TABLE OF CONTENTS

CHAPTER

1 INTRODUCTION

1.1 Problem Statement

1.2 Aims and Objectives

2 LITERATURE REVIEW

2.1 Student Attendance System

2.2 Face Detection

3 METHODOLOGY

3.1 Methodology Flow

3.1.1 Limitations of the Images

3.2 Face Detection

3.2.1 Pre-Processing

4 LITERATURE SURVEY

4.1 TOOLS AND TECHNOLOGIES

4.1.1 PYTHON

4.1.2 EEL

4.1.3 PICKLE

4.1.4 BOTTLE

5 REQUIREMENTS

5.1 HARDWARE REQUIREMENTS

5.2 SOFTWARE REQUIREMENTS

6 REQUIREMENT ANALYSIS

6.2 Functional Requirements

6.1.1 Performance requirements

6.2.4 Software Quality Attributes

6.2.3 Security Requirements

6.1.2 Safety Requirements

7 SYSTEM DESIGN

7.1 SYSTEM PRESPECTIVE

7.1 COMPONENT DIAGRAM

8 DETAILED DESIGN

8.1 Use Case Diagram

8.2 Data Flow Diagram

8.3 Database Design

8.3.1.1 E-R Diagram

9 CODE SNIPPET

10.1.2 main py Code

10 TYPES OF TESTS

10.1 Introduction to software testing

10.2 Test Cases

11.3.1 ADMIN

11.3.2 Upload student Details

11.3.3 User

11.3.4 User Registration

ABSTRACT

Face is the representation of one's identity. Hence, we have proposed an automated student attendance system based on face recognition. Face recognition system is very useful in life applications especially in security control systems. The airport protection system uses face recognition to identify suspects and FBI (Federal Bureau of Investigation) uses face recognition for criminal investigations. In our proposed approach, firstly, video framing is performed by activating the camera through a userfriendly interface. The face ROI is detected and segmented from the video frame by using Viola-Jones algorithm. In the pre-processing stage, scaling of the size of images is performed if necessary in order to prevent loss of information. The median filtering is applied to remove noise followed by conversion of colour images to grayscale images. After that, contrast-limited adaptive histogram equalization (CLAHE) is implemented on images to enhance the contrast of images. In face recognition stage, enhanced local binary pattern (LBP) and principal component analysis (PCA) is applied correspondingly in order to extract the features from facial images.

In our proposed approach, the enhanced local binary pattern outperform the original LBP by reducing the illumination effect and increasing the recognition rate. Next, the features extracted from the test images are compared with the features extracted from the training images. The facial images are then classified and recognized based on the best result obtained from the combination of algorithm, enhanced LBP and PCA. Finally, the attendance of the recognized student will be marked and saved in the excel file. The student who is not registered will also be able to register on the spot and notification will be given if students sign in more than once. The average accuracy of recognition is 100 % for good quality images, 94.12 % of low-quality images and 95.76 % for Yale face database when two images per person are trained.

CHAPTER 1

INTRODUCTION

The main objective of this project is to develop face recognition based automated student attendance system. In order to achieve better performance, the test images and training images of this proposed approach are limited to frontal and upright facial images that consist of a single face only. The test images and training images have to be captured by using the same device to ensure no quality difference. In addition, the students have to register in the database to be recognized. The enrolment can be done on the spot through the user-friendly interface.

1.1 Problem Statement

Traditional student attendance marking technique is often facing a lot of trouble. The face recognition student attendance system emphasizes its simplicity by eliminating classical student attendance marking technique such as calling student names or checking respective identification cards. There are not only disturbing the teaching process but also causes distraction for students during exam sessions. Apart from calling names, attendance sheet is passed around the classroom during the lecture sessions. The lecture class especially the class with a large number of students might find it difficult to have the attendance sheet being passed around the class. Thus, face recognition student attendance system is proposed in order to replace the manual signing of the presence of students which are burdensome and causes students get distracted in order to sign for their attendance. Furthermore, the face recognition based automated student attendance system able to overcome the problem of fraudulent approach and lecturers does not have to count the number of students several times to ensure the presence of the students. The paper proposed by Zhao, W et al. (2003) has listed the difficulties of facial identification. One of the difficulties of facial identification is the identification between known and unknown images. In addition, paper proposed by Pooja G.R et al. (2010) found out that the training process for face recognition student attendance system is slow and timeconsuming. In addition, the paper proposed by Priyanka Wagh et al. (2015) mentioned that different lighting and head poses are often the problems that could degrade the performance of face recognition based student attendance system.

Hence there is a need to develop a real time operating student attendance system which means the identification process must be done within defined time constraints to prevent omission. The extracted features from facial images which represent the identity of the students have to be consistent towards a change in background, illumination, pose and expression. High accuracy and fast computation time will be the evaluation points of the performance.

1.2 Aims and Objectives

The objective of this project is to develop face recognition based automated student attendance system. Expected achievements in order to fulfill the objectives are:

- To detect the face segment from the video frame.
- To extract the useful features from the face detected.
- To classify the features in order to recognize the face detected.
- To record the attendance of the identified student.

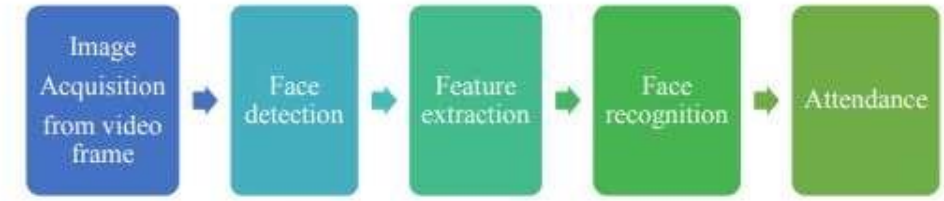


Figure 1.1 Block Diagram of the General Framework

CHAPTER 2

LITERATURE REVIEW

2.1 Student Attendance System

Arun Katara et al. (2017) mentioned disadvantages of RFID (Radio Frequency Identification) card system, fingerprint system and iris recognition system. RFID card system is implemented due to its simplicity. However, the user tends to help their friends to check in as long as they have their friend's ID card. The fingerprint system is indeed effective but not efficient because it takes time for the verification process so the user has to line up and perform the verification one by one. However for face recognition, the human face is always exposed and contains less information compared to iris. Iris recognition system which contains more detail might invade the privacy of the user. Voice recognition is available, but it is less accurate compared to other methods. Hence, face recognition system is suggested to be implemented in the student attendance system.

Table 2.1 Advantages & Disadvantages of Different Biometric System (Arun Katara et al., 2017)

System type	Advantages	Disadvantages
RFID card system	Simple	Fraudulent usage
Fingerprint system	Accurate	Time-consuming
Voice recognition system	-	Less accurate compared to others
Iris recognition system	Accurate	Privacy Invasion

2.2 Face Detection

Difference between face detection and face recognition are often misunderstood. Face detection is to determine only the face segment or face region from image, whereas face recognition is to identify the owner of the facial image. S.Aanjanadevi et al. (2017) and Wei-Lun Chao (2007) presented a few factors which cause face detection and face recognition to encounter difficulties. These factors consist of background, illumination, pose, expression, occlusion, rotation, scaling and translation. The definition of each factor is tabulated in Table 2.2.

Table 2.2 Factors Causing Face Detection Difficulties (S.Aanjanadevi et al., 2017)

Background	Variation of background and environment around people in the image which affect the efficiency of face recognition.
Illumination	Illumination is the variation caused by various lighting environments which degrade the facial feature detection.
Pose	Pose variation means different angle of the acquired the facial image which cause distortion to recognition process, especially for Eigen face and Fisher face recognition method.
Expression	Different facial expressions are used to express feelings and emotions. The expression variation causes spatial relation change and the facial-feature shape change.
Occlusion	Occlusion means part of the human face is unobserved. This will diminish the performance of face recognition algorithms due to deficiency information.
Rotation, scaling and translation	Transformation of images which might cause distortion of the original information about the images.

There are a few face detection methods that the previous researchers have worked on. However, most of them used frontal upright facial images which consist of only one face. The face region is fully exposed without obstacles and free from the spectacles.

Akshara Jadhav et al. (2017) and by P. Arun Mozhi Devan et al. (2017) suggested Viola-Jones algorithm for face detection for student attendance system. They concluded that out of methods such as face geometry- based methods, Feature Invariant methods and Machine learning based methods, Viola-Jones algorithm is not only fast and robust, but gives high detection rate and perform better in different lighting condition. Rahul V. Patil and S. B. Bangar (2017) also agreed that Viola-Jones algorithm gives better performance in different lighting condition. In addition, in the paper by Mrunmayee Shirodkar et al. (2015), they mentioned that Viola-Jones algorithm is able to eliminate the issues of illumination as well as scaling and rotation.

Table 2.3 Advantages & Disadvantages of Face Detection Methods (Varsha Gupta and Dipesh Sharma, 2014)

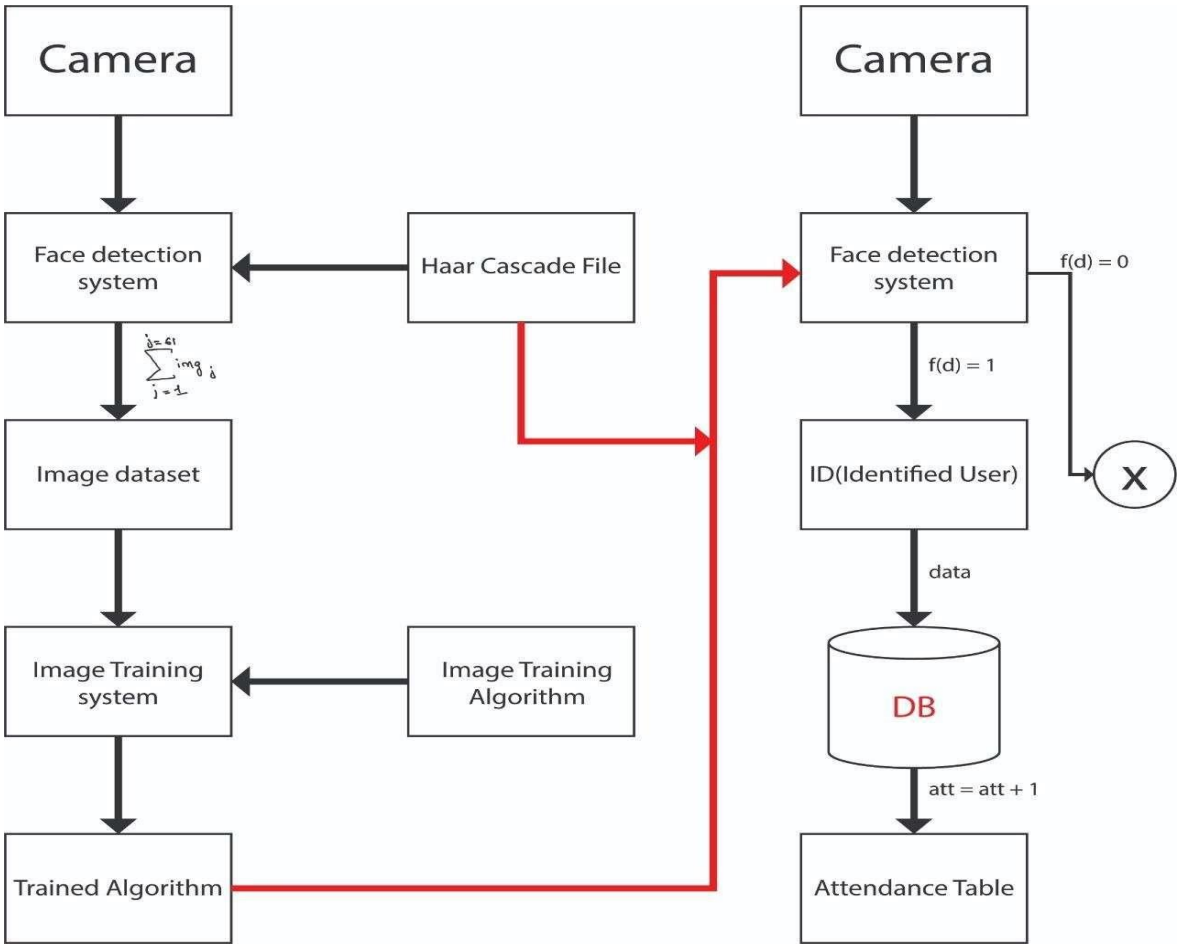
Face detection method	Advantages	Disadvantages
Viola jones algorithm	1. High detection speed 2. High accuracy.	1. Long training time. 2. Limited head pose. 3. Not able to detect dark faces.
Local Binary pattern	1. Simple computation. 2. High tolerance against the monotonic illumination changes.	1. Only used for binary and grey images. 2. Overall performance is inaccurate compared to Viola-Jones algorithm.
AdaBoost algorithm (part of Viola jones algorithm)	Need not to have any prior knowledge about face structure.	The result highly depends on the training data and affected by weak classifiers.
SMQT Features and SNOW Classifier Method	1. Capable to deal with lighting problem in object detection. 2. Efficient in computation.	The region contain very similar to grey value regions will be misidentified as face.
Neural-Network	High accuracy only if large size of image were trained.	1. Detection process is slow and computation is complex. 2. Overall performance is weaker than Viola-Jones algorithm.

Chapter 3

3.1 Methodology Flow

The approach performs face recognition based student attendance system. The methodology flow begins with the capture of image by using simple and handy interface, followed by pre-processing of the captured facial images, then feature extraction from the facial images, subjective selection and lastly classification of the facial images to be recognized. LBP feature extraction methods are studied in detail and computed in this proposed approach in order to make comparisons. LBP is enhanced in this approach to reduce the illumination effect. An algorithm to c enhanced LBP is also designed for subjective selection in order to increase the accuracy. The details of each stage will be discussed in the following sections.

The flow chart for the proposed system is categorized into two parts, first training of images followed by testing images (recognize the unknown input image) shown in Figure 3.1 and Figure 3.2 respectively.



3.4 Face Detection

Viola-Jones object detection framework will be used to detect the face from the video camera recording frame. The working principle of Viola-Jones algorithm is mentioned in Chapter 2. The limitation of the Viola-Jones framework is that the facial image has to be a frontal upright image, the face of the individual must point towards the camera in a video frame.

3.4.1 Pre-Processing

Testing set and training set images are captured using a camera. There are unwanted noise and uneven lighting exists in the images. Therefore, several pre-processing steps are necessary before proceeding to feature extraction.

Pre-processing steps that would be carried out include scaling of image, median filtering, conversion of colour images to grayscale images and adaptive histogram equalization. The details of these steps would be discussed in the later sections.

Chapter 4

TOOLS AND TECHNOLOGIES

4.1 PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including objectoriented, imperative, functional and procedural, and has a large and comprehensive standard library

Advantages:

1. **Easy-to-learn and Easy-to-use:-**

Python programming language has a syntax similar to the English language, making it extremely easy and simple for anyone to read and understand its codes.

2. **Improves Productivity:-**

Another one of Python benefits is that it is an extremely productive language, and because of its simplicity, Python Programmers can easily focus on solving issues.

3. **Interpreted Language:-**

Python, being an interpreted language, can execute the code directly, one line after the other. Moreover, if there is any error, then rather than continuing with further execution, it instead reports back the error that occurred.

4. **Simple and Easy:-**

When working with Java, you may have to create a class to print ‘Hello World’. But in Python, just a print statement will do.

5. **Readable:-**

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code.

Disadvantages:

1. **Low Speed :-**

Strengths can, unfortunately, lead to some weaknesses at times. Here is such a case. Yes, Python is a dynamically-typed and interpreted language, but this means that the code is executed line-by-line, further leading to its slow execution.

2. **Inefficient Memory Consumption :-**

To offer some simplicity to programmers and developers, Python needs to make some tradeoffs. This language uses a huge amount of memory, which acts as a disadvantage, especially when you are developing an app with a preference for memory optimization.

3. **Weak in Programming for Mobile Devices:-**

Developers usually use Python for server-side programming, rather than using it for mobile applications or client-side programming. This is because Python has slow processing power and is hardly memory efficient when compared to other programming languages.

4.2 EEL

Eel is a Python library for making simple offline HTML/JS GUI apps, with full access to Python capabilities and libraries. Eel hosts a local webserver, then lets you annotate functions in Python so that they can be called from Javascript, and vice versa.

Getting Started

First, create a project folder and make another folder called web under it. The web folder consists of all the website files. Create a main.py python file outside the web folder inside project folder.

Files and Folders

To use eel in the frontend javascript. Include the following code in the HTML file :

<script type=”text/javascript” src=”/eel.js”></script>

Including this library creates an eel object which can be used to communicate with the Python side. Any functions in the Python code which are decorated with @eel.expose like this:

@eel.expose

def function():

4.3 PICKLE

Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it "serializes" the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

Advantages of using Pickle Module:

1. Recursive objects (objects containing references to themselves): Pickle keeps track of the objects it has already serialized, so later references to the same object won't be serialized again. (The marshal module breaks for this.)
2. Object sharing (references to the same object in different places): This is similar to self- referencing objects; pickle stores the object once, and ensures that all other references point to the master copy. Shared objects remain shared, which can be very important for mutable objects.
3. User-defined classes and their instances: Marshal does not support these at all, but pickle can save and restore class instances transparently. The class definition must be importable and live in the same module as when the object was stored.

4.4 BOTTLE

There are many frameworks in python which allows you to create webpage like bottle, flask, django. In this article you will learn how to create simple app bottle.Bottle is a fast, simple and lightweight WSGI micro web-framework for Python. It is distributed as a single file module and has no dependencies other than the Python Standard Library.

Routing: Requests to function-call mapping with support for clean and dynamic URLs.

Templates: Fast and pythonic built-in template engine and support for mako, jinja2 and cheetah templates.

Utilities: Convenient access to form data, file uploads, cookies, headers and other HTTP-related metadata.

Server: Built-in HTTP development server and support for paste, fapws3, bjoern, gae, cherrypy or any other WSGI capable HTTP server.

In order to create the app using bottle we have to install it first

Windows

pip install bottle

Ubuntu

pip3 install bottle

By default, if we pass a template name to the SimpleTemplate and it will look for a file by that name in a subdirectory views with the extension .tpl.

First we have to create the directory for our project Test_project Inside that

create a file and name it as app.py

Chapter 5

SYSTEM SPECIFICATION:

5.1 HARDWARE REQUIREMENTS:

- System : Intel Core i3 or higher
- Hard Disk : 50 GB available hard disk space
- Monitor : Generic Colour Monitor.
- Mouse : Optical Mouse.
- Ram : 4 GB.

5.2 SOFTWARE REQUIREMENTS:

- Operating system : Windows, Linux and MacOS.
- Coding Language : Python.
- Front-End : Html, CSS, JavaScript, Bootstrap
- Data Base : SQLite.

Chapter 6

SOFTWARE REQUIREMENTS SPECIFICATION

6.1 REQUIREMENT ANALYSIS

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well-ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

REQUIREMENT SPECIFICATION

Functional Requirements

- Graphical User interface with the User.

Software Requirements

For developing the application the following are the Software Requirements:

1. Python
2. EEL
3. Bottle
4. Pickle
5. OpenCV
6. PyCharm

Operating Systems supported

1. Windows 10
2. Linux
3. MacOS

Technologies and Languages used to Develop

1. Python
2. HTML CSS, Javascript, Bootstrap
3. SQLite

Debugger and Emulator

- Any Browser (Particularly Chrome)

Hardware Requirements

For developing the application the following are the Hardware Requirements:

- Processor: Pentium IV or higher
- RAM: 4 GB
- Space on Hard Disk: 16 GB available hard disk space (32-bit) or 20 GB (64-bit)

6.2 Functional Requirements

In encoding gathering, a strong need is used to demonstrate the steadiness of a thing structure or it's a section. The large portion of a purpose behind existing is take an appeared as a system of information sources, the lead, and yields. Utilitarian prerequisites implies the principal activity must performed. This clarifies about what the framework must do and clarifies about which sorts of action will premed. Useful necessities for the framework are separated into three principle classifications they are Recipient subtleties Administrator subtleties, cloud subtleties and there capacities

Non Functional Requirements

6.2.1 Performance requirements

- Performance requirements means is to define a how much time its take to response for accepting the system functionality.
- Our project should support what and which type of the requirements user wants. And also supports the end users requirements
- First the system will be check the login details within a few seconds. It will be in a user, cloud and admin details.

- Our product should give the security with the help of a device. The device which contains a half part of a password. Another part will be stored in their system.

.

6.2.3 Security Requirements

In our system should provide the security requirements and give the device authentication password. Avoid an error with the help of an exception handling mechanisms.

6.2.4 Software Quality Attributes

Availability:

Our applications will not hanging problems, it will open quickly and access the data fast. The use will sufficient with these problems.

Reliability

Our system should identify the invalid inputs it will never take it should be verified. If any error occurs the system will show the error messages. And our system will not crash. Our system behaves like a user-acceptable manner when operating system within the manner for the system is in interface

Usability:

Our system interacts with another user easily and shares the data easily.

Maintainability

This system will be used for long-time. The system should be maintain efficiently, the device password can be maintain safely. Portability

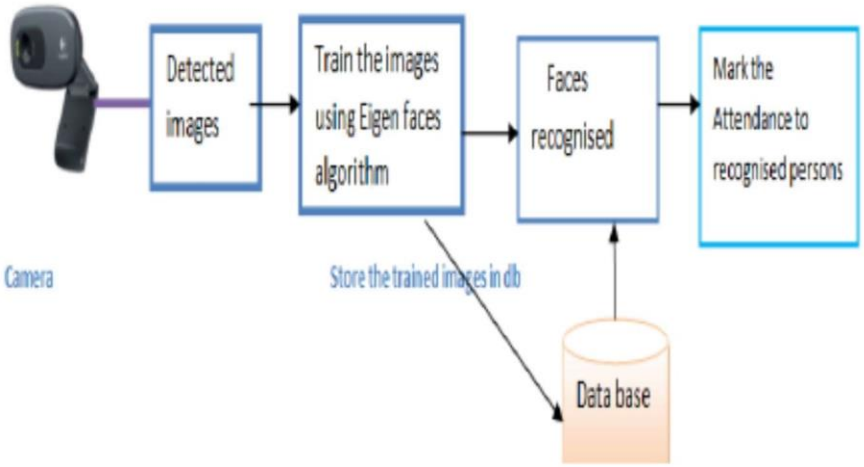
Our system can be run on any web browser on any platform with a little or no modification.

Chapter7

SYSTEM DESIGN

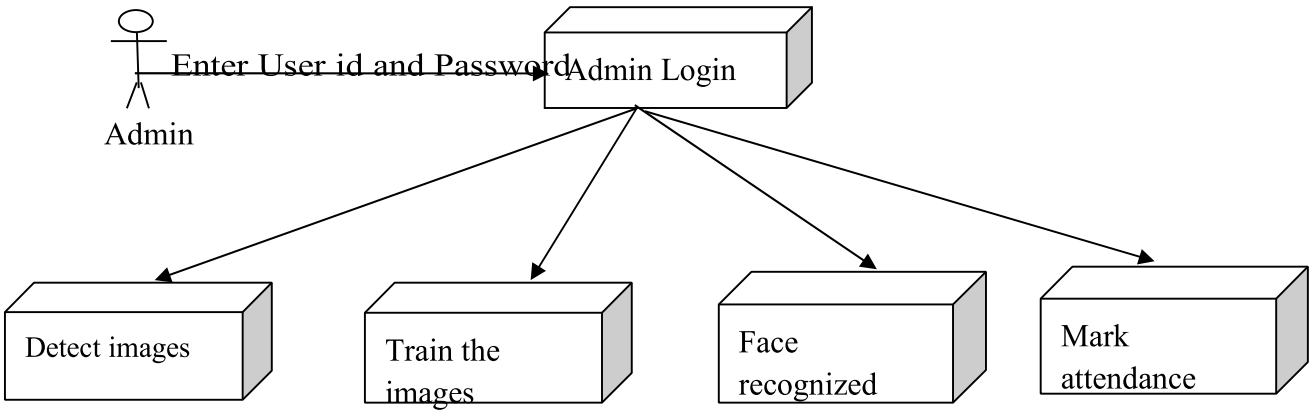
7.1 SYSTEM PRSPECTIVE

This investigation intends to plan a design demonstrate adjusted to the misrepresentation triangle factors, supplemented with the human factor and breaking down suspicious conduct to recognize conceivable instances of extortion. In this unique circumstance, a few investigations were found in the writing, which add to this point. The majority of the records address the issue of financial extortion and the diverse conditions encompassing it. In any case, distinguishing individuals who may be associated with false exercises is a deciding component. The attack into the social examination is cited to, whose creators present a programmed content mining process by email for the location of various kinds of examples in messages. What's more, it plays out the exemplary quantitative investigation of business exchanges that are as of now connected as a component of the extortion identification review.



Architectural Design

7.2 COMPONENT DIAGRAM



Chapter 8

DETAILED DESIGN

8.1 Use Case Diagram

An utilization case chart in the bound together displaying vernacular is a sort of lead plot delineated by and passed on utilizing a use case examination. The fundamental motivation driving the usage case diagram is to demonstrate what limits are performed for which on-screen character. Use case diagram consolidates set performng capable workers and use

8.5 Data Flow Diagram

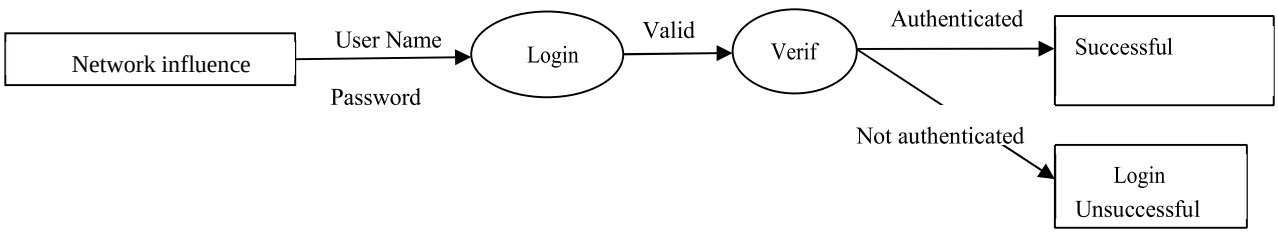
Data Flow Diagram is reliably utilized amidst the examination mastermind. The examination is delineated by the ace pictorially with the assistance of Data Flow Diagrams (DFDs).

A DFD demonstrates the Input/yield stream of information, Reports conveyed and Data stores that are utilized by the framework, It sees a structure as a technique that changes the responsibilities to preferred yield.

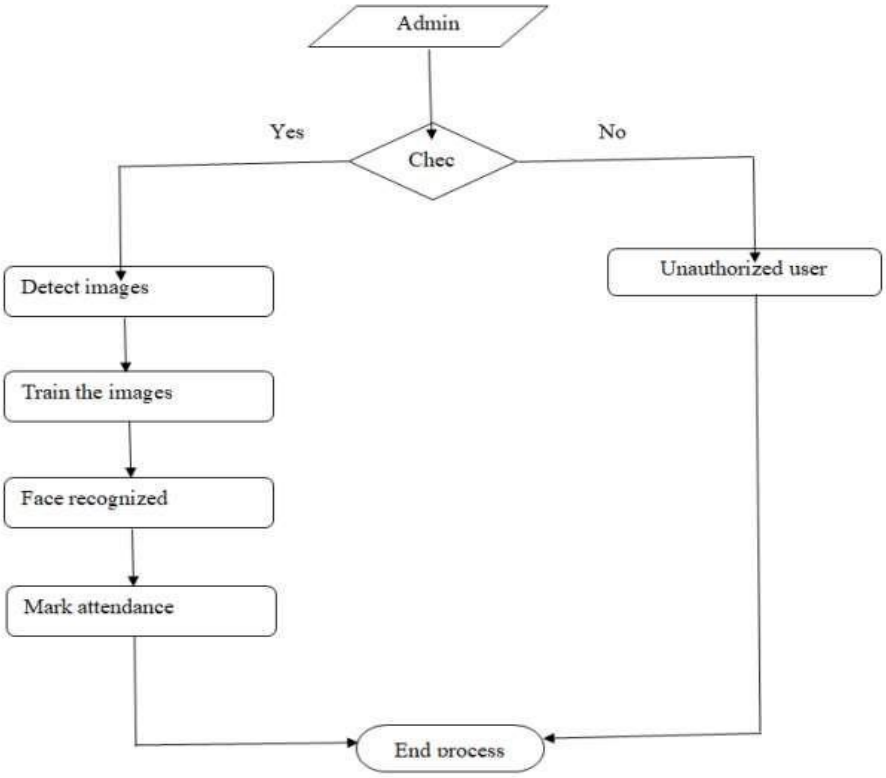
That construes a DFD demonstrates the distinction in information from promise to yield, through methods, might be portrayed keenly and energetically of the physical parts related with the framework. Data Flow Diagram is reliably utilized amidst the examination mastermind. The examination is delineated by the ace pictorially with the assistance of Data Flow Diagrams (DFDs).

Level -0 DFD

It shows structures critical methods, data streams, and the data stores at an unusual condition of reflection. It depends over dimension 0 and it sub-segregates the system into various disconnected social occasions.



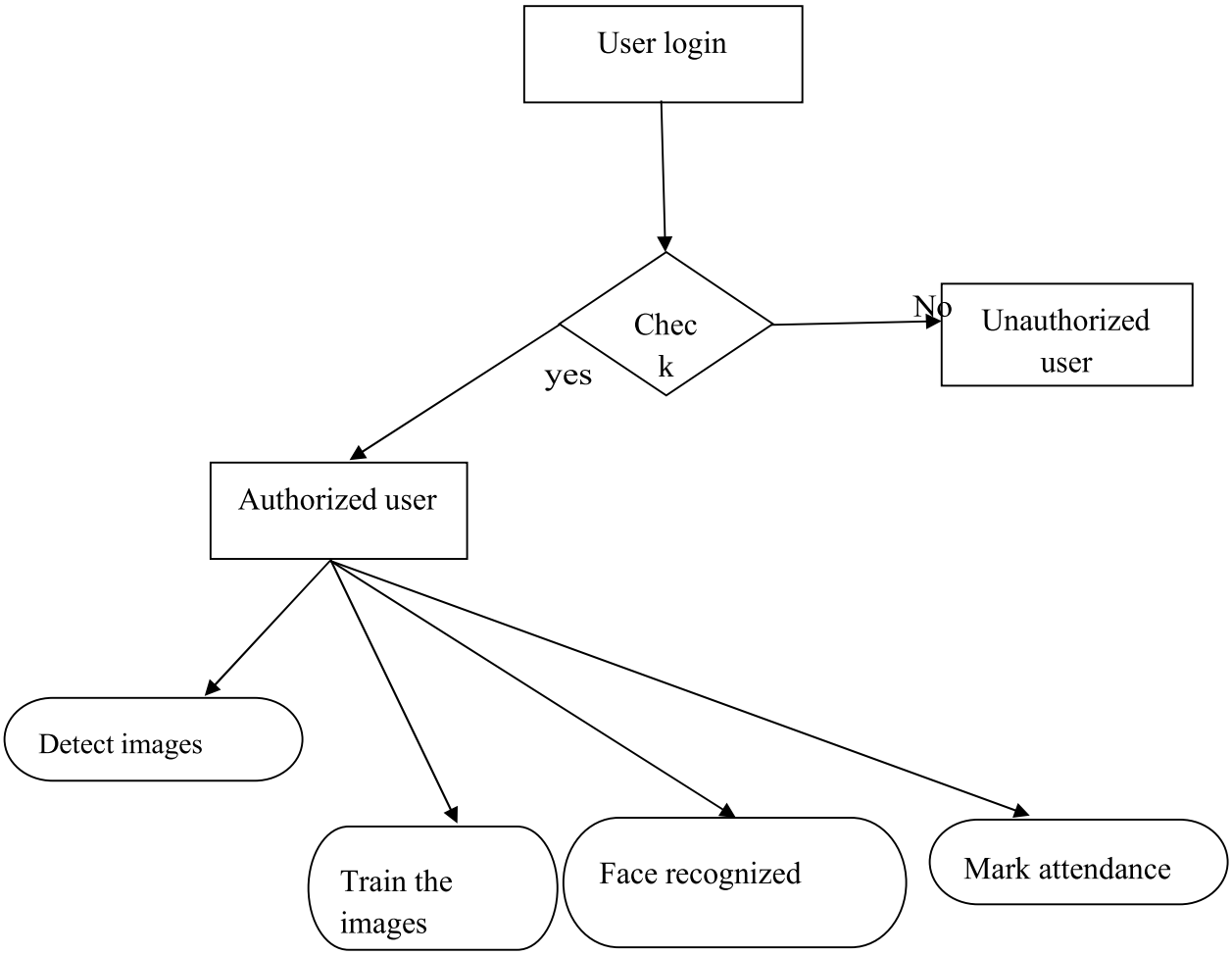
Level -1 DFD


Admin
Yes
Chec
No
Detect images
Unauthorized user
Train the images
Face recognized
Mark attendance
End process

8.6 Database Design

8.6.1 E-R Diagram

It is a challenge develop show and is arranged in light of a perspective on this present reality that is contained an aggregation of things or activities and associations among these. E-R graph all around used a top-down methodology for new structure.



Chapter 10

SYSTEM IMPLEMENTAION

10.1 Code Snippet

10.1.2 main py Code

```
import eel import cv2 import io import numpy as np import
base64 import os import time import face_recognition
import pickle import imutils import datetime from
multiprocessing.pool import ThreadPool import random
import shutil from database import * from camera import
VideoCamera from SceneChangeDetect import
sceneChangeDetect import photo import login import
encode_student_data
import warnings
warnings.filterwarnings('ignore')
eel.init('web')
"""
def recognizeFromPhoto(image):
            #cap = cv2.VideoCapture('http://192.168.0.2:4747/video')
        #retval, image = cap.read()     retval, buffer =
cv2.imencode('.jpg', image)    jpg_as_text =
base64.b64encode(buffer)      cap.release()
            return " " + jpg_as_text
"""
```

```
#------ Global Variable ----

camera_status = 1 capture_status =

False student_id = ''

"""     def     show_error(title,

msg):

   root = Tk()     root.withdraw()  # hide

main window

messagebox.showerror(title, msg)

root.destroy()

"""

def recogFace(data,encoding):

   return     face_recognition.compare_faces(data["encodings"],     encoding,     tolerance=0.5)     def
recogEncodings(rgb,boxes):

   return    face_recognition.face_encodings(rgb,    boxes)    def

recogLoc(rgb):

   return face_recognition.face_locations(rgb, model = "hog") def

gen1(url,student_class):     #change camera status for loading

eel.camera_status(3)     pool1 = ThreadPool(processes = 1)     pool2 =

ThreadPool(processes = 2)     pool3 = ThreadPool(processes = 3)     pool4 =

ThreadPool(processes = 4)     pool5 = ThreadPool(processes = 5)     conn =

create_connection()     cursor = conn.cursor()     sql = "SELECT student_id

FROM student_data WHERE class = ? "     val =[student_class]

cursor.execute(sql,val)     student_data = cursor.fetchall()

   #print(student_data)
```

```
   # Load the known face and encodings     #print("[INFO]
loading encodings ..")     data =
pickle.loads(open("encodings.pickle","rb").read())
Attendees_Names = {}     encodings = []     boxes = []
   frame = 0
   Scene = sceneChangeDetect()
video = cv2.VideoCapture(url)
time.sleep(1.0)     global
camera_status     camera_status = 1
#change the camera status
eel.camera_status(1)     while
camera_status == 1:
       frame            +=1
if(frame==100):
frame = 0
       #print(camera_status)
#img = camera.get_frame()
success, img = video.read()
       #if camera can't read frame(Camera error)
if success == False:            eel.camera_status(2)
break
       if(Scene.detectChange(img) == True):
          #Convert the BGR to RGB
          # a width of 750px (to speed up processing)            rgb
= cv2.cvtColor(img, cv2.COLOR_BGR2RGB)            rgb =
```

```
imutils.resize(img, width = 750)          r =
img.shape[1]/float(rgb.shape[1])
        #detect                    boxes
if(frame%2 == 0):
            boxes = pool1.apply_async(recogLoc,(rgb,)).get()          encodings =
pool3.apply_async(recogEncodings,(rgb,boxes,)).get()          names = []
            # loop over the facial encodings
for encoding in encodings :
            # attempt to match each face then initialise a dicationary
            #matches = face_recognition.compare_faces(data["encodings"], encoding,tolerance=0.5)
matches = pool2.apply_async(recogFace,(data,encoding,)).get()          name = "Unkown"
            # check to see if we have found a match
if True in matches:
                #find the indexes of all matched faces then initialize a
                # dicationary to count the total number of times each face matched
matchedIds = [i for (i,b) in enumerate(matches) if b]          counts ={}
                #loop over the recognized faces
for i in matchedIds:
                    name          =          data["names"][i]
counts[name] = counts.get(name,0)+1
                #determine the recognized faces with largest number
                # of votes (note: in the event of an unlikely tie Python will select first entry in the dictionary)
name = max(counts , key = counts.get)
```

```
            if(name not in Attendees_Names):
Attendees_Names[name]= 1                    for y in
student_data:
                    if name in y:
                        x = datetime.datetime.now()                    date = str(x.day)+"-
"+str(x.month)+"-"+str(x.year)
pool4.apply_async(submit_live_attendance,(name,student_class,date,))
                        #submit_live_attendance(name,student_class,date)
eel.updateAttendance(name)()            names.append(name)
    # loop over recognized faces
    """
    for ((top,right,bottom,left),name)in zip(boxes, names):
       top = int(top*r)
right = int(right*r)
bottom = int(bottom*r)
left = int(left *r)
       # draw the predicted face name on the image          cv2.rectangle(img,
(left, top), (right, bottom),
          (0, 255, 0), 2)          y = top - 15 if top - 15 > 15 else top + 15
cv2.putText(img, name, (left, y), cv2.FONT_HERSHEY_SIMPLEX,
          0.75,  (0,  255,  0),  2)
"""
    ret, jpeg = cv2.imencode('.jpg', img)
    img     =     jpeg.tobytes()
yield img
```

```
   #camera is stopped by user      if
success == True:
        eel.camera_status(0)
        #cv2.imshow("Frame", img)
        #key = cv2.waitKey(1) & 0xFF
        # if the `q` key was pressed, break from the loop
        #if key == ord("q"):
           #break
@eel.expose                                              def
start_video_py(cam_type,student_class,url = ''):
   #x = VideoCamera()     switch={
        '1' : 0,
        '2' : 1,
        '3' : url,
   }
   y = gen1(switch[cam_type],student_class)       for
each in y:
        # Convert bytes to base64 encoded str, as we can only pass json to frontend
blob = base64.b64encode(each)        blob = blob.decode("utf-8")
eel.updateImageSrc(blob)()
        # time.sleep(0.1)
@eel.expose def stop_video_py():     global camera_status
camera_status = 0 @eel.expose def
photoUpload(b64_string,student_class):     encoded_data =
b64_string.split(',')[1]     decoded_data =
```

```
base64.b64decode(encoded_data)     nparr =
np.fromstring(decoded_data, np.uint8)
   #nparr = np.fromstring(encoded_data.decode('base64'), np.uint8)     img =
cv2.imdecode(nparr, cv2.IMREAD_COLOR)
photo.recognizeFromPhoto(img,student_class)
   ##print(photo_string)
   #return photo_string
@eel.expose def
capture_photo_py(url):     #x
= VideoCamera()     y =
gen(url)     for each in y:
       # Converted to base64 encoded str, as we can only pass json to frontend
blob = base64.b64encode(each)         blob = blob.decode("utf-8")
eel.updateStudentImageSrc(blob)()
       #   time.sleep(0.1)   def
gen(url):
   video = cv2.VideoCapture(url)
time.sleep(2.0)     global
camera_status     global
capture_status     camera_status = 1
#change the camera statu     while
camera_status == 1:         #img =
camera.get_frame()         success,
img = video.read()         #if camera
```

```
can't read frame        if success ==
False:            #print("cam nt cnt")
break        if capture_status ==
True:
          save_path   =   'dataset/'+student_id                              filename   =
save_path+"/photo"+str(random.randint(0,  999))+".jpg"                       if  not
os.path.exists(save_path):
              os.makedirs(save_path)
cv2.imwrite(filename, img)
send_capture_photo(img)
capture_status = False        ret, jpeg =
cv2.imencode('.jpg', img)        img =
jpeg.tobytes()        yield img
#add new user data
#@eel.expose
#def add_new_student(new_student_id):
   ##print(new_student_id)
   #encode_student_data.encode_student_data(new_studentId)              def
submit_live_attendance(stu_id, student_class,date):
   attendance_class={
      "xii" : "INSERT INTO xii(student_id,attendance_date) VALUES(?, ?);",
      "xi"  : "INSERT INTO xi(student_id,attendance_date) VALUES(?, ?);",
   }
   #adding data to database     conn =
create_connection()     cursor =
```

```
conn.cursor()     sql =
attendance_class[student_class]     val =
[stu_id, date]     cursor.execute(sql, val)
conn.commit()     conn.close()
@eel.expose def
save_photo(studentId):     global
student_id     global
capture_status     student_id =
studentId     capture_status =
True def
send_capture_photo(img):
   ret, jpeg = cv2.imencode('.jpg', img)
   img = jpeg.tobytes()     blob =
base64.b64encode(img)     blob =
blob.decode("utf-8")
eel.showCapturePhoto(blob)
#adding new student data
@eel.expose  def  submit_student_data(stu_id,  fullname,  student_class,
session):
   try:
      encode_student_data.encode_student_data(stu_id)
      #adding data to database        conn = create_connection()        cursor = conn.cursor()        sql =
"INSERT INTO student_data(student_id,fullname,class,session) VALUES(?, ?, ?, ?);"        val = [stu_id,
fullname, student_class, session]        cursor.execute(sql, val)        conn.commit()
eel.student_data_saved()        conn.close()     except:
```

```
        #delete face data from file         delete_student_data_file(student_id)
eel.failed_data_submit()
@eel.expose                                 def
fetch_class_data(search_class):
    conn = create_connection()
    cursor = conn.cursor()      val = [search_class]      sql =
"SELECT * FROM student_data WHERE class = ?"      result =
cursor.execute(sql,val)      for x in result:
eel.setTableData(x[0],x[1],x[2],x[3])      conn.close() def
delete_student_data_file(student_id):
    #delete face data from file      #load the
face data      with open('encodings.pickle',
'rb') as f:
            face_data = pickle.load(f)
index = []      encodings =
face_data['encodings']      names     =
face_data['names']      #count face data
length      for i,item in enumerate(names):
        if student_id in item:
index.append(i)      #delete
id      for i in index:
        names.remove(student_id)         #delete
encoding
    for i in index:
        del encodings[index[0]]
```

```
    #saved modified face data
face_data['names'] = names
face_data['encodings'] = encodings     f =
open("encodings.pickle","wb")
    f.write(pickle.dumps(face_data))
    f.close()
@eel.expose                         def
deleteStudent(student_id):
    try:
        ##print("connect to database")
        ##print(student_id)
        #delete student image folder
        try:
            path    =    'dataset/'+student_id
shutil.rmtree(path)                    except
Exception as e:
            print(e)
        #delete student data from database         conn =
create_connection()         cursor = conn.cursor()         val =
[student_id]         sql = "DELETE FROM student_data where
student_id = ?"         cursor.execute(sql,val)         conn.commit()
        conn.close()
        ##print("delete success database")
```

```
    #delete face data from file
delete_student_data_file(student_id)
eel.deleteStatus(student_id)                except
Exception as e:
    print(e)
eel.deleteStatus("")
@eel.expose        def        fetchAttendance(attendanceClass,
attendanceDate):
  student_class={
    'xi'  : "SELECT DISTINCT(d.student_id),d.fullname,d.class,ac.attendance_date FROM xi ac,student_data d WHERE ac.student_id=d.student_id AND attendance_date = ?;",
    'xii'  :  "SELECT  DISTINCT(d.student_id),d.fullname,d.class,ac.attendance_date  FROM  xii ac,student_data d WHERE ac.student_id=d.student_id AND attendance_date = ?;",
  }
  conn = create_connection()   cursor =
conn.cursor()   val = [attendanceDate]
sql = student_class[attendanceClass]
cursor.execute(sql,val)   result =
cursor.fetchall()   print(len(result))   if
len(result)>0:      for x in result:
eel.attendanceTable(x[0],x[1],x[2],x[3])
else:
    eel.attendanceTable("no result found","","","")   conn.close()
```

```
@eel.expose def
fetch_graph_data(graphClass):
student_class = {
      'xi'  : "SELECT DISTINCT(attendance_date) FROM xi ORDER BY attendance_date ASC LIMIT
06 ",
      'xii' : "SELECT DISTINCT(attendance_date) FROM xii ORDER BY attendance_date ASC LIMIT 06 ",
   }
   attendance_class = {
      'xi'  : "SELECT COUNT(DISTINCT(student_id)) FROM xi WHERE attendance_date = ? ;",
      'xii' : "SELECT COUNT(DISTINCT(student_id)) FROM xii WHERE attendance_date = ? ;",
   }

   conn = create_connection()     cursor
= conn.cursor()     sql = student_class[
graphClass ]     result =
cursor.execute(sql)     date_arr = []
data_arr = []     for x in result:
      date_arr.append(x[0])
   #print(date_arr)     sql =
attendance_class[ graphClass ]     for x in
date_arr:         val = [x]         result =
cursor.execute(sql,val)         for x in result:
data_arr.append(x[0])     #print(data_arr)
cursor.close()
```

```
eel.updateGraph(date_arr,data_arr)

@eel.expose def get_user_details():

    return login.session['user_name']

eel.start('login.html',size =(1307,713))
```

Chapter 11

SOFTWARE TESTING

11.1 Introduction

SYSTEM TEST

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

11.2 TYPES OF TESTS

Unit test

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration test

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.

Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.

- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

11.3 Test Cases

11.3.1 ADMIN

Serial number of test case	1
Name of the test	Unit test
Item/ feature being tested	ADMIN Login Page
Expected output	Username and Password should be verified with the database, if it is valid then navigate to respective pages else shows Invalid message.
Actual output by referring to snapshot number	Username and Password is verified with the database, if it is valid then navigate to respective pages else it successfully shows Invalid Username and Password.
Remarks	Tested Successfully

11.3.2 Upload student Details

Serial number of test case	2
Name of the test	Integration test
Item/ feature being tested	Add file Details
Expected output	Data owner Login with their username and password. After they can add the particular file details. And upload file
Actual output by referring to snapshot number	All the fields of add file details are filled

11.3.3 User

Serial number of test case	3
Name of the test	Unit test
Item/ feature being tested	User Login
Expected output	Username and Password should be verified with the database, if it is valid then navigate to respective pages else shows Invalid message.

Actual output by referring to snapshot number	Username and Password is verified with the database, if it is valid then navigate to respective pages else it successfully shows Invalid Username and Password
Remarks	Tested Successfully

11.3.4 User Registration

Serial number of test case	4
Name of the test	Integration test
Item/ feature being tested	New User registration
Expected output	After filling all the fields if user provides all correct details then it will register the new user and gets the Username and Password.
Actual output by referring to snapshot number	New user got the Username and Password and registered successfully
Remarks	Tested Successfully

11.3.5 Search student

Serial number of test case	5
Name of the test	Integration test
Item/ feature being tested	Query
Expected output	User login with their username and password. After that user should insert a query.
Actual output by referring to snapshot number	When the user searches query it will display the message query successfully registered
Remarks	Tested Successfully

CONCLUSION

Student's attendance is used to be usually achieved by classical way-this means record papers or more novel approaches by hardware tools such as radiofrequency identification (RFID), near field communication (NFC), biometric identification or combination of just presented. But in our proposed system does not require to carry any hardware device nor to perform some kind of direct biometric identification. The proposed system an easy way for marking attendance where student is identified by camera, where the faces are matched to the one stored in the database after comparing the trained images. In this way students are automatically and indirectly monitored during classes and lectures, which is a better way for attendance system. Another area of future work is improving our neural network classifier. As mentioned in the previously, it is possible to construct the network to take its input directly from the image data rather from the vector that results from an images projection into face-space. Perhaps learning the face projection function could increase the accuracy of the neural network classifier. Additionally, more experiments are needed to see if there are other ways to tweak the network configuration to produce better results

References

C. Lin, 2005, “Face Detection By Color And Multilayer Feedforward Neural Network”, Proc. 2005

IEEE International Conference on Information Acquisition, pp.518-523, Hong Kong and Macau, China

OpenCV Documentation https://docs.opencv.org/master/d9/df8/tutorial_root.html

Chen, S.K; Chang, Y.H (2014). 2014 International Conference on Artificial Intelligence and Software Engineering (AISE2014). DEStech Publications, Inc. p. 21. ISBN 9781605951508.

Bramer, Max (2006). Artificial Intelligence in Theory and Practice: IFIP 19th World Computer Congress, TC 12: IFIP AI 2006 Stream, August 21–24, 2006, Santiago, Chile. Berlin: Springer Science+Business Media. p. 395. ISBN 9780387346540.

Office, U. S. Government Accountability. "Facial Recognition Technology: Federal Law Enforcement Agencies Should Have Better Awareness of Systems Used By Employees". www.gao.gov. Retrieved September 5, 2021.

Staff, By GCN; Jun 10, 2020. "IBM bows out of facial recognition market -". GCN. Retrieved October 7, 2021.

Nilsson, Nils J. (2009). The Quest for Artificial Intelligence. Cambridge University Press. ISBN 9781139642828.

de Leeuw, Karl; Bergstra, Jan (2007). The History of Information Security: A Comprehensive Handbook. Elsevier. p. 266