

# **MAIN PROJECT REPORT ON ATTENDANCE AUTOMATION USING FACE RECOGNITION**

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR  
THE AWARD OF THE  
DEGREE IN  
BACHELOR OF COMPUTER APPLICATIONS  
OF

**MAHATMA GANDHI UNIVERSITY  
KERALA**

*Submitted by*

**K M FARIS MOHAMED**

**ROLL NO: 170134**



**DEPARTMENT OF COMPUTER APPLICATIONS  
(2019-2020)**

**M.E.S. COLLEGE MARAMPALLY**

**ALUVA -7**

# **M E S COLLEGE, MARAMPALLY**

## **ALUVA-7**



### **DEPARTMENT OF COMPUTER APPLICATIONS**

#### *Certificate*

This to certify that the report entitled

#### **ATTENDANCE AUTOMATION USING FACE RECOGNITION**

Has been submitted by

**K M FARIS MOHAMED**

ROLL NO: 170134

In partial fulfillment of the award of the degree in

#### **BACHELOR OF COMPUTER APPLICATION**

OF

#### **MAHATMA GANDHI UNIVERSITY**

During the academic year 2019-2020

Register No: 170021096202

Project Guide

Head of the

Department

Submitted for the examination held on .....

Examiners

- 1.
- 2.

## **ACKNOWLEDGEMENT**

At the very outset I would like to give first honour to GOD who gave the wisdom and knowledge to complete this project.

My extreme thanks goes to the **PRINCIPAL, Dr. A. BIJU** for providing the necessary facilities for the completion of this project work in my college.

I sincerely extend our thanks to **Dr. Murugan R**, Associate Professor and Head of the Computer Applications Department for all the help.

I wish to extend our sincere thanks to my guide **Mr. Joseph Deril**, Assistant Professor, Department of Computer Applications for his valuable guidance and support. I thank them for the timely suggestions and constant encouragement that boosted up my morale and led to accomplishment of this project.

I finally like to thank my parents, my friends and all my well-wishers who supported me directly and indirectly during my project.

**K M FARIS MOHAMED**

# Contents

<b>1. INTRODUCTION</b>	<b>6</b>
1.1 Overview of the system	<b>Error! Bookmark not defined.</b>
1.2 Problem definition and objective	7
<b>2. REQUIREMENT ANALYSIS</b>	<b>8</b>
2.1 Problem definition	9
2.2 Software development model.	9
2.3 Requirement specification includes existing and proposed system	9
2.3.1 Existing System	9
2.3.2 Proposed System	10
2.4 Project Planning	10
2.5 Project Scheduling	11
2.6 Feasibility Study	11
2.6.1 Economic Feasibility	11
2.6.2 Technical Feasibility	12
2.6.3 Operational Feasibility	12
<b>3. SOFTWARE REQUIREMENT SPECIFICATION (SRS)</b>	<b>13</b>
3.1 Introduction	14
3.1.1 Purpose	14
3.1.2 Scope	14
3.1.3 Definitions, acronyms, and abbreviations	14
3.1.4 References	15
3.1.5 Overview	15
<b>3.2 Overall description</b>	<b>15</b>
3.2.1 Product perspective	15
3.2.2 Product functions	16
3.2.3 User characteristics	16
3.2.4 Constraints	16
3.2.5 Assumptions and dependencies	16
<b>3.3 Specific requirements</b>	<b>17</b>
3.3.1 External interfaces	17

3.3.2	Functional requirements	18
3.3.3	Performance requirements	18
3.3.4	Logical database requirements	18
3.3.5	Design constraints	19
3.3.6	Software system attributes	19
3.3.7	Organizing the specific requirements	19
<b>4.</b>	<b>SOFTWARE AND HARDWARE REQUIREMENT</b>	<b>20</b>
4.1.	Software Requirements	21
4.2.	Hardware Requirements	21
<b>5.</b>	<b>SYSTEM DESIGN</b>	<b>22</b>
5.1	Introduction	23
5.2	Input Design	24
5.3	Output Design	26
5.4	Number of modules and their description	26
5.5	Functional Diagram	27
5.6	Database design	28
5.7	E R Diagram	29
5.8	Data Flow Diagram	30
<b>6.</b>	<b>SYSTEM DEVELOPMENT</b>	<b>32</b>
<b>7.</b>	<b>VALIDATION CHECKS</b>	<b>39</b>
<b>8.</b>	<b>SYSTEM IMPLEMENTATION</b>	<b>41</b>
<b>9.</b>	<b>FUTURE SCOPE</b>	<b>43</b>
9.1	FUTURE SCOPE OF THE PROJECT	44
<b>10.</b>	<b>CONCLUSION</b>	<b>45</b>
10.1	Conclution	46
<b>11.</b>	<b>APPENDIX</b>	<b>47</b>
<b>12.</b>	<b>BIBLIOGRAPHY</b>	<b>51</b>
12.1	Bibliography	52

## **1. INTRODUCTION**

## **1.1 Overview of the system**

Attendance automation has become one of the most important needs in educational institutions and work places across the world, since it saves time and accurate too. Face recognition system needs least human cooperation and is viable too. The system automatically detects the student's entry in the class and marks attendance for the particular student periodically. The data collected can be used by the system further for attendance score calculation and other managerial decisions. Arduino is used to create and control the system that could automatically mark the attendance for the students. Thus the system reduces the manual collection of attendance and the time taken for report generation.

## **1.2 Problem definition and objective**

Attendances of every student are being maintained by every school, college and university. Empirical evidences have shown that there is a significant correlation between students attendances and their academic performances. There was also a claim stated that the students who have poor attendance records will generally link to poor retention. Therefore, faculty has to maintain proper record for the attendance .The manual attendance record system is not efficient and requires more time to arrange record and to calculate the average attendance of each student. Hence there is a requirement of a system that will solve the problem of student record arrangement and student average attendance calculation. One alternative to make student attendance system automatic is provided by facial recognition.

## **2. REQUIREMENT ANALYSIS**



## **2.1 Problem definition**

Attendances of every student are being maintained by every school, college and university. Empirical evidences have shown that there is a significant correlation between students attendances and their academic performances. There was also a claim stated that the students who have poor attendance records will generally link to poor retention. Therefore, faculty has to maintain proper record for the attendance. The manual attendance record system is not efficient and requires more time to arrange record and to calculate the average attendance of each student. Hence there is a requirement of a system that will solve the problem of student record arrangement and student average attendance calculation. One alternative to make student attendance system automatic is provided by facial recognition.

## **2.2 Software development model.**

I selected Prototyping model as the software development model. The prototyping model is a systems development method in which a prototype is built, tested and then reworked as necessary until an acceptable outcome is achieved from which the complete system or product can be developed. This model works best in scenarios where not all of the project requirements are known in detail ahead of time. It is an iterative, trial-and-error process that takes place between the developers and the users.

## **2.3 Requirement specification includes existing and proposed system**

### **2.3.1 Existing System**

The current existing system is just a CCTV camera that keeps recording everything in the college premises. It keeps the recorded footage in a storage place in the college for some days and get auto deleted. At many circumstances the concerned persons in-charge of the camera have to look in each frame if the person needs to find anything.

### 2.3.2 Proposed System

The proposed system is designed to eliminate all the disadvantages of the existing system. The proposed system is an interactive system, highly user friendly and designed exclusively for the authorized individuals.

The proposed system use reduces manpower effort, time wastage etc. The system is having many added advantages which are having higher number of economic features for user interface. The system modular structure will allow the system to pair with its other system for easy data transfer with high control levels in user hands. The system itself detects faces and give attendance for the respective students found in the class.

## 2.4 Project Planning

The approximate time and date planning is as follows:-

TASK	START DATE	DURATION	END DATE
Logic design	20-DEC-2019	2 DAYS	22-DEC-2019
System selection	23-DEC-2019	2 DAYS	23- DEC -2019
System study	24- DEC -2019	11 DAYS	31- DEC -2019
Initial report	01-JAN-2020	6 DAYS	06-JAN-2020
Form design	07- JAN-2020	10 DAYS	17- JAN-2020
Coding	17- JAN-2020	20 DAYS	07-FEB-2020
SQL	08- FEB-2020	7 DAYS	15- FEB-2020
Testing	16- FEB-2020	5 DAYS	20- FEB-2020
Documentation	21- FEB-2020	10 DAYS	03- FEB-2020

## 2.5 Project Scheduling

A Gantt chart, visual; view of tasks scheduled over time, is used for planning projects and they are a useful way of showing what work is scheduled to be done on a specific time.

Task	DEC	JAN	FEB	MAR
Requirement gathering				
System analysis				
System design				
Form design				
Coding				
Testing				
Build				

## 2.6 Feasibility Study

The purpose of feasibility study is to investigate the present system, evaluate the cost and effectiveness of the proposed system, evaluate the possible applications of computer based methods, select a tentative system, evaluate the impact of the proposed system on existing personnel and ascertain the need for new personnel.

### 2.6.1 Economic Feasibility

Economic feasibility study presents tangible and intangible benefits from the project by comparing the development and operational cost. The technique of cost benefit analysis is used as a basis of attaining economic feasibility.

The proposing system is very economically feasible. The hardware used for the propose is already installed in the college premises. The software is made using the open sourced libraries and languages. The system needs some training datasets to train itself, this where it might need economical support required.

### **2.6.2 Technical Feasibility**

The system is aimed to be constructed in a way that it can be easily added to the existing system. Other than a high performance system to read the footage and run the system code.

### **2.6.3 Operational Feasibility**

Since the software is designed in a user friendly manner. Training is provided for the users of this software. It seems that the system can be used if it is developed and implemented. So the system is operationally feasible.

### **3. SOFTWARE REQUIREMENT SPECIFICATION (SRS)**

## **3.1 Introduction**

### **3.1.1 Purpose**

The purpose of this document is to describe the System. This document contains the functional, behavioral and non-functional requirements of the project and it also contains the guidelines for system engineers and designers to start working the project. The main purpose of this project is to reduce and automate most of the work in the college/university.

### **3.1.2 Scope**

The project is developed mainly for the use of the college/university authority. This project will work as a complete user interface. This system can be implemented under various conditions. We can add new information and when we require making reusability, it is possible as there is flexibility in all the modules. The language used for developing this project is python as backend which is very efficient in storing and retrieving the information as needed.

### **3.1.3 Definitions, acronyms, and abbreviations**

HTML – Hyper Text Markup Language

SQL – Structured Query Language

ER – Entity Relationship

IDE – Integrated Development Environment

SRS – Software Requirement Specification

CSS–Cascading Style Sheet

OpenCV – Open Computer Vision

HTML – Hyper Text Markup Language

IDE – Integrated Development Environment

SRS – Software Requirement Specification

CSS–Cascading Style Sheet

### **3.1.4 References**

- Roger S Pressman - Software Engineering: A Practitioner's Approach, Sixth Edition, McGraw-Hill Higher Education.
- Ian Sommerville - Software Engineering , Seventh Edition, Pearson Education.
- Pankaj Jalote - An Integrated approach to Software Engineering, Second Edition, Narosa Publishing Company.

### **3.1.5 Overview**

The document contains the detailed documentation of the requirements and functions of 'Attendance automation using face recognition' like product function, functional requirements, constraints, dependencies etc. The SRS is organized as its discuss the product description first then the requirements of the products, and the dependencies that will face by the systems etc. The developer is responsible for:

- Installing the software.
- Maintaining the system.
- Developing the system.

## **3.2 Overall description**

### **3.2.1 Product perspective**

Attendance automation using face recognition is an add-on Feature that can be incorporated to an existing feature or equipment in the college. This system would help the college to reduce the traditional attendance taking methods .

### **3.2.2 Product functions**

The main purpose of this system is to reduce complexity of existing system. This system is able to find the students in the classroom and mark attendance for those who are present in the class

### **3.2.3 User characteristics**

The system is will be used by an existing camera in the college. The footages collected by the camera will undergo through the algorithm of OpenCV libraries and finding a match, it marks attendance for those students present in the class. Here the authority/teacherse becomes the user who can view this live and in case of any exception they can manipulate this(give attendance) with the permission of supper user.

### **3.2.4 Constraints**

The type and positioning of the camera is important to meet the objective System must be installed in an environment with suitable hardware requirements

### **3.2.5 Assumptions and dependencies**

#### **The assumptions are:**

The coding should be error free. The system should be user friendly so that the users can easily access data which has more storage capacity and provide fast access to the database. Search facilities and provide quick search results. Save money and time unlike the existing system.

#### **The dependencies are:**

- The specific hardware and software are required for the product to will run
- On the basis of listing requirements and specification, the system will be developed and run
- Updates are to be made correctly and data entered without any mistakes



### **3.3 Specific requirements**

#### **3.3.1 External interfaces**

##### **GUI**

The software provides good graphical interface for the user and the administrator can operate on the system, performing the required task such as create, update, viewing the details of the Program list, Registration of students, Adding L2 users etc. It allows user to view scoreboard, participants list, schedules and venues, rules etc. The user interface must be customizable by the administrator. All the modules provided with the software must fit into this graphical user interface and accomplish to the standard defined. The design should be simple and all the different interfaces should follow a standard template

##### **Login interface**

The system is provided with a user name and password for L1 and L2. If the user gives an incorrect user name or password, an error message occurs.

##### **Search interface**

The authorized person can search for a particular detail which he is interested in. The search can be done using id or name

##### **Report**

The user can generate different reports like Attendance list, etc.

##### **Hardware Interfaces**

Only the recommended configuration (basic requirements of a computer system) no other specific hardware is required to run the software.

##### **Software Interfaces**

It is platform independent, any device with an active internet connection and a web browser can run this .

### **3.3.2 Functional requirements**

- Insert records: This action is done to add new records into fields .
- Update records: This event is to modify or update the information on each process.
- Delete records: This action is to remove records from the system whenever they are no longer needed.
- Search for records: Whenever the admin wants to search for a record, this action is performed.
  - The validation of data entered should be done.
  - Specific condition has to be met.

### **3.3.3 Performance requirements**

The proposed system that we are going to develop will be used as the Chief performance system. Therefore, it is expected that the database would perform functionally all the requirements that are specified by the college/university. The performance of the system should be fast and accurate. Adding/removing L2 users etc. Response of the system to an operation should be within reasonable time. The system should be able to handle large amount of data. The system should also be capable for updating program list, rules,

### **3.3.4 Logical database requirements**

A database requirement is dealing with the requirements for the collection of data. Some of major tasks, using a computer system are to store and manage data. To handle these tasks, you need a specialized computer program known as a database management system (DBMS). A DBMS stores, processes, and retrieves data. The database used in this project is SQL.

### **3.3.5 Design constraints**

The program is designed for every platform. Before accepting the system the developer will have to show through test cases that all conditions are satisfied.

### **3.3.6 Software system attributes**

- Reliability: The software should not have any reliability issues. The software will be thoroughly tested and any issues resolved.
- Availability: The software will execute as a standalone system so as long as the machine is running, the program will be available. The key to maintaining availability will be by ensuring a connection to the database server is available. Failure to connect to the database will make data unavailable.
- Security: This software is intended to communicate over an internal network; therefore security is of little concern. The user will have to enter the username and password so the program can connect to the database server. The username and password will not be stored because encryption of such information is outside the scope of the project.
- Maintainability: The software will be composed of various modules decreasing the complexity of expansion.
- Portability: As states previously, this software will only run under the Windows OS. The setup file, setup.info, can be copied to multiple machines so that each program does not have to be setup separately.

### **3.3.7 Organizing the specific requirements**

In this system the overall functionality is organized by Data flow diagrams and E-R diagrams. Based on these diagrams, data relationships and dependencies are found and a functional hierarchy is made for organizing the specific requirements.

## **4. SOFTWARE AND HARDWARE REQUIREMENT**

#### **4.1. Software Requirements**

Libraries required OpenCV, Pandas, Numpy, Python3

#### **4.2. Hardware Requirements**

NVIDIA GTX Graphics chip, Camera with high resolution

## **5. SYSTEM DESIGN**

## **5.1 Introduction**

The system will be installed as a software in a system that has access to the cameras. The software will take the footages recorded by the system and then run the OpenCV algorithms and codes over it. The code is designed to cut the footage in frame by frame and then looks for a match with the trained algorithm. On finding a match, the software recognizes the face and waits for the next frame result. Once the required number of frames has matched the face, attendance is provided to the corresponding student

## 5.2 Input Design

Student name	<input type="text"/>
Departement	<input type="text"/>
Year	<input type="text"/>
	<input type="button" value="Image"/>

<b>LOGIN</b>	
Username :	<input type="text"/>
Password	<input type="password"/>
	<input type="button" value="REGISTER"/> <input type="button" value="LOGIN"/>



Teacher ID:

Student ID:

Date:

☐ H1

☐ H2

☐ H3

☐ H4

☐ H5

☐ H6

Status

Absent



### 5.3 Output Design

Student name	<input type="text"/>
Class	<input type="text"/>

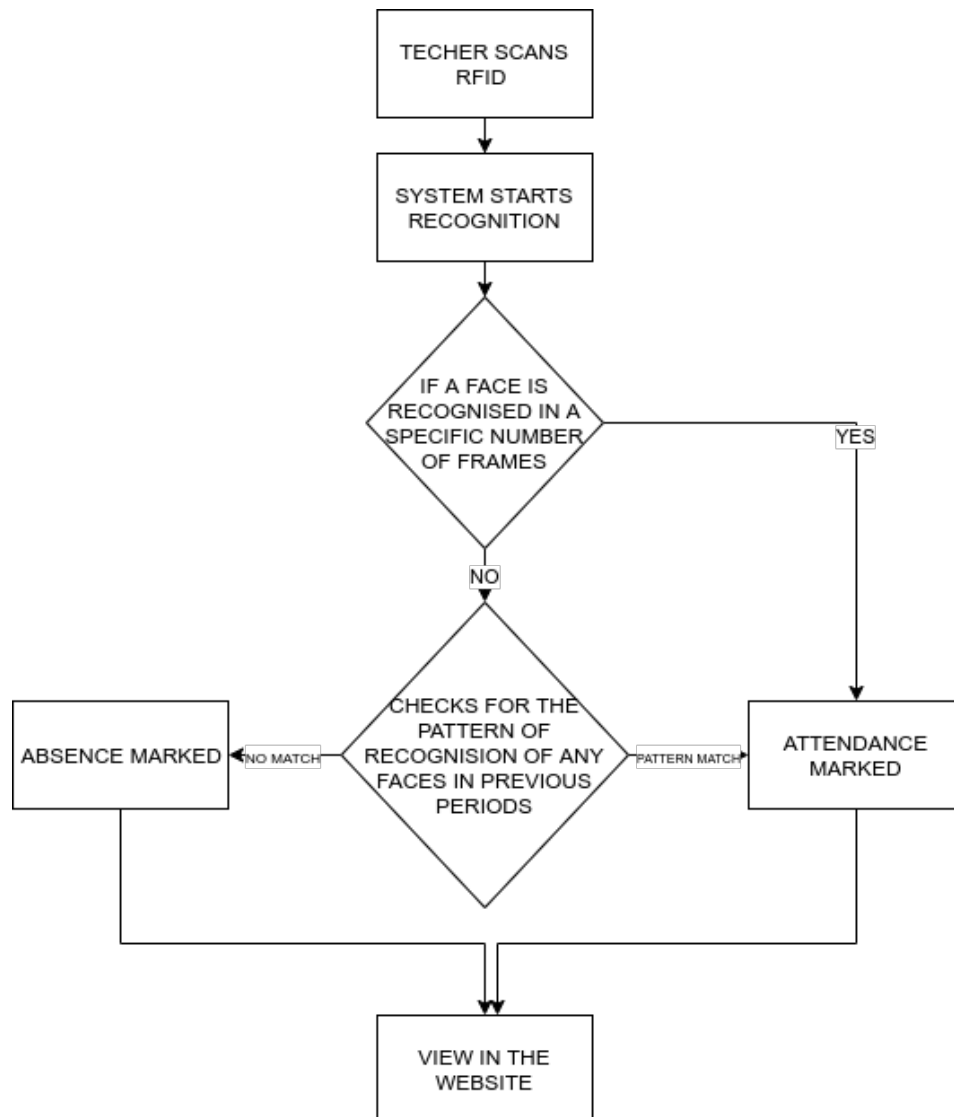
  

Date	Hour 1	Hour 2	Hour 3	Hour 4	Hour 5	Hour 6
02/03/2020	P	P	P	P	P	P
03/03/2020	A	A	A	A	A	A
04/03/2020	P	P	P	A	A	A

### 5.4 Number of modules and their description

The system has three modules, one is for detecting and recognizing faces from each frame of the live cctv footage. Second module evaluates the patterns in the recognition of a particular face in a frame using appropriate regression algorithms. The final module is for creating user interfaces to interact with the system, python will be used as the main language in the development..

## 5.5 Functional Diagram



## 5.6 Database design

**Table 1: Students**

Column	Type	Attributes	Null	Default	Extra
roll_number	int(11)		No		auto_increment
name	varchar(50)		No		
department	varchar(50)		No		
year	int(4)		No		
encoding	blob		Yes	NULL	

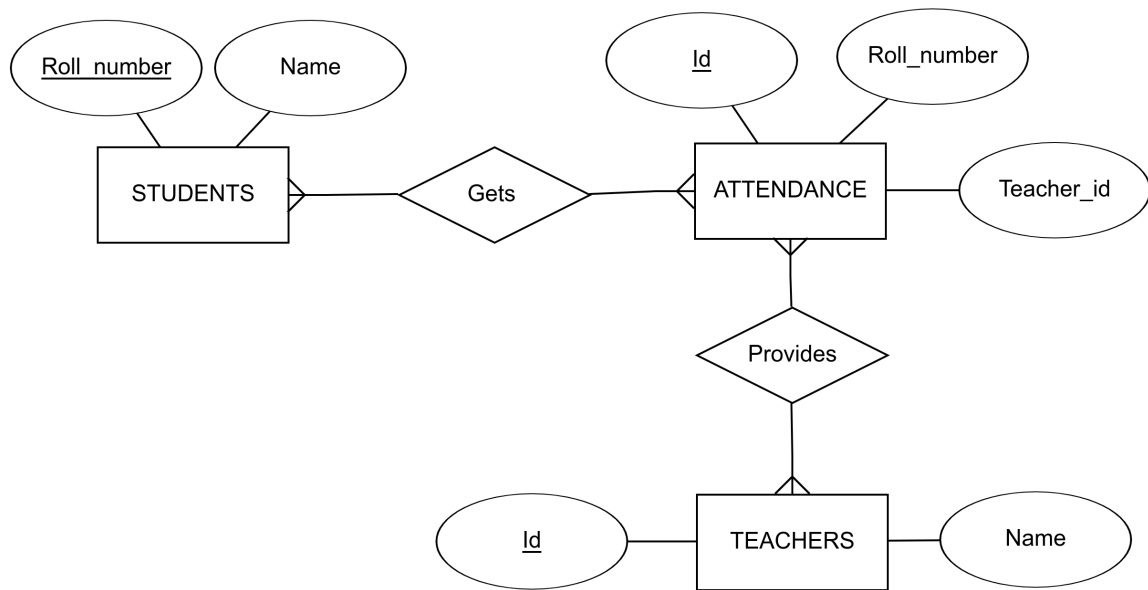
**Table 2: Attendance**

Column	Type	Attributes	Null	Default	Extra
id	int(11)		No		auto_increment
roll_number	int(11)		No		
date	date		No		
hour_1	tinyint(1)		No	0	
hour_2	tinyint(1)		No		
hour_3	tinyint(1)		No		
hour_4	tinyint(1)		No		
hour_5	tinyint(1)		No		
hour_6	tinyint(1)		No		
teacher_id	int(11)		No		

**Table 3: Teachers**

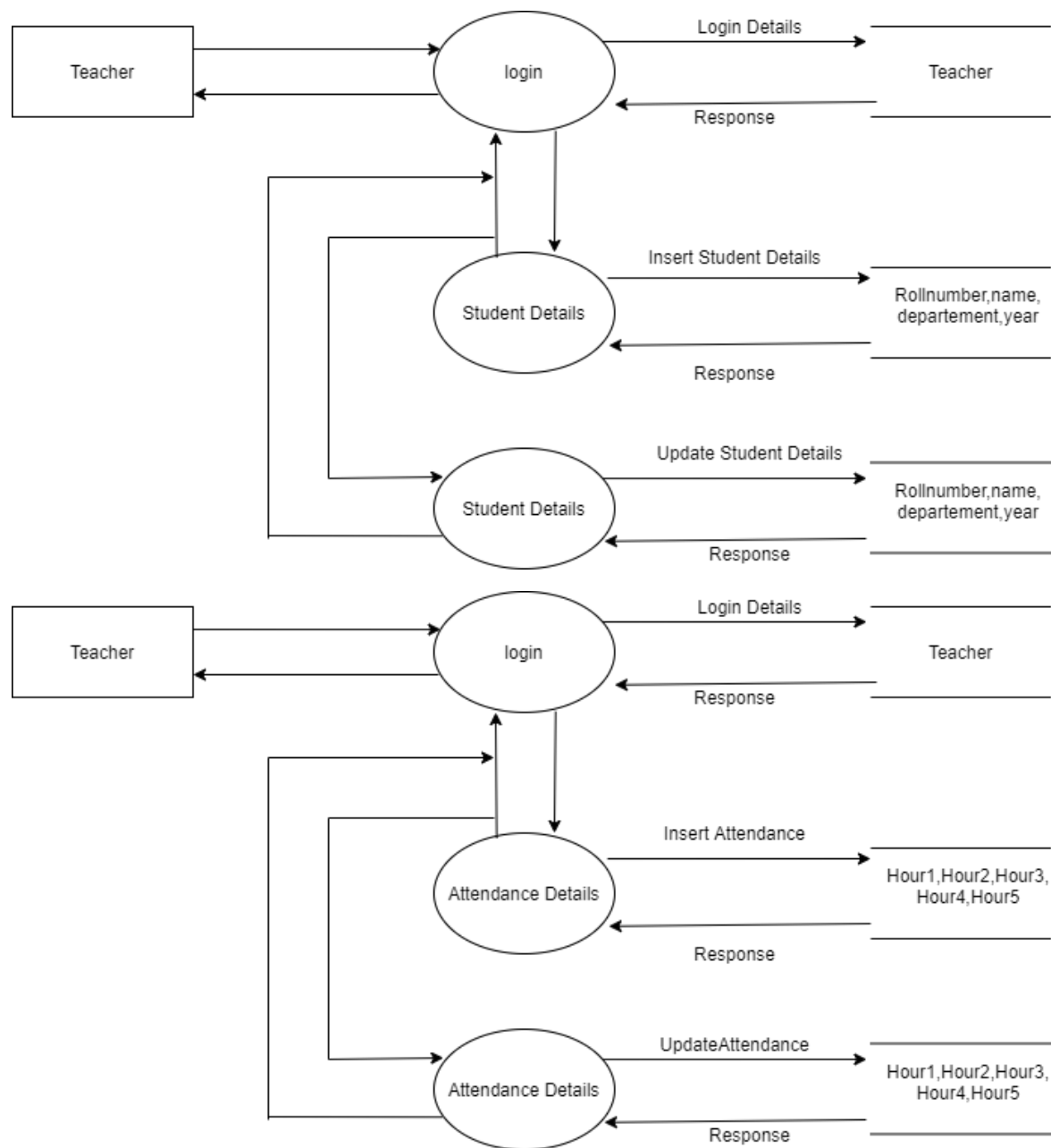
Column	Type	Attributes	Null	Default	Extra
id	int(11)		No		auto_increment
name	varchar(30)		No		
department	varchar(50)		No		
rfid	varchar(20)		No		

## 5.7 E R Diagram





## Level 2



## **6. SYSTEM DEVELOPMENT**



## 6.1 Process Description

The system consist of an RFID reader, in which the teacher scans their unique tag and the system starts the recognition for a predefined amount of time (for demo purpose it is set t one minute). The system scans each and every faces on each and every frames during the time interval. if a face is recognised for more than a threshold amount, attendance is marked for the corresponding hour. Teachers will also have an option to update attendance manually.

The System detects face on a video frame with the help of HOG algorithm. and once the face is detected, it uses convolutional neural networks to convert the faces to 128 feature encoding with the help of deep learning. The 128 encoding is based on the different features of the face. Then the system uses SVM classifiers to match the encoding with previously known encodings.

## 6.2 Pseudo code

### *recognise.py*

```
import mysql.connector
import face_recognition
import cv2
import numpy as np
import pickle
import datetime

def fetchActiveClass(x):
    return x

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="trajanco_face")
```

```

video_capture = cv2.VideoCapture(0)
mycursor = mydb.cursor()
mycursor.execute("SELECT * FROM students")
myresult = mycursor.fetchall()
known_face_encodings=[]
known_face_names=[]
known_face_id=[]
known_faces=[]
name=""
counter =0

for x in myresult:
    face_data = pickle.loads(x[4])
    known_face_encodings.append(face_data)
    known_face_names.append(x[1])
    known_face_id.append(x[0])
    known_faces.append(0)

if(fetchActiveClass(1)):
    limit = datetime.datetime.now() + datetime.timedelta(seconds = 60)

while(datetime.datetime.now()<=limit):
    # Grab a single frame of video

    ret, frame = video_capture.read()
    #print(ret)

    # Convert the image from BGR color (which OpenCV uses) to RGB color (which
    face_recognition uses)
    rgb_frame = frame[:, :, :-1]

    # Find all the faces and face enqcodings in the frame of video
    face_locations = face_recognition.face_locations(rgb_frame)

```

```

#print(face_locations)

face_encodings = face_recognition.face_encodings(rgb_frame, face_locations)

# Loop through each face in this frame of video
for (top, right, bottom, left), face_encoding in zip(face_locations, face_encodings):
    # See if the face is a match for the known face(s)
    matches = face_recognition.compare_faces(known_face_encodings, face_encoding)

    name = "Unknown"

    # If a match was found in known_face_encodings, just use the first one.
    # if True in matches:
    #     first_match_index = matches.index(True)
    #     name = known_face_names[first_match_index]

    # Or instead, use the known face with the smallest distance to the new face
    face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
    best_match_index = np.argmin(face_distances)
    if matches[best_match_index]:
        name = known_face_names[best_match_index]
        known_faces[best_match_index] = known_faces[best_match_index]+1

    # Draw a box around the face
    cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)

    # Draw a label with a name below the face
    cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255), cv2.FILLED)
    font = cv2.FONT_HERSHEY_DUPLEX
    cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)
    counter+=1

# Display the resulting image

```

```

cv2.imshow('Video', frame)

# print(name)

# Hit 'q' on the keyboard to quit!
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

finalArr=[]
index = 0
for x in known_faces:
    if(x/counter >= .2):
        finalArr.append(known_face_id[index])
        index+=1
# print(counter)
# print(finalArr)

for id in finalArr:
    sql = "INSERT INTO attendance (roll_number, hour_1, date) VALUES(%s,%s,%s)"
    val = (id, 1, "2020-03-06")
    mycursor.execute(sql, val)
    mydb.commit()

# Release handle to the webcam
video_capture.release()
cv2.destroyAllWindows()

```

### **models.py**

```

from django.db import models

# Create your models here.

class Department(models.Model):

```

```

department_name = models.CharField(max_length=25)

def __str__(self):

    return self.department_name


class Class(models.Model):

    department_id = models.ForeignKey(Department,on_delete=models.CASCADE)

    class_name = models.CharField(max_length=25)

    def __str__(self):

        return self.class_name


class Student(models.Model):

    class_id = models.ForeignKey(Class,on_delete=models.CASCADE)

    student_name = models.CharField(max_length=100)

    def __str__(self):

        return self.student_name


class Teacher(models.Model):

    class_id = models.ForeignKey(Class,on_delete=models.CASCADE)

    teacher_name = models.CharField(max_length=100)

    rf_id = models.CharField(max_length=100)

    def __str__(self):

        return self.teacher_name


class Attendance(models.Model):

    ATTENDENCE_STATUS = (

```

```

('A', 'Absent'),
('F', 'Full Day'),
('H', 'Half Day'),
)

teacher_id = models.ForeignKey(Teacher,on_delete=models.CASCADE)
student_id = models.ForeignKey(Student,on_delete=models.CASCADE)

date = models.DateField(blank=False)

h1 = models.BooleanField(default=False)
h2 = models.BooleanField(default=False)
h3 = models.BooleanField(default=False)
h4 = models.BooleanField(default=False)
h5 = models.BooleanField(default=False)
h6 = models.BooleanField(default=False)

attendance_status =
models.CharField(max_length=1,choices=ATTENDENCE_STATUS,default='A')

```

## **7. VALIDATION CHECKS**





## **8. SYSTEM IMPLEMENTATION**

## **8.1 Testing**

We have done unit testing, integration testing, performance testin, security testing etc. we also tested the deep learning networks with variety of image samples to verify the accuracy of the model.

## **8.2 System Implementation**

We are planning to implement the system on educational institutions to reduce the time wastage in attendance taking.

## **8.3 Security**

The system is well secure, and has secure authentication system for teachers and students. The IoT device uses unique rfid tags for increased security.

## **9. FUTURE SCOPE**

## **9.1 FUTURE SCOPE OF THE PROJECT**

The system promises of great chances in the future. The system can also add many other different features in it. If the system is trained enough it can be used to mark the attendance and also monitor the behavior of students inside the class. The system can be used to identify vehicles entered to the college similarly it can also find if any student of different class is sitting in the classroom. A finely trained system can be even used to monitor the students during exams, and prevent from cheating.

## **10. CONCLUSION**

## **10.1 Conclusion**

The system helps to reduce the efforts of taking attendance in a campus, also reducing the errors caused during the manual work. This feature of a simple camera opens a window to more possibilities to the college. This system has its possibilities in the industrial places also.

## **11. APPENDIX**

# 11.1 Sample Input

Change attendance | Django site

Change booking | Django site

Index-Doctor

pass value in url - Bing

127.0.0.1:8000/admin/automated\_attendance\_admin/attendance/2/change/

Django administration

WELCOME ADMIN VIEW SITE / CHANGE PASSWORD / LOG OUT

Home Automated Attendance Admin Attendances Attendance object (2)

Change attendance

HISTORY

Teacher id: Joseph Deril

Student id: Alna Surash

Date: 2020-03-09 Today

Note: You are 5.5 hours ahead of server time.

☒ H1

☒ H2

☐ H3

☒ H4

☐ H5

☒ H6

Attendance status: Half Day

Delete

Save and add another

Save and continue editing

SAVE

Change user | Django site admin

Change booking | Django site

Index-Doctor

pass value in url - Bing

127.0.0.1:8000/admin/auth/user/1/change/

Django administration

WELCOME ADMIN VIEW SITE / CHANGE PASSWORD / LOG OUT

Home Authentication and Authorization Users admin

Change user

HISTORY

Username: admin

Required: 150 characters or fewer: Letters, digits and @/./+/-/\_ only.

Password: algorithm: pbkdf2\_sha256 iterations: 180000 salt: rXBaR\*\*\*\*\* hash: 01b9F4\*\*\*\*\*

Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

Personal info

First name:

Last name:

Email address: pranaykprince@gmail.com

Permissions

☒ Active

Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☒ Staff status

Designates whether the user can log into this admin site.

☒ Superuser status

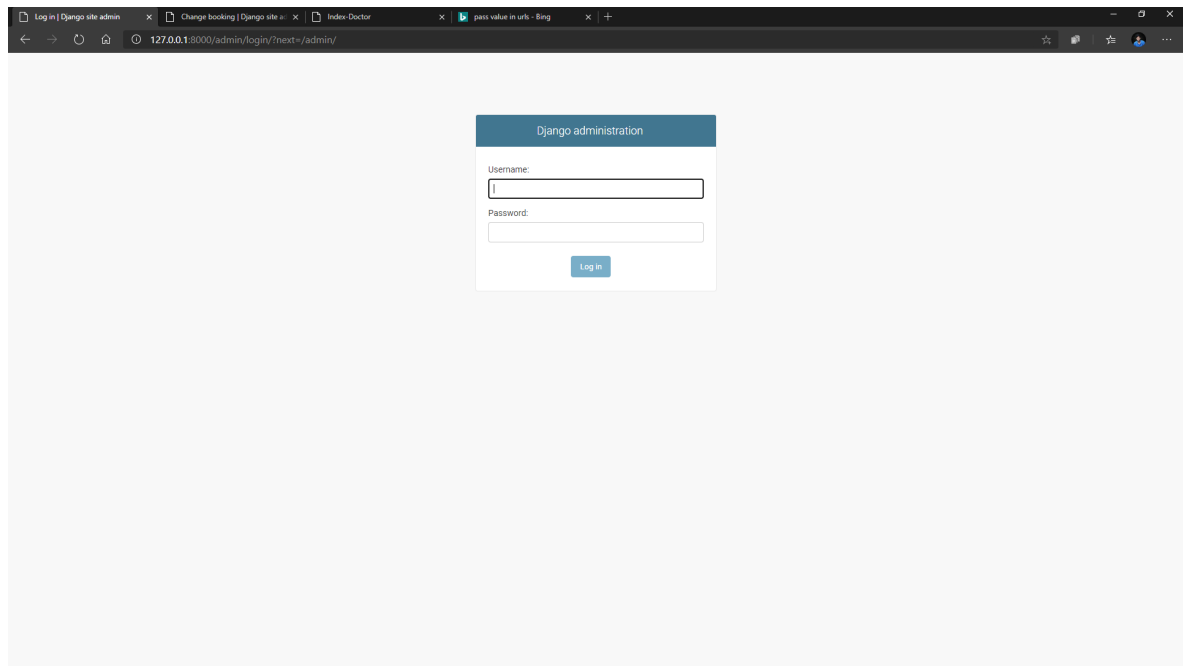
Designates that this user has all permissions without explicitly assigning them.

Groups:

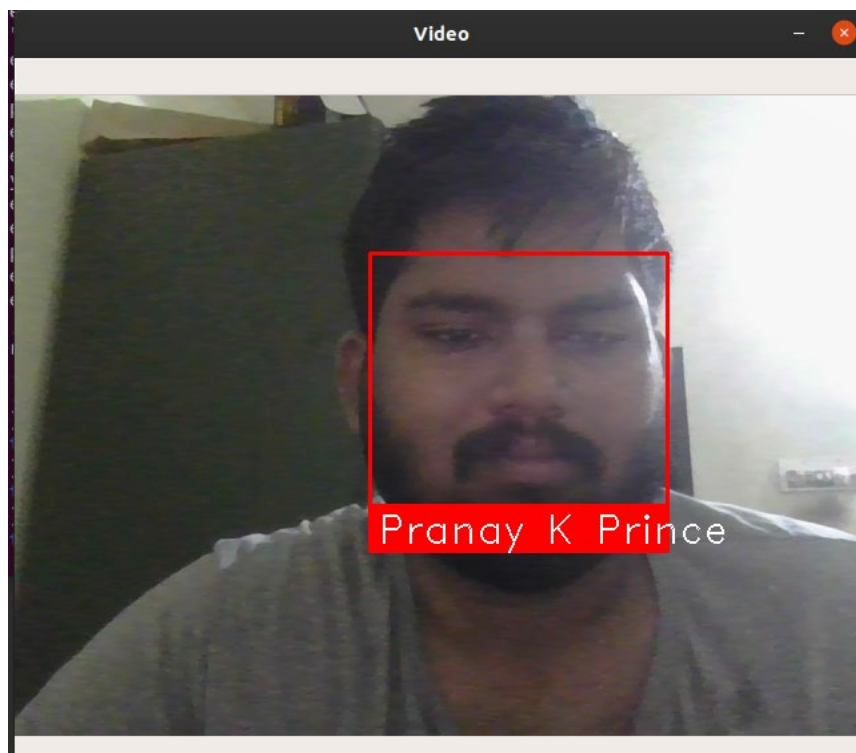
Available groups

Chosen groups





## 11.2 Sample Output



## Attendance

Rollno	Name	Date	Attendance
1	Adith Roshan	2019-12-07	Present
2	Pranay K Prince	2019-12-07	Present

## **12. BIBLIOGRAPHY**

## 12.1 Bibliography

<https://docs.opencv.org/>

[https://github.com/ageitgey/face\\_recognition](https://github.com/ageitgey/face_recognition)

<https://towardsdatascience.com/>

<https://colab.research.google.com/>