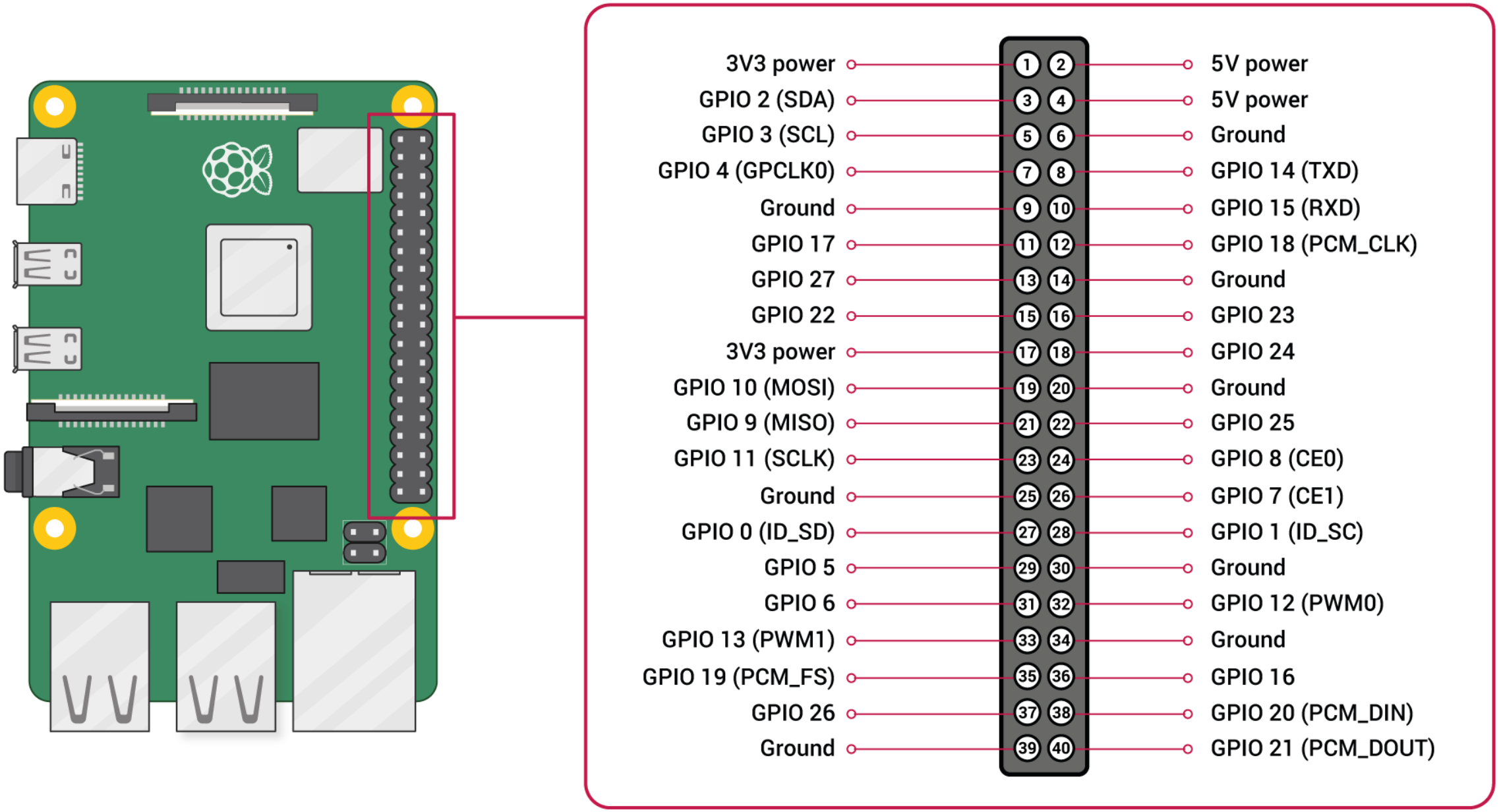


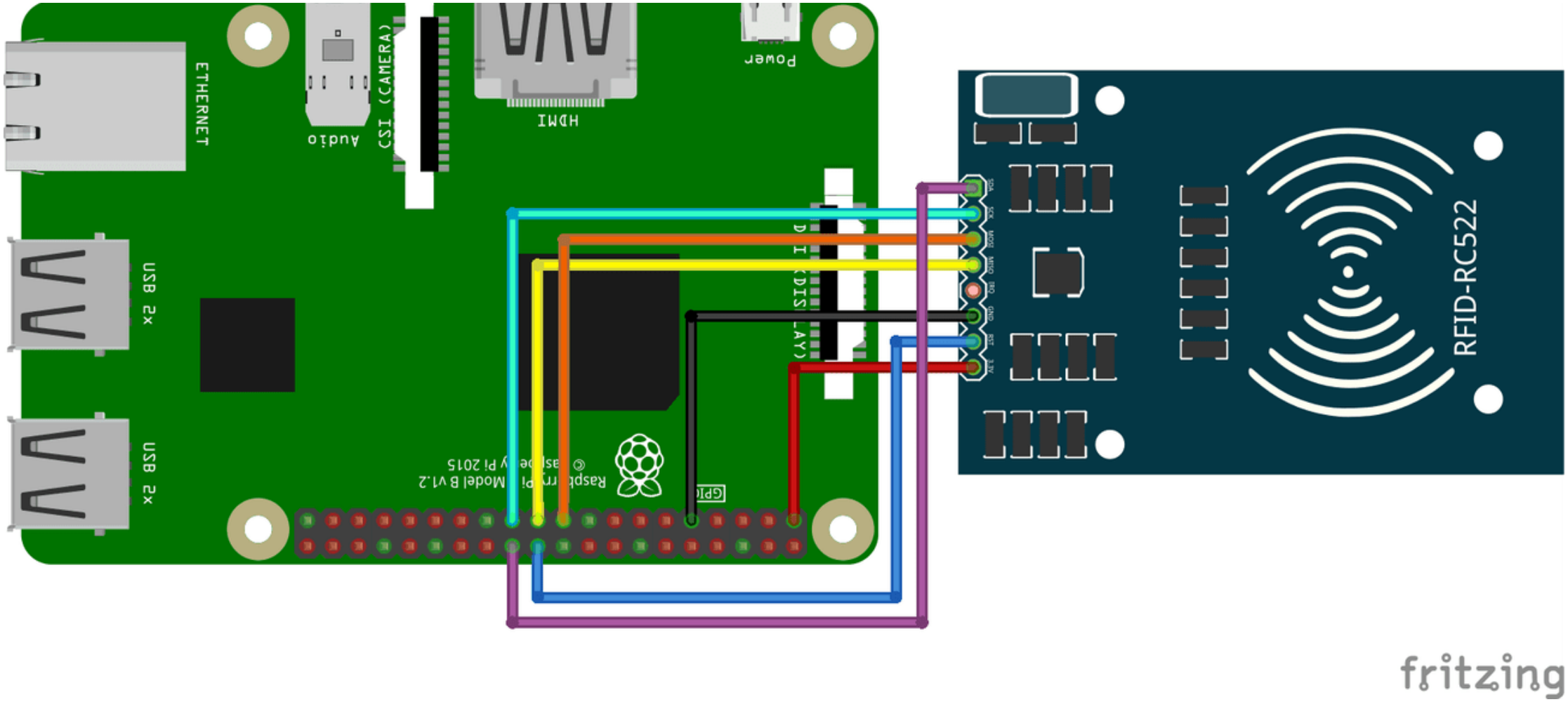
FLOWCHART

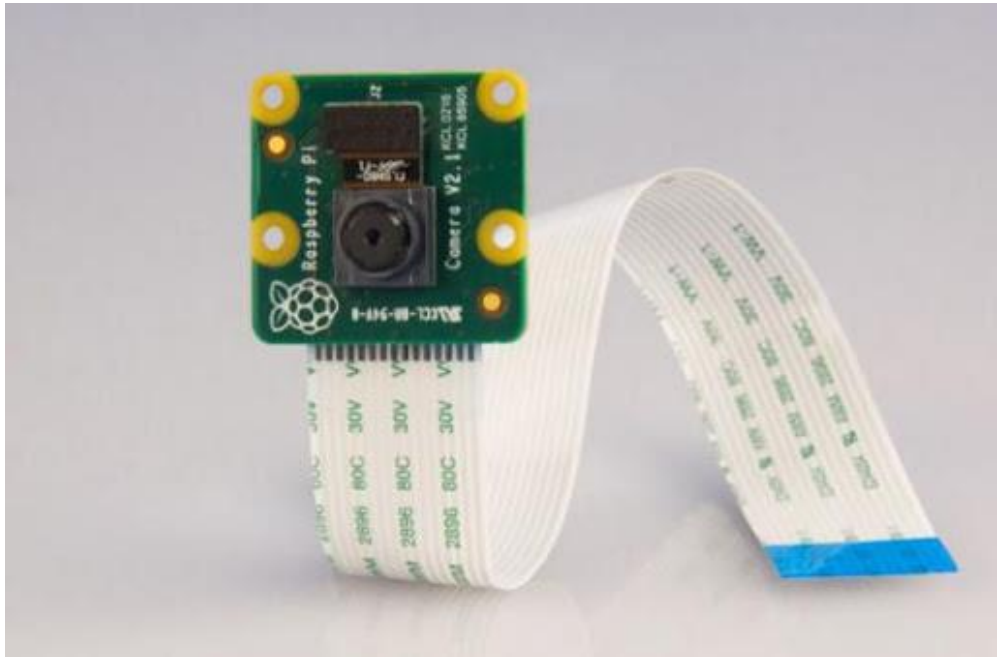

Start
Read student's RFID tag value
Check student against current session
Load image of student who having that ID value
Check student using camera
Is confirm
Yes
No
Mark attendance as Present
Send email notification to the student
End

SYSTEM DESIGN

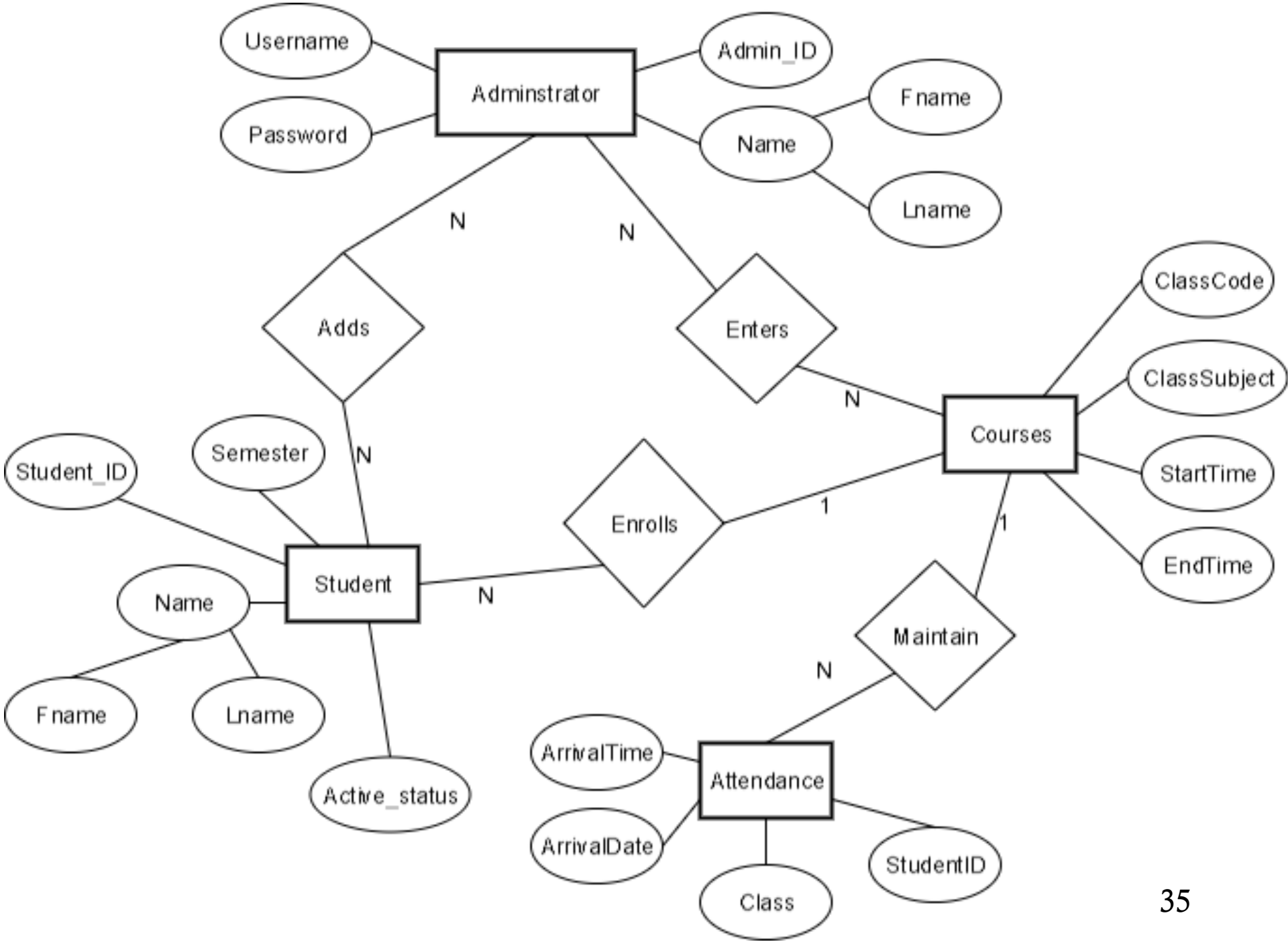



3V3 power 1
GPIO 2 (SDA) 3
GPIO 3 (SCL) 5
GPIO 4 (GPCLK0) 7
Ground 9
GPIO 17 11
GPIO 27 13
GPIO 22 15
3V3 power 17
GPIO 10 (MOSI) 19
GPIO 9 (MISO) 21
GPIO 11 (SCLK) 23
Ground 25
GPIO 0 (ID_SD) 27
GPIO 5 29
GPIO 6 31
GPIO 13 (PWM1) 33
GPIO 19 (PCM_FS) 35
GPIO 26 37
Ground 39
2 5V power
4 5V power
6 Ground
8 GPIO 14 (TXD)
10 GPIO 15 (RXD)
12 GPIO 18 (PCM_CLK)
14 Ground
16 GPIO 23
18 GPIO 24
20 Ground
22 GPIO 25
24 GPIO 8 (CE0)
26 GPIO 7 (CE1)
28 GPIO 1 (ID_SC)
30 Ground
32 GPIO 12 (PWM0)
34 Ground
36 GPIO 16
38 GPIO 20 (PCM_DIN)
40 GPIO 21 (PCM_DOUT)

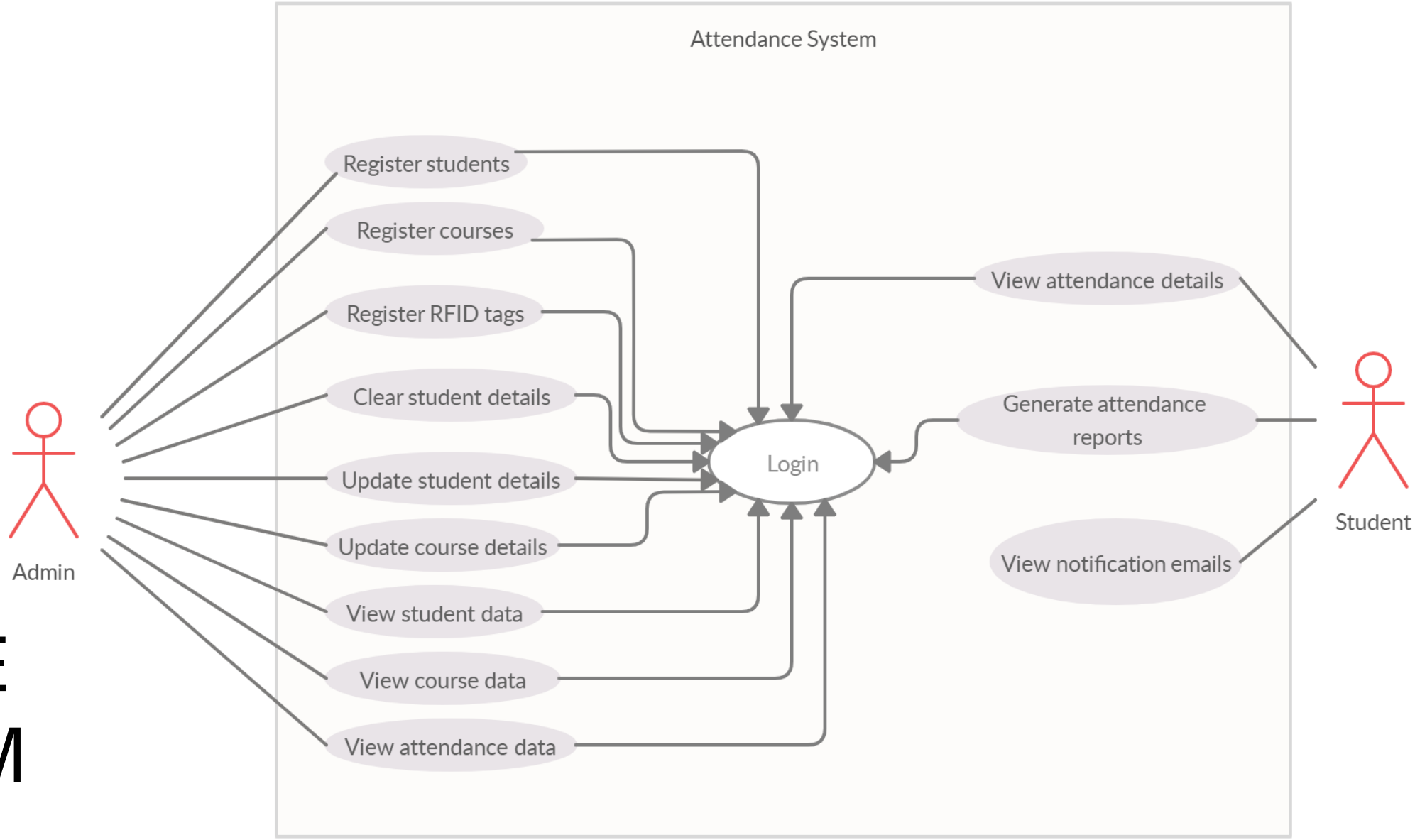

ETHERNET
Audio
CSI (CAMERA)
HDMI
Power
USB 2x
USB 2x
DISPLAY
GPIO
Raspberry Pi Model B v1.2
© Raspberry Pi 2015
3.3V
RST
GND
MISO
MOSI
SCK
SDA
RFID-RC522
fritzing


Raspberry Pi
Camera V2.1

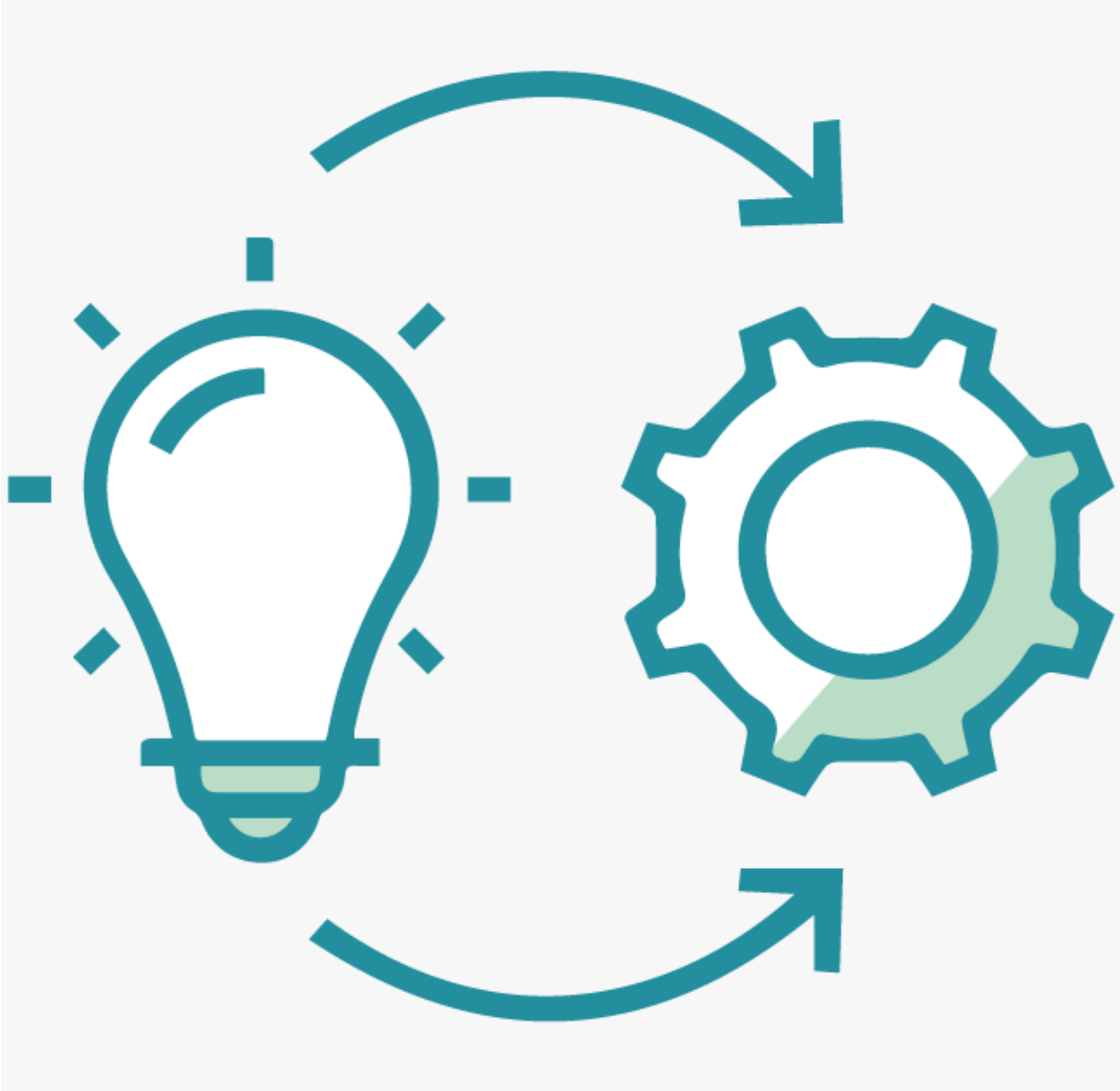
ER DIAGRAM OF DATABASE



USECASE DIAGRAM



IMPLEMENTATION



DATABASE

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id_atd**	int(255)		UNSIGNED	No	*None*		AUTO_INCREMENT
2	**first_name**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		
3	**last_name**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		
4	**student_number**	int(7)			Yes	*NULL*		
5	**class_number**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		
6	**clock_in**	varchar(255)	utf8mb4_general_ci		No	current_timestamp()		

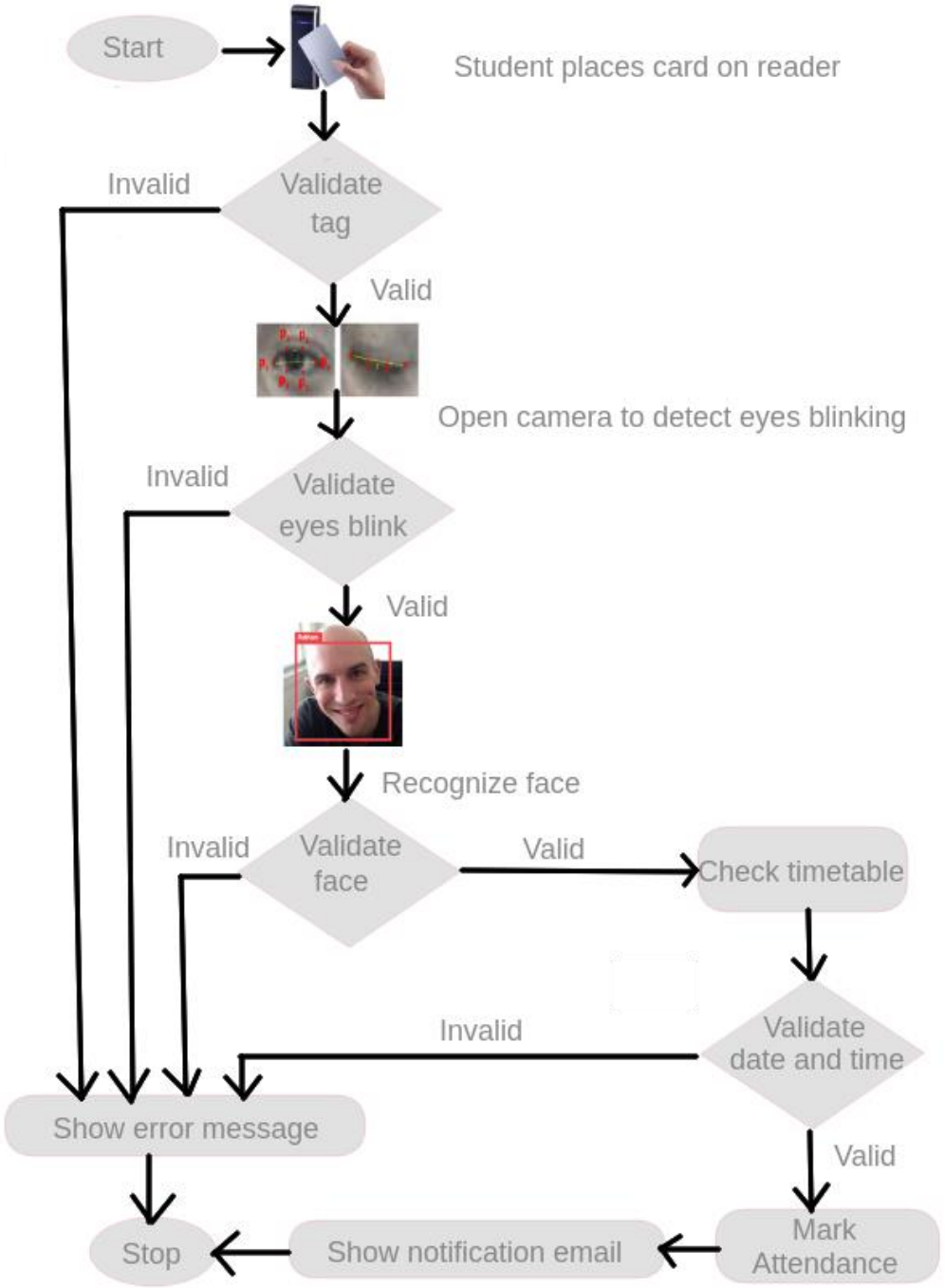
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id_class**	int(255)			No	*None*		AUTO_INCREMENT
2	**subject_code**	varchar(10)	utf8mb4_general_ci		Yes	*NULL*		
3	**subject_name**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		
4	**day_in_week**	varchar(10)	utf8mb4_general_ci		Yes	*NULL*		
5	**start_time**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		
6	**end_time**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		
7	**room**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id_stu**	int(10)			No	*None*		AUTO_INCREMENT
2	**first_name**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		
3	**last_name**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		
4	**student_number**	int(7)			Yes	*NULL*		
5	**email**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		
6	**rfid_uid**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		
7	**class_list**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		
8	**created**	varchar(255)	utf8mb4_general_ci		No	current_timestamp()		

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id_login**	int(11)			No	*None*		AUTO_INCREMENT
2	**fname**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		
3	**lname**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		
4	**email**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		
5	**username**	varchar(100)	utf8mb4_general_ci		Yes	*NULL*		
6	**password**	varchar(100)	utf8mb4_general_ci		Yes	*NULL*		
7	**userlevel**	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
8	**reset_token**	varchar(255)	utf8mb4_general_ci		Yes	*NULL*		

PYTHON APP

MARKING ATTENDANCE



RFID

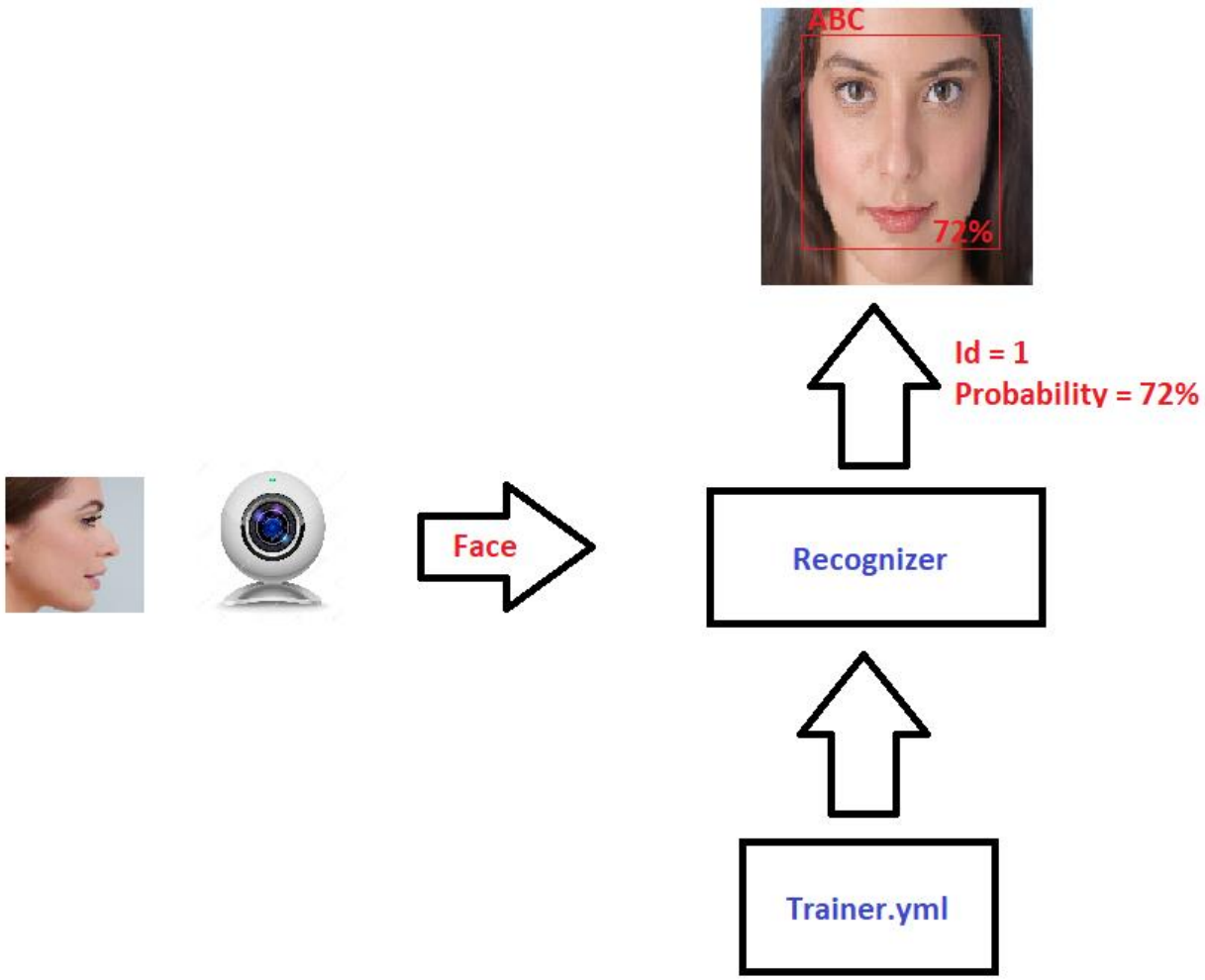
```
#!/usr/bin/env python

import RPi.GPIO as GPIO
import SimpleMFRC522

reader = SimpleMFRC522.SimpleMFRC522()

try:
        text = raw_input('New data:')
        print("Now place your tag to write")
        reader.write(text)
        print("Written")
finally:
        GPIO.cleanup()
```

FACE DETECTION

```
faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (int(minW), int(minH)),
       )
```

```
for(x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
```

FACE RECOGNITION



EYE BLINK DETECTION

```
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

def midpoint(p1,p2):
    return int((p1.x + p2.x)/2),int((p1.y + p2.y)/2)

font = cv2.FONT_HERSHEY_SIMPLEX

def get_blinking_ratio(eye_points, facial_landmarks):
    left_point = (facial_landmarks.part(eye_points[0]).x, facial_landmarks.part(eye_points[0]).y)
    right_point = (facial_landmarks.part(eye_points[3]).x, facial_landmarks.part(eye_points[3]).y)
    hor_line = cv2.line(img, left_point, right_point,(0,255,0), 1)

    center_top = midpoint(facial_landmarks.part(eye_points[1]), facial_landmarks.part(eye_points[2]))
    center_bottom = midpoint(facial_landmarks.part(eye_points[5]), facial_landmarks.part(eye_points[4]))
    ver_line = cv2.line(img, center_top, center_bottom,(0,255,0), 1)

    #length of the line
    hor_line_length = hypot((left_point[0] - right_point[0]), (left_point[1] - right_point[1]))
    ver_line_length = hypot((center_top[0] - center_bottom[0]), (center_top[1] - center_bottom[1]))
    ratio = hor_line_length/ ver_line_length, ver_line_length
    return ratio
```

EYE BLINK DETECTION

```
while True:
    ret,img = video_capture.read()


    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = detector(gray)

    for face in faces:
        #x, y = face.left(), face.top()
        #x1, y1 = face.right(), face.bottom()
        #cv2.rectangle(frame, (x,y), (x1,y1), (0,255,0), 3 )# green box, thickness of box
        landmarks = predictor(gray, face)

        left_eye_ratio,_ = get_blinking_ratio([36,37,38,39,40,41], landmarks)

        right_eye_ratio, myVerti = get_blinking_ratio([42,43,44,45,46,47], landmarks)

        blinking_ratio = (left_eye_ratio+right_eye_ratio)/2

        if(blinking_ratio >= 6):
            cv2.putText(img, "blinking", (50,50), cv2.FONT_HERSHEY_SIMPLEX, 2, (255,0,0))
            print("blinking")

            img = recognize(img,clf,faceCascade)

        elif((blinking_ratio < 6) and (delta>30)):
            print("Fake image----")
            messagebox.showerror('Error','-------Fake image-------')
            call_var()
```

smart-checking-atte... | Thonny - /home/pi/... | Face recognition sys... | face detection | Error | 21:59

Face recognition system

First name
Last name
Student id
Email

Check Attendance

Training Dataset

Error

-------Fake image-------

OK

Low voltage warning

Please check your power supply

face detection

```
for face in faces:
    #x, y = face.left(), face.top()
    #x1, y1 = face.right(), face.botto
    #cv2.rectangle(frame, (x,y), (x1,y
    landmarks = predictor(gray, face)

    left_eye_ratio,_ = get_blinking_ra

    right_eye_ratio, myVerti = get_bli

    blinking_ratio = (left_eye_ratio+r
```

Shell

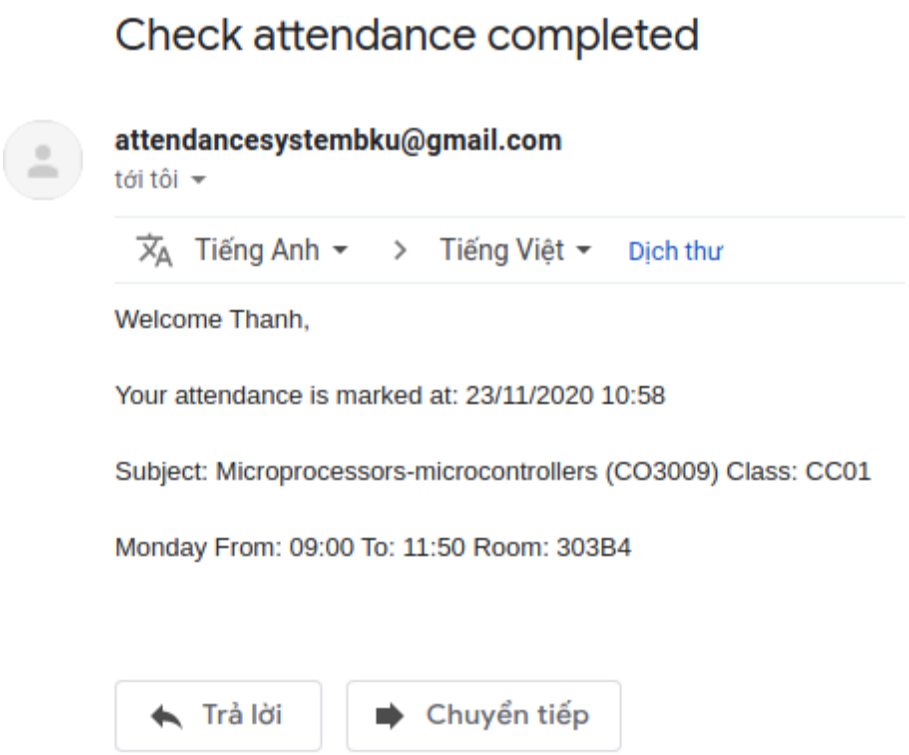
```
1
Fake image----
```

Python 3.7.3

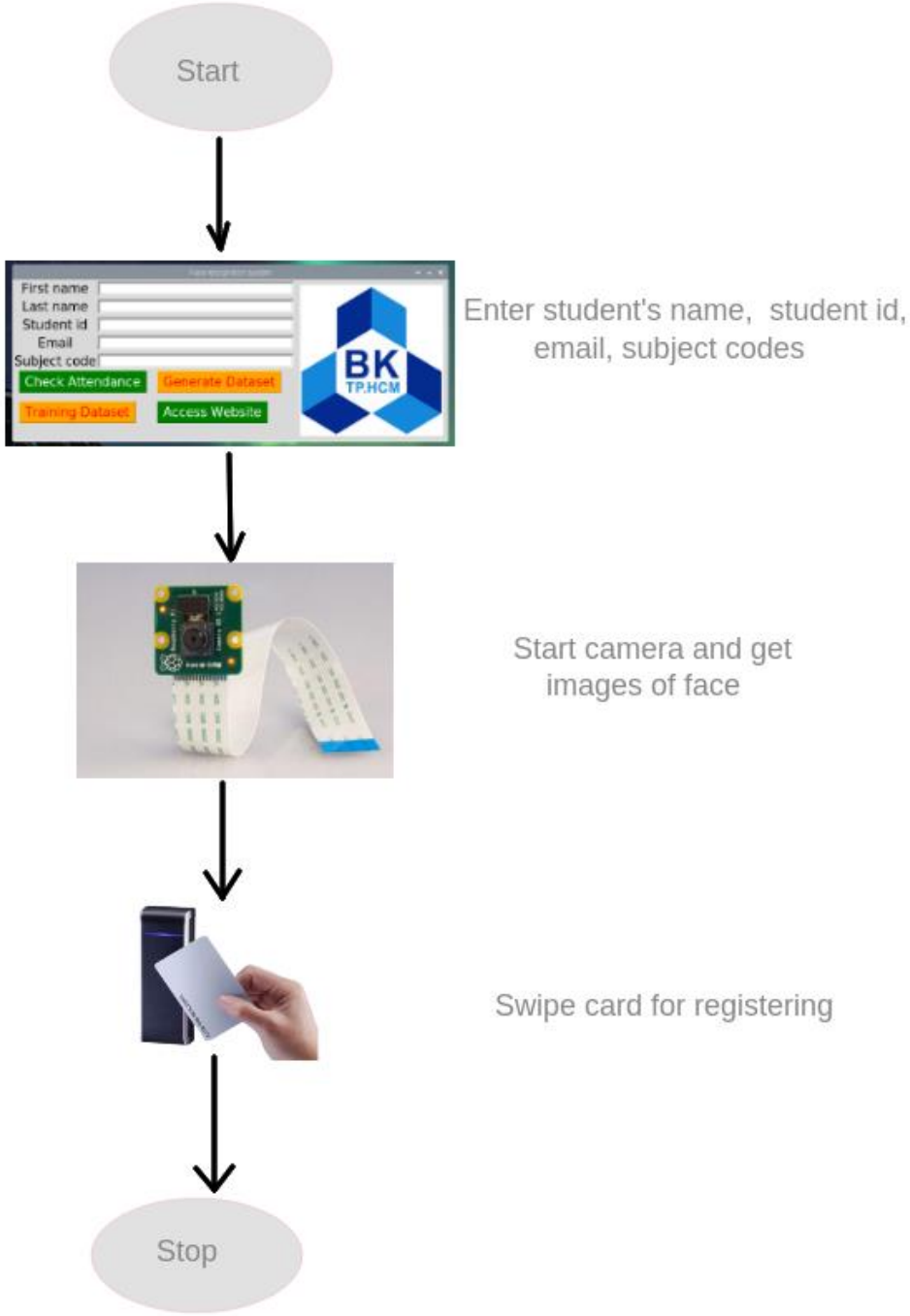
SEND MAIL

```
to = email
gmail_user = 'attendancesystembku@gmail.com'
gmail_pwd = '!attendancesystem'
smtpserver = smtplib.SMTP("smtp.gmail.com",587)
smtpserver.ehlo()
smtpserver.starttls()
smtpserver.ehlo
smtpserver.login(gmail_user, gmail_pwd)
date = datetime.datetime.now().strftime( "%d/%m/%Y %H:%M" )
header = 'To: ' + to + '\n' + 'From: ' + gmail_user + '\n' + 'Subject: Check attendance completed\n'
body = '\nWelcome ' + first_name + ',\n\nYour attendance is marked at: ' + date + '\n\nSubject: ' + subject_name +' (' +
footer = '\n\n' + day_in_week + ' From: ' +start_time + ' To: ' + end_time + ' Room: ' + room
msg = header  + body + footer
print(msg)
smtpserver.sendmail(gmail_user, to, msg)
print('sent mail\n_________________________________')
smtpserver.close()
```

SEND MAIL


Check attendance completed
attendancesystembku@gmail.com
tới tôi
Tiếng Anh
Tiếng Việt
Dịch thư
Welcome Thanh,
Your attendance is marked at: 23/11/2020 10:58
Subject: Microprocessors-microcontrollers (CO3009) Class: CC01
Monday From: 09:00 To: 11:50 Room: 303B4
Trả lời
Chuyển tiếp

REGISTER


Start
First name
Last name
Student id
Email
Subject code
Check Attendance
Generate Dataset
Training Dataset
Access Website
BK
TP.HCM
Enter student's name, student id, email, subject codes
Start camera and get images of face
Swipe card for registering
Stop

WEB APP

WEB APP

- Html, CSS, JavaScript, Ajax, Bootstrap, etc.
- 3 roles: admin, teacher, student
- Data charts, search bar
- Login, logout with session
- Registration
- Forgot password, confirm via email token

localhost/webapp/register.php

Register

Pill this form to create an account.

First Name

Enter First Name

Last Name

Enter Last Name

Username

Enter Username

Email

Enter Email

Password

Enter password

Confirm Password

Enter Confirm password

Register

You already have an account? Sign in now.

localhost/webapp/login.php

Login

Please fill in your credentials to login.

Username

Password

Login

Don't have an account? Sign up now.

Forgot Your Password?

localhost/webapp/forgot_password.php

Change Password

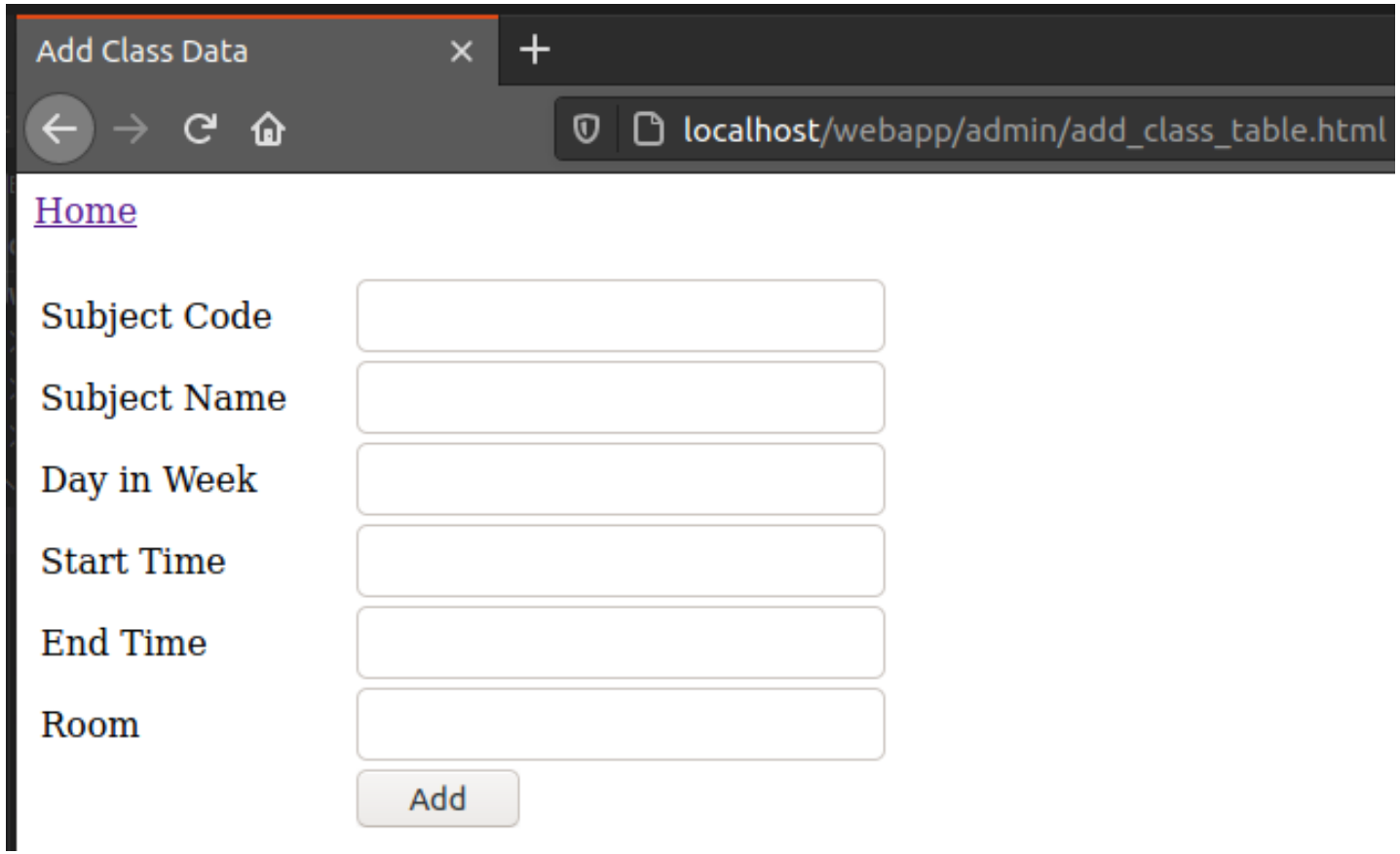
Email

Enter Email

Change your password

You already have an account? Sign in now.

ADMIN


Add Class Data
localhost/webapp/admin/add_class_table.html
Home
Subject Code
Subject Name
Day in Week
Start Time
End Time
Room
Add

TEACHER

Attendance Table

Add New Attendance Data

View Class Members

Gprahical Results

Search 2020-11-02 filter

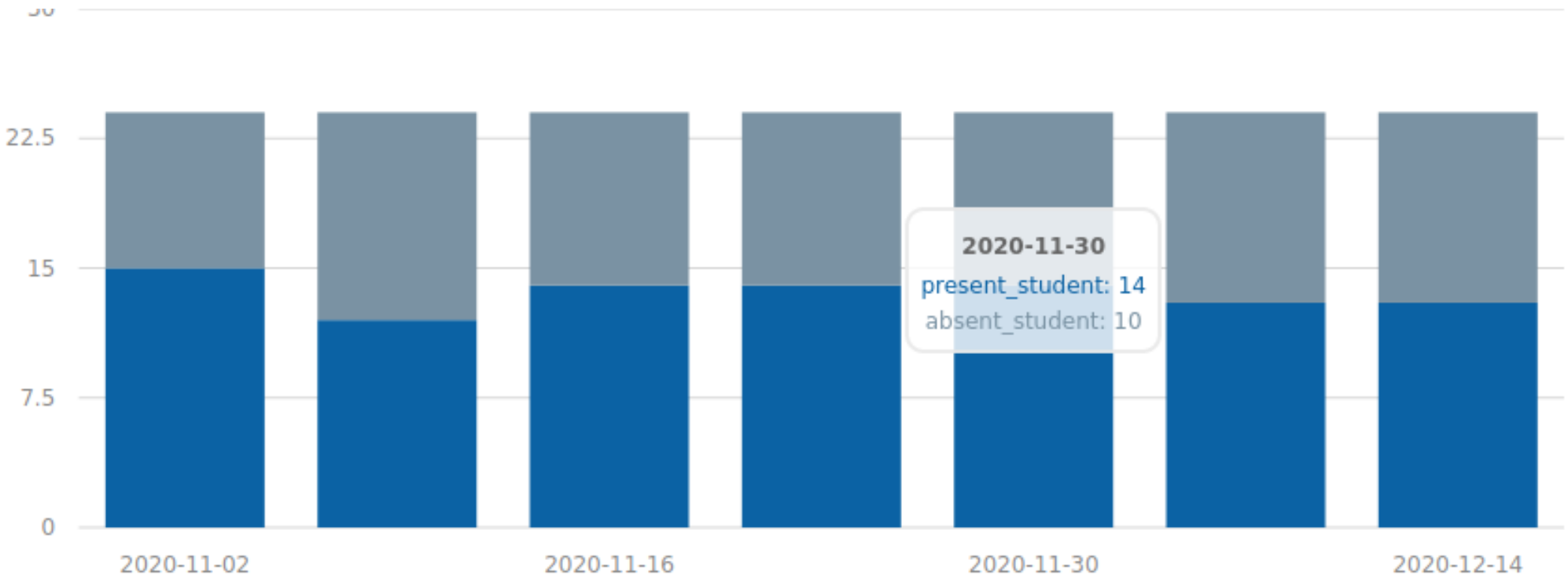
ID	First Name	Last Name	Student Number	Class Number	Clock In	Update
1	Thanh	Nguyen Truong	1652557	8	2020-11-02 09:46:00.000000	Edit | Delete
175	Quan	Nguyen Anh	1552310	8	2020-11-02 09:46:00.000000	Edit | Delete
182	Long	Nguyen Thanh	1655353	8	2020-11-02 09:46:00.000000	Edit | Delete
188	Hang	Nguyen Thi Thu	1656868	8	2020-11-02 09:46:00.000000	Edit | Delete
194	Anh	Le Thi Dieu	1651000	8	2020-11-02 09:46:00.000000	Edit | Delete
199	An	Tran Thien	1651001	8	2020-11-02 09:46:00.000000	Edit | Delete
205	Dang	Nguyen Hai	1651002	8	2020-11-02 09:46:00.000000	Edit | Delete
212	Dat	Nguyen Thanh	1651003	8	2020-11-02 09:46:00.000000	Edit | Delete
219	Hung	Nguyen Manh	1651004	8	2020-11-02 09:46:00.000000	Edit | Delete
225	Khoa	Nguyen Dang	1651005	8	2020-11-02 09:46:00.000000	Edit | Delete
231	Kiet	Nguyen Tuan	1651006	8	2020-11-02 09:46:00.000000	Edit | Delete
238	Dan	Nguyen Ngoc Linh	1651007	8	2020-11-02 09:46:00.000000	Edit | Delete
244	Diep	Le Ngoc	1651008	8	2020-11-02 09:46:00.000000	Edit | Delete
251	Dung	Nguyen Ngoc Nghi	1651009	8	2020-11-02 09:46:00.000000	Edit | Delete
258	Nhat	Nguyen An	1651010	8	2020-11-02 09:46:00.000000	Edit | Delete

TEACHER

Absent students

ID	First Name	Last Name	Student Number	Email	RFID_UID	Class List	Created
16	Nhat	Nguyen Tran Bao	1651011	1651011@hcmut.edu.vn	454646321131316	8	2020-11-09 14:46:44
17	Thanh	Le An	1651012	1651012@hcmut.edu.vn	776631913749421	8	2020-11-09 14:46:44
18	Nguyen	Do Cao	1651013	1651013@hcmut.edu.vn	9173913794214	8	2020-11-09 14:46:44
19	Sang	Nguyen Quang	1651014	1651014@hhcmut.edu.vn	686382140149174	8	2020-11-09 14:46:44
20	Minh	Nguyen Ngoc	1651016	1651016@hcmut.edu.vn	8683197359735	8	2020-11-09 14:46:44
21	Chau	Nguyen Ngoc	1651017	1651017@hcmut.edu.vn	8997993113752	8	2020-11-09 14:46:44
22	Hoang	Le Huy	1651018	1651018@hcmut.edu.vn	87197497249454	8	2020-11-09 14:46:44
23	Cat	Nguyen Ngoc Gia	1651018	1651018@hcmut.edu.vn	8082474279147	8	2020-11-09 14:46:44
24	Cuong	Nguyen Duy	1651019	1651019@hcmut.edu.vn	97359752759725	8	2020-11-09 14:46:44

ATTENDANCE CHART



Back to Results

EVALUATION AND TESTING

Testing

Evaluation

TESTING

TESTING

Time (s)	1	2	3	4	5	6	7	8	9	10	Average	Accuracy
Register	87	93	86	72	100	69	91	110	98	105	91.1	\
Correct all	56	58	43	61	63	54	59	60	64	62	58	47/50 = 94 %
Wrong timetable	50	41	48	42	39	46	43	39	49	34	43.2	49/50 = 98 %
Other face in db	32	30	34	42	37	43	34	30	41	29	35.2	46/50 = 92 %
Unknown face	30	32	37	40	33	44	45	51	38	33	38.3	49/50 = 98 %
Unregistered card	26	25	31	28	27	29	31	28	25	29	27.9	50/50 = 100 %
Fake image	75	78	80	77	81	79	86	76	84	87	80.3	50/50 = 100 %

EVALUATION

EVALUATION

Advantages

- The system is easy to setup
- Have all a checking attendance system need
- Have data chart
- Low cost

Disadvantages

- Timing constraint
- Low capacity of RAM
- User interface is not good
- Localhost

CONCLUSION

Contributions

Result

Future Works

CONTRIBUTIONS

- Give a solution to check student attendance using IoT devices.
- Combine RFID and Facial recognition
- Detect fake image of faces

RESULT

FUTURE WORK

- Setup web server and database server to another hosting
- Move the project to the stronger system
- Use other methods to improve the efficiency of anti-spoofing to replace eye blink detection
- Create a better user interface
- Improved code logic of the system

THANK YOU
ANY QUESTION?