“INTELLIGENT ATTENDANCE MANAGEMENT SYSTEM”

***A Major Project submitted to***

***Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal***

***In partial fulfillment of the requirements for the award of***

***Degree of***

**Bachelor of Engineering**

***By***

**SHUBHAM GAUR**

**NOURIN KHAN**

**PALLAVI CHOURE**

***Under the guidance of***

**Prof. Shreyas Pagare**

****

**Department of Computer Science Engineering**

**DECLARATION**

We certify that the work contained in this report is original and has been done by under the guidance of my supervisor(s).

1. The work has not been submitted to any other Institute for any degree or diploma.
2. I have followed the guidelines provided by the Institute in preparing the report.
3. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
4. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Signature of the Student (s)

**CERTIFICATE**

Certified that the project report entitled, “INTELLIGENT ATTENDANCE MANAGEMENT SYSTEM” is a bonafide work done under my guidance by NOURIN KHAN, PALLAVI CHOURE, and SHUBHAM GAUR in partial fulfillment of the requirements for the award of degree of Bachelor of Engineering in Computer Science Engineering.

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Prof. Shreyas Pagare )

Guide

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Mr. Surendra kumar Shukla Sir) (Dr. C.N.S Murthy Sir)

Head of the Department Dean

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

E mail: Phone No: Fax No:

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| S. No | CONTENTS | Page No. |
| 1. | Title Page | i |
| 2. | Declaration | ii |
| 3. | Certificate by the Supervisor | iii |
| 4. | Acknowledgement | iv |
| 5. | Abstract | v |
| 6. | Project Overview | vii |
| 7. | Technology Used | xi |
| 8. | Data Tables | xiii |
| 9. | Data Flow Diagrams | xvi |
| 10. | Use case Diagrams | xviii |
| 11. | Activity Diagrams | xix |
| 12. | Sequence Diagrams | xxv |
| 13. | ER Diagram | xxv |
| 14. | Testing | xxvi |
| 15. | Algorithm | xxxi |
| 16. | Code | xxxiii |
| 17. | User Manual | xxxxi |
| 18. | References | xxxxiv |

**ACKNOWLEDGEMENT**

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. We thank our dean **Dr. C.N.S Murthy Sir** who have helped a lot in our successful completion of the project. We are grateful to our project guide **Prof. Shreyas Pagare Sir** for the guidance, inspiration and constructive suggestions that helpful us in the preparation of this project.

**ABSTRACT**

Being one of the most successful applications of the image processing, face recognition has a vital role in technical field especially in the field of security purpose. Human face recognition is an important field for verification purpose especially in the case of student’s attendance. This paper is aimed at implementing a digitized system for attendance recording. Current attendance marking methods are monotonous & time consuming. Manually recorded attendance can be easily manipulated. Hence the paper is proposed to tackle all these issues. Face Detection and Recognition is an important area in the field of substantiation. Maintenance of records of students along with monitoring of class attendance is an area of administration that requires significant amount of time and efforts for management. Intelligent Attendance Management System performs the daily activities of attendance analysis, for which face recognition is an important aspect. The prevalent techniques and methodologies for detecting and recognizing face like PCA-LDA, etc. fail to overcome issues such as scaling, pose, illumination, variations, rotation, and occlusions. The proposed system provides features such as detection of faces, extraction of the features, detection of extracted features, analysis of students' attendance and monthly attendance report generation. The proposed system integrates techniques such as image contrasts, integral images features and cascading classifier for feature detection. Faces are recognized using advanced Eigen face algorithm using the database that contains images of students and is used to recognize student using the captured image. Better accuracy is attained in results and the system takes into account the change that occurs in the face over the period of time.

**PROJECT OVERVIEW**

**PROJECT OBJECTIVE:**

The key objective of intelligent attendance system is to present an automated system for human face recognition in a real time background for an organization to mark the attendance of their employees or student. So automated attendance using real time face recognition is a real world solution which comes with day to day activities of handling employees or student. The proposed system will provide an optimized platform that will help users to easily mark attendance, view user data and their related course records. The proposed system will provide secure attendance management via biometric techniques. Face recognition will be the core feature of the proposed system. The proposed system will allow users to print the desired attendance records. Users will be able to have a secured system with authenticated attendance management.

**INTRODUCTION:**

Every institute has its own method in the regard to take attendance. Some are taking attendance manually using the old paper or file based approach and some have adopted methods of automatic attendance using some biometric techniques. Traditionally, student’s attendances are taken manually by using attendance sheet given by the faculty members in class, which is a time consuming event. An automatic attendance management system using biometrics would provide the needed solution. The results showed improved performance over manual attendance management system. Biometric based technologies include identification based on physiological characteristics (such as face, fingerprints, finger geometry, hand geometry, hand veins, palm, iris, retina, ear and voice) and behavioral traits (such as signature and keystroke dynamics). Face recognition appears to offer several advantages over other biometric methods. Facial recognition or face recognition as it is often referred to as, analyses characteristics of a person's face image input through a camera. It measures overall facial structure, distances between eyes, nose, mouth, and jaw edges. These measurements are retained in a database and used as a comparison when a user stands before the camera. One of the strongest positive aspects of facial recognition is that it is non-intrusive. Verification or identification can be accomplished from two feet away or more, without requiring the user to wait for long periods of time or do anything more than look at the camera. Traditionally student’s attendance is taken manually by using attendance sheet, given by the faculty member in class. The Current attendance marking methods are monotonous & time consuming. Manually recorded attendance can be easily manipulated. Moreover, it is very difficult to verify one by one student in a large classroom environment with distributed branches whether the authenticated students are actually responding or not. Hence the paper is proposed to tackle all these issues .The proposed system consists of a high resolution digital camera to monitor the classroom or office room. It is embedded on a micro-controller based motor system which enables it to rotate in left & right directions. The data or images obtained by the camera are sent to a computer programmed system for further analysis. The obtained images are then compared with a set of reference images of each of the employees or students & mark the corresponding attendance. The system also provides for continuous monitoring of the classroom by an operator if needed. The camera module can be a wireless or wired system.

**Problems in Existing System**

Organizations of all sizes use time based attendance systems to record when employees start and stop work and the department where work is performed. In school, colleges and universities attendance system is used to keep record of a student presence and absence. It is also used to keep record of a teacher and all other employees working in an organization. It is usually done on registers. There is a separate register for each class, department etc. Organizations keep record for the sake of: leave management, work-hour management. In short, attendance is something which every organization Introduction Face Recognition Based Attendance System 3 keeps record of and which is essential part of every organization whether these are schools, hospitals, government or private firms or media related organization.

**The drawbacks in existing system are:**

∙ These attendance systems are manual

∙ There is always a chance of forgery (one person signing the presence of the other one)

∙ Since these are manual so there is great risk of error

∙ More man-power is required (some person to take attendance)

∙ Calculations related to attendance are done manually (total classes attended in month) which is prone to error

∙ It is difficult to maintain database or register in manual systems

∙ It is more costly (price of register, pen and the salary of person taking attendance)

∙ It is difficult to search for particular data from this system (especially if that data we are asking for is of every long ago)

**Features of Face Recognition based Attendance System**

∙ Automatically identify or verify a person from a digital image or a video frame from a video source.

∙ Reduces cost and time

∙ This system is digitized

∙ It is more secure than manual one (face recognition)

∙ It uses biometrics which is even more secure method than simple security systems (password oriented)

∙ It uses less man power i.e. it does not require a person to take attendance; either every person can come and mark his own attendance or it can automatically take attendance (if we take into account video streaming; projects future work)

∙ It can store more database

∙ Search of any particular data is easy

∙ It gives calculations more easily and in less time

**Proposed System Components**

Following are the main components of the proposed system

∙ Face Detection

∙ Face Recognition

∙ Attendance Records

∙ Printing Records

∙ Login

∙ Attendance management system

It will allow uploading, updating and deletion of the contents of the system. Attendance management will handle:

∙ Automated Attendance marking

∙ Manual Attendance marking

∙ Printing attendance record

∙ Attendance details of users.

**Proposed System Outcome**:

The proposed system will only allow authenticated user to login to the system and/or make changes to it. The proposed system will allow user to mark attendance of the students via face recognition technique. The proposed system will detect faces from live stream via webcam and then recognize the faces. After recognition it will mark the Introduction Face Recognition Based Attendance System attendance of the recognized student and update the attendance record. The user will be able to print these record details afterward.

**TECHNOLOGY USED**

**PYTHON:**

Python is an interpreted, object-oriented programming language similar to PERL that has gained popularity because of its clear syntax and readability. Python is said to be relatively easy to learn and portable, meaning its statements can be interpreted in a number of operating systems, including UNIX-based systems, Mac OS, MS-DOS, OS/2, and various versions of Microsoft Windows 98.

Python was created by Guido van Rossum, a former resident of the Netherlands, whose favourite comedy group at the time was Monty Python's Flying Circus. The source code is freely available and open for modification and reuse. Python has a significant number of users. A notable feature of Python is its indenting of source statements to make the code easier to read. Python offers dynamic data type, ready-made class, and interfaces to many system calls and libraries. It can be extended, using the C or C++ language.

Python can be used as the script in Microsoft's Active Server Page (ASP) technology. The scoreboard system for the Melbourne (Australia) Cricket Ground is written in Python. Z Object Publishing Environment, a popular Web application server, is also written in the Python language.

**SQL SERVER:**

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network. SQL Server has capability to store and manage large amount of data.

**Why we use SQL Server?**

We are using SQL Server due to following reasons

**Security** SQL Server provides security models including integration with Windows Authentication to provide robust security platform for managing information. Security can be managed across all database objects including tables, views, and stored procedures and at a very granular level, to provide complete information security.

**Scalability** SQL Server is a highly scalable database platform capable of managing terabytes of data while delivery high performance. Implementation Face Recognition Based Attendance System

**Integration** SQL Server provides a robust set of integration tools call SQL Server Integration Services (SSIS) which can be used consulate data from multiple heterogeneous data sources for holistic data view and comprehensive data analysis. SSIS can also be used to migrate data from other database formats (Oracle, DBase, Paradox, legacy formats, Microsoft Access) to SQL Server.

**Reliability** with SQL Server the clients do not talk directly with the tables but with an intelligent data manager on the server. This in turn reads and writes data from and to the tables. If a client machine crashes, or the network hiccups, this will not affect the underlying tables; instead the data manager realizes that the transaction has not been completed and does not commit the partially transmitted data to the database. The database therefore continues to run without problem.

**QT DESIGNER:**

Qt Designer is Qt's tool for designing and building graphical user interfaces (GUIs) from Qt components. You can compose and customize your widgets or dialogs in a what-you-see-is-what-you-get (WYSIWYG) manner, and test them using different styles and resolutions.

Widgets and forms created with Qt Designer integrated seamlessly with programmed code, using Qt's signals and slots mechanism, that lets you easily assign behavior to graphical elements. All properties set in Qt Designer can be changed dynamically within the code. Furthermore, features like widget promotion and custom plugins allow you to use your own components with Qt Designer.

**DATA TABLES**

**STUDENT TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO. | Field name | Data type | Description |
| 1. | StudentName | varchar | Store student’s name |
| **2.** | enrollment | varchar | Store student’s enrollment number |

**TIMETABLE TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO. | Field name | Data type | Description |
| 1. | starttime | time | Store starting time of lecture |
| 2. | endtime | time | Store ending time of lecture |
| 3. | monday | varchar | Stores subject name of respective day and time |
| 4. | Tuesday | varchar | Stores subject name of respective day and time |
| 5. | wednesday | varchar | Stores subject name of respective day and time |
| 6. | thursday | varchar | Stores subject name of respective day and time |
| 7. | friday | varchar | Stores subject name of respective day and time |

**ATTENDANCE TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO. | Field name | Data type | Description |
| 1. | enrollment | varchar | Stores enrollment of present student |
| 2. | date | date | Stores attendance recording date |
| 3. | time | time | Stores attendance recording time |
| 4. | day | varchar | Stores attendance recording day |

**MASTER TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO. | Field name | Data type | Description |
| 1. | studentname | varchar | Stores student’s name |
| 2. | enrollment | varchar | Stores student’s enrollment |
| 3. | day | varchar | Stores attendance recording day |
| 4. | date | date | Stores attendance recording date |
| 5. | time | time | Stores attendance recording time |
| 6. | subject | varchar | Stores attendance recording subject |
| 7. | facultyName | varchar | Stores faculty name of respective subject |

**FACULTY TABLE**

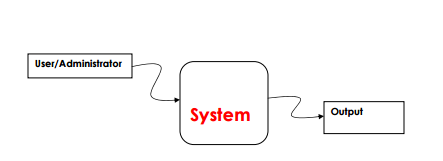
|  |  |  |  |
| --- | --- | --- | --- |
| S.NO. | Field name | Data type | Description |
| 1. | Faculty Name | varchar | Store name of the faculty |
| 2. | subject | varchar | Store the name of the subject taught by respective faculty |

**AUTHENTICATION TABLE**

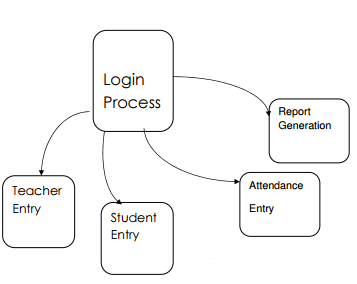
|  |  |  |  |
| --- | --- | --- | --- |
| S.NO. | Field name | Data type | Description |
| 1. | Login Id | varchar | Login Id or username used to login for admin |
| 2. | Password | varchar | Password used for authentication |

**DATAFLOW DIAGRAMS**

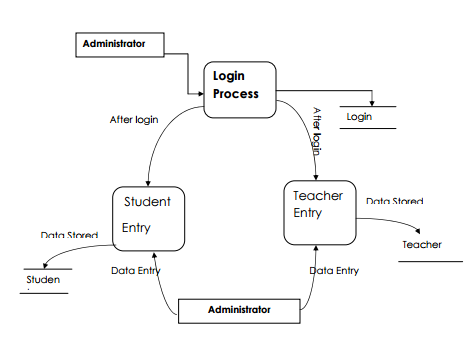
**0**-Level:

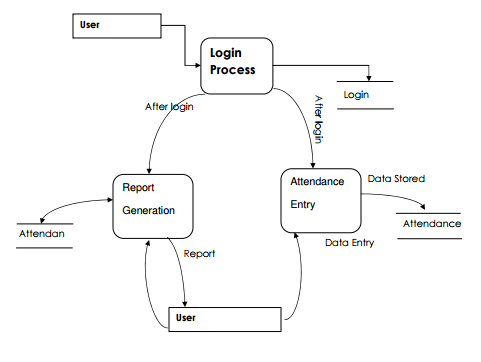
****

1-Level :



2- Level - 2.1 and 2.2

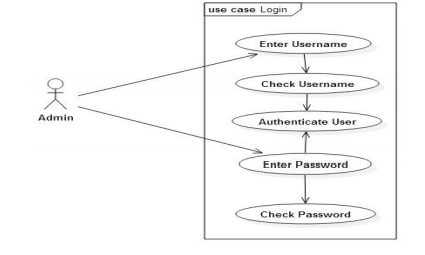




**USECASE DIAGRAMS**

Use case illustrates a unit of functionality provided by the system. The main purpose of the use-case diagram is to help development teams visualize the functional requirements of a system, including the relationship of "actors" (human beings who will interact with the system) to essential processes, as well as the relationships among different use cases.

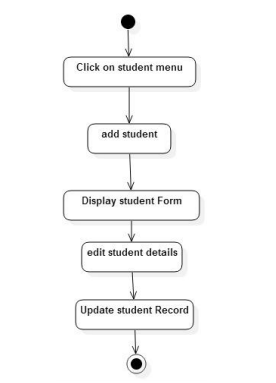
**5.1 Use Case Diagram for Login**

****

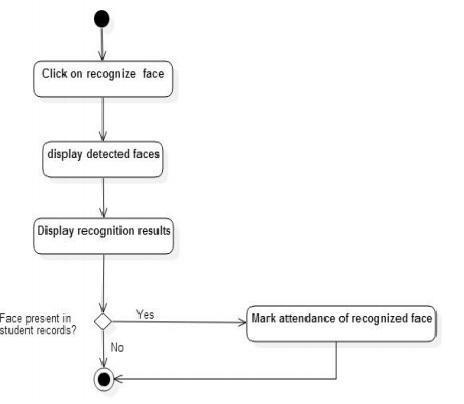
**ACTIVITY DIAGRAMS**

Activity diagrams show the procedural flow of control between two or more class objects while processing an activity. Activity diagrams can be used to model higher-level business process at the business unit level, or to model low-level internal class actions. In my experience, activity diagrams are best used to model higher-level processes, such as how the company is currently doing business, or how it would like to do business. This is because activity diagrams are "less technical" in appearance, compared to sequence diagrams, and business-minded people tend to understand them more quickly.

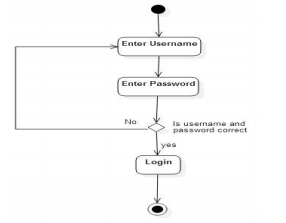
* 1. **Activity Diagram for Add Student**



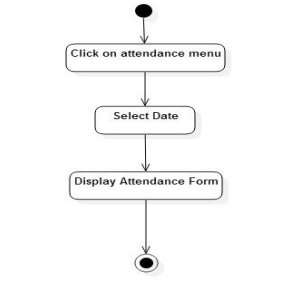
* 1. **Activity Diagram for Automated Attendance**



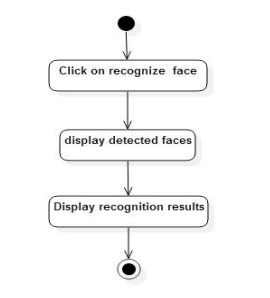
* 1. **Activity Diagram for Login**



* 1. **Activity Diagram for view Attendance**

****

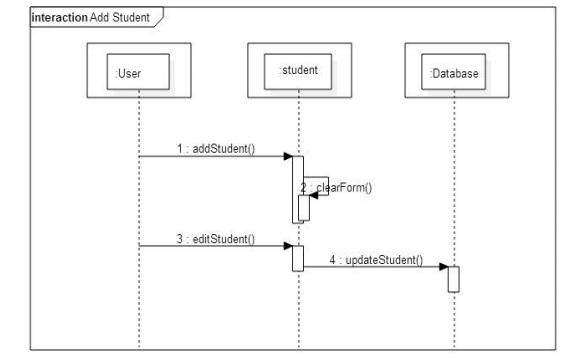
* 1. **Activity Diagram for Recognize Image**

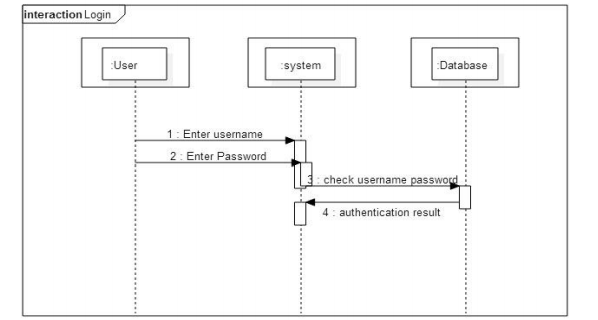
****

**Sequence Diagram**

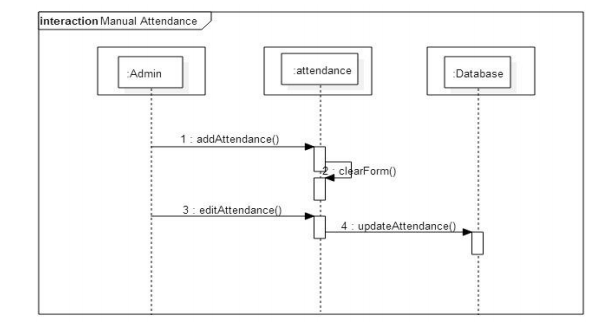
Sequence diagrams show a detailed flow for a specific use case or even just part of a specific use case. They are almost self-explanatory; they show the calls between the different objects in their sequence and can show, at a detailed level, different calls to different objects. A sequence diagram has two dimensions: The vertical dimension shows the sequence of messages/calls in the time order that they occur; the horizontal dimension shows the object instances to which the messages are sent.

**7.1 Sequence Diagram for Add Student**

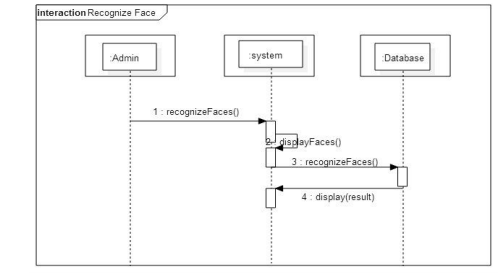


**7.2 Sequence Diagram for Login **

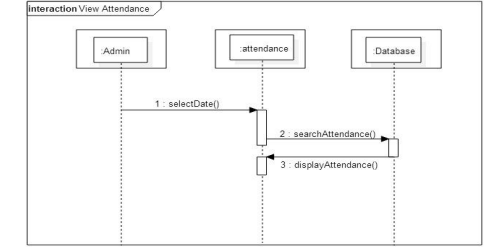
**7.3 Sequence Diagram for Attendance updation**

****

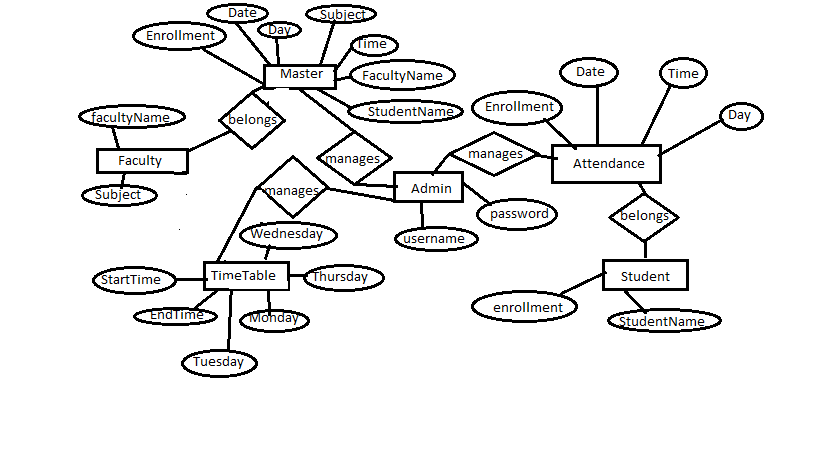
**7.4 Sequence Diagram for Recognize Face**

****

**7.5 Sequence Diagram for View Attendance**



**ER DIAGRAM:**



**TESTING**

**Introduction**

In general, testing is finding out how well something works. In computer hardware and software development, testing is used at key checkpoints in the overall process to determine whether objectives are being met. Software testing is the process of evaluation a software item to detect differences between given input and expected output.

**Types of testing**

Different types of testing include

1. **Unit Testing**: Unit testing is the testing of an individual unit or group of related units.
2. **Integration Testing**: Integration testing is testing in which a group of components are combined to produce output.
3. **Functional Testing**: Functional testing is the testing to ensure that the specified functionality required in the system requirements works.
4. **System Testing:**  System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works.
5. **Stress Testing**: Stress testing is the testing to evaluate how system behaves under unfavorable conditions.
6. **Performance Testing:** Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements.
7. **Usability Testing**: Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly?
8. **Acceptance Testing:**  Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected.
9. **Regression Testing** : Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results
10. **Beta Testing** : Beta testing is the testing which is done by end users, a team outside development, or publicly releasing full pre-version of the product which is known as beta version

**LOGIN**

|  |  |
| --- | --- |
| Test Id: | TC1 |
| Tester: | Admin |
| Purpose: | Login to system |
| Pre-requisites: | Must fill login form |
| Test Data: | Username, Password |
| Steps: | 1. Start System 2. Login form will be displayed 3. Enter password. 4. 6- Click login. 5. If the username and password is correct main form will appear. 6. If the username and password is not correct message box will appear. |
| Status: | Pass |

**Add Student Record**

|  |  |
| --- | --- |
| Test Id: | TC2 |
| Tester: | Admin |
| Purpose: | To verify that admin can add student record |
| Pre-requisites: | User must login |
| Test Data: | Name,enrollment,picture,year,branch |
| Steps: | 1. Login 2. Main page will appear 3. Click on student menu 4. Drop down menu will appear 5. Click on add student button 6. Student form will appear 7. Fill all the textboxes with student information and add photo of student. 8. Click save button 9. Students record will be added to existing record |
| Status: | Pass |

**DETECT FACE**

|  |  |
| --- | --- |
| Test Id: | TC3 |
| Tester: | Admin |
| Purpose: | Detect Face |
| Pre-requisites: | User must login |
| Test Data: | Image |
| Steps: | 1. Login 2. Main page will appear 3. Click on attendance menu 4. Drop down menu will appear 5. Click on mark attendance button 6. Attendance form will appear 7. Click on detect image. 8. Webcam will start video streaming. 9. If a system detects human face, it will highlight it with rectangle around the face |
| Status: | Pass |

**VIEW ATTENDANCE**

|  |  |
| --- | --- |
| Test Id: | TC4 |
| Tester: | Admin |
| Purpose: | View attendance record |
| Pre-requisites: | User must login |
| Steps: | 1. Login 2. Main page will appear 3. Click on attendance menu 4. Drop down menu will appear 5. Click on view attendance button 6. Attendance form will appear 7. User can view attendance record of student. |
| Status: | Pass |

**RECOGNIZE FACE**

|  |  |
| --- | --- |
| Test Id: | TC3 |
| Tester: | Admin |
| Purpose: | Recognize Face |
| Pre-requisites: | User must login |
| Steps: | 1. Login 2. Main page will appear 3. Click on attendance menu 4. Drop down menu will appear 5. Click on mark attendance button 6. Attendance form will appear 7. Click on detect and recognize image.   Webcam will start video streaming.   1. If a system detects human face, it will highlight it with rectangle around the face 2. If a system recognizes the face it will label it with the respective name of student. |
| Status: | PASS |

**ALGORITHM**

**Eigenfaces** is the name given to a set of [eigenvectors](https://en.wikipedia.org/wiki/Eigenvector) when they are used in the [computer vision](https://en.wikipedia.org/wiki/Computer_vision) problem of human [face recognition](https://en.wikipedia.org/wiki/Facial_recognition_system).[[1]](https://en.wikipedia.org/wiki/Eigenface#cite_note-1) The approach of using eigenfaces for [recognition](https://en.wikipedia.org/wiki/Facial_recognition_system) was developed by Sirovich and Kirby (1987) and used by Matthew Turk and [Alex Pentland](https://en.wikipedia.org/wiki/Alex_Pentland) in face classification.[[2]](https://en.wikipedia.org/wiki/Eigenface#cite_note-2) The eigenvectors are derived from the [covariance matrix](https://en.wikipedia.org/wiki/Covariance_matrix) of the [probability distribution](https://en.wikipedia.org/wiki/Probability_distribution) over the high-[dimensional](https://en.wikipedia.org/wiki/Dimension) [vector space](https://en.wikipedia.org/wiki/Vector_space) of face images. The eigenfaces themselves form a basis set of all images used to construct the covariance matrix. This produces dimension reduction by allowing the smaller set of basis images to represent the original training images. Classification can be achieved by comparing how faces are represented by the basis set.

[**Algorithmic Description**](http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#id33)

Let X = \{ x_{1}, x_{2}, \ldots, x_{n} \}be a random vector with observations x_i \in R^{d}.

1. Compute the mean \mu

\mu = \frac{1}{n} \sum_{i=1}^{n} x_{i}

1. Compute the the Covariance Matrix *S*

S = \frac{1}{n} \sum_{i=1}^{n} (x_{i} - \mu) (x_{i} - \mu)^{T}`

1. Compute the eigenvalues \lambda_{i}and eigenvectors v_{i}of S

S v_{i} = \lambda_{i} v_{i}, i=1,2,\ldots,n

1. Order the eigenvectors descending by their eigenvalue. The kprincipal components are the eigenvectors corresponding to the klargest eigenvalues.

The kprincipal components of the observed vector xare then given by:

y = W^{T} (x - \mu)

where W = (v_{1}, v_{2}, \ldots, v_{k}).

The reconstruction from the PCA basis is given by:

x = W y + \mu

where W = (v_{1}, v_{2}, \ldots, v_{k}).

The Eigen faces method then performs face recognition by:

* Projecting all training samples into the PCA subspace.
* Projecting the query image into the PCA subspace.
* Finding the nearest neighbour between the projected training images and the projected query image.

Still there’s one problem left to solve. Imagine we are given 400images sized 100 \times 100pixel. The Principal Component Analysis solves the covariance matrix S = X X^{T}, where {size}(X) = 10000 \times 400in our example. You would end up with a 10000 \times 10000matrix, roughly 0.8 GB. Solving this problem isn’t feasible, so we’ll need to apply a trick. From your linear algebra lessons you know that a M \times Nmatrix with M > Ncan only have N - 1non-zero eigenvalues. So it’s possible to take the eigenvalue decomposition S = X^{T} Xof size N \times Ninstead:

X^{T} X v_{i} = \lambda_{i} v{i}

and get the original eigenvectors of S = X X^{T}with a left multiplication of the data matrix:

X X^{T} (X v_{i}) = \lambda_{i} (X v_{i})

The resulting eigenvectors are orthogonal, to get orthonormal eigenvectors they need to be normalized to unit length. I don’t want to turn this into a publication, so please look into [[Duda01]](http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#duda01) for the derivation and proof of the equations.

**DATABASE CONNECTIVITY CODE**

import mysql.connector

import datetime

from time import strftime

from datetime import datetime as date

from os.path import join,basename

#from pyfacesgui import \*

#global student

#student=basename(matchfile)

class DatabaseUtility:

def \_\_init\_\_(self, database, tableName):

self.db = database

self.tableName = tableName

# Open database connection

self.cnx = mysql.connector.connect(user = 'root',password = '2406',host = '127.0.0.1')

# prepare a cursor object using cursor() method

self.cursor = self.cnx.cursor()

self.ConnectToDatabase()

#self.CreateTable()

def ConnectToDatabase(self):

try:

self.cnx.database = self.db

except mysql.connector.Error as err:

if err.errno == errorcode.ER\_BAD\_DB\_ERROR:

self.CreateDatabase()

self.cnx.database = self.db

else:

print(err.msg)

def CreateDatabase(self):

try:

self.RunCommand("CREATE DATABASE %s DEFAULT CHARACTER SET 'utf8';" %self.db)

except mysql.connector.Error as err:

print("Failed creating database: {}".format(err))

def Attendence(self, student, dt, dy , t):

cmd = "replace into attendence(enrollment,date,day,time) values ('%s','%s','%s','%s' );" %(student,dt,dy,t)

try:

self.cursor.execute(cmd);

except mysql.connector.Error as err:

self.cnx.rollback()

#print err

self.cnx.commit()

def \_\_del\_\_(self):

self.cnx.commit()

self.cursor.close()

self.cnx.close()

if \_\_name\_\_ == '\_\_main\_\_':

db = 'db'

tableName = 'attendence'

dbu = DatabaseUtility(db, tableName)

**AUTHENTICATION CODE**

#!/usr/bin/python

import sys

#reload(sys)

#sys.setdefaultencoding('utf8')

import mysql.connector

from mysql.connector import errorcode

from datetime import datetime

import PIL

from PIL import Image

class DatabaseUtility:

def \_\_init\_\_(self, database, tableName):

self.db = database

self.tableName = tableName

self.cnx = mysql.connector.connect(user = 'root',password = '2406',host = '127.0.0.1')

self.cursor = self.cnx.cursor()

self.ConnectToDatabase()

#self.CreateTable()

def ConnectToDatabase(self):

try:

self.cnx.database = self.db

except mysql.connector.Error as err:

if err.errno == errorcode.ER\_BAD\_DB\_ERROR:

self.CreateDatabase()

self.cnx.database = self.db

else:

print(err.msg)

def CreateDatabase(self):

try:

self.RunCommand("CREATE DATABASE %s DEFAULT CHARACTER SET 'utf8';" %self.db)

except mysql.connector.Error as err:

print("Failed creating database: {}".format(err))

def CreateTable(self):

cmd = (" CREATE TABLE IF NOT EXISTS " + self.tableName + " ("

" `ID` int(5) NOT NULL AUTO\_INCREMENT,"

" `username` char(50) NOT NULL,"

" `password` char(50) NOT NULL,"

" PRIMARY KEY (`ID`)"

") ENGINE=InnoDB;")

self.RunCommand(cmd)

def GetTable(self):

self.CreateTable()

return self.RunCommand("SELECT \* FROM %s;" % self.tableName)

def GetColumns(self):

return self.RunCommand("SHOW COLUMNS FROM %s;" % self.tableName)

def RunCommand(self, cmd):

print ("RUNNING COMMAND: " + cmd)

try:

self.cursor.execute(cmd)

except mysql.connector.Error as err:

print ('ERROR MESSAGE: ' + str(err.msg))

print ('WITH ' + cmd)

try:

msg = self.cursor.fetchall()

except:

msg = self.cursor.fetchone()

return msg

def AddEntryToAdmin(self, username, password):

cmd = " INSERT INTO " + self.tableName + " (username, password)"

cmd += " VALUES ('%s','%s');" % (username, password)

self.RunCommand(cmd)

def AddEntryToStudent(self, name, enrollment, branch, year):

cmd = " INSERT INTO " + self.tableName + "(enrollment,name,branch,year)"

cmd += "VALUES ('%s','%s','%s','%s');"%(enrollment,name,branch,year)

self.RunCommand(cmd)

def AddEntryToFaculty(self, name,subject):

cmd = " INSERT INTO " + self.tableName + "(name,subject)"

cmd += "VALUES ('%s','%s');"%(name,subject)

self.RunCommand(cmd)

def \_\_del\_\_(self):

self.cnx.commit()

self.cnx.close()

if \_\_name\_\_ == '\_\_main\_\_':

db = 'UsernamePassword\_DB'

tableName = 'masterTable'

dbu = DatabaseUtility(db, tableName)

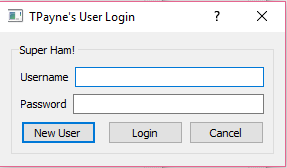
# dbu.AddEntryToTable ('asdf', 'asdf')

# print (dbu.GetColumns())

# print (dbu.GetTable())

**SCREENSHOTS**

**LOGIN FORM**

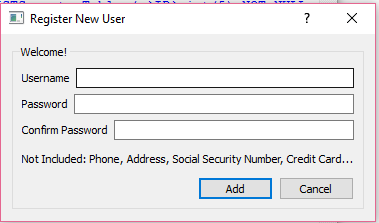


This login Form is made For Security purpose. So only Authenticated User only Access in to the Project. There are two Type of persons can enter in the project:

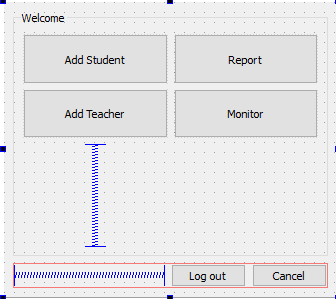
1. Administrator

2. User

**ADD NEW USER FORM:**

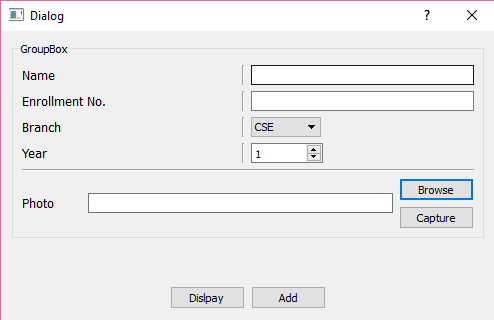


**ADD INFORMATION FORM**

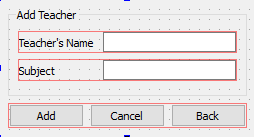


This form is showed when authorized administrator enters his correct User Name and Password. This Form gives the option to fill the name of Students and the name of Teacher if a new faculty has joined.

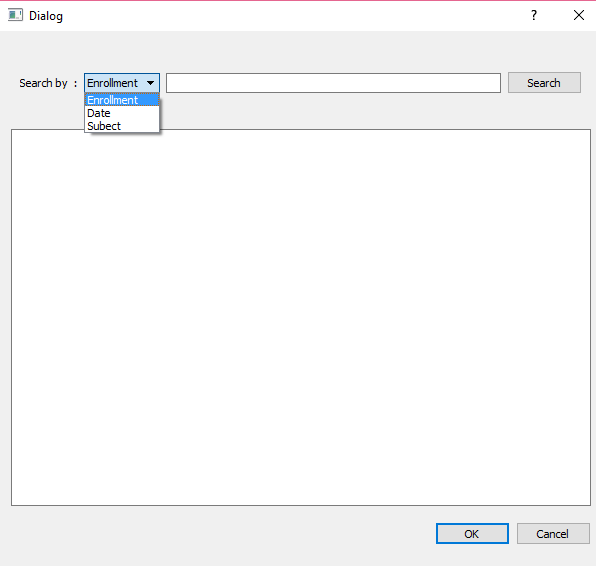
**ADD NEW STUDENT**



**ADD NEW TEACHER**



**TO SHOW DATABASE**



**REFERENCES**

* [*"Facial Recognition Applications"*](http://www.animetrics.com/technology/frapplications.html)*. Animetrics. Retrieved 2008-06-04.*
* [*"Facial Recognition: Who's Tracking You in Public?"*](http://www.consumerreports.org/privacy/facial-recognition-who-is-tracking-you-in-public1)*. Consumer Reports. Retrieved 2016-04-05.*
* [*"Airport Facial Recognition Passenger Flow Management"*](http://www.hrsid.com/company/technology/face-recognition)*. hrsid.com.*
* *Bonsor, K.* [*"How Facial Recognition Systems Work"*](http://computer.howstuffworks.com/facial-recognition.htm)*. Retrieved 2008-06-02.*
* *Smith, Kelly.* [*"Face Recognition"*](http://www.biometrics.gov/Documents/FaceRec.pdf) *(PDF). Retrieved 2008-06-04.*
* R. Brunelli and T. Poggio, "Face Recognition: Features versus Templates", IEEE Trans. on PAMI, 1993, (15)10:1042-1052
* R. Brunelli, *Template Matching Techniques in Computer Vision: Theory and Practice*, Wiley, [ISBN 978-0-470-51706-2](https://en.wikipedia.org/wiki/Special:BookSources/9780470517062), 2009 *(*[*[1]*](http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470517069.html) *TM book)*
* *Williams, Mark.* [*"Better Face-Recognition Software"*](http://www.technologyreview.com/Infotech/18796/?a=f)*. Retrieved 2008-06-02.*
* *Crawford, Mark.* [*"Facial recognition progress report"*](http://spie.org/x57306.xml)*. SPIE Newsroom. Retrieved 2011-10-06.*
* *Kimmel, Ron.* [*"Three-dimensional face recognition"*](http://www.cs.technion.ac.il/%7Eron/PAPERS/BroBroKimIJCV05.pdf) *(PDF). Retrieved 2005-01-01.*
* <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4430529&tag=1>
* <http://dl.acm.org/citation.cfm?id=1896448>
* [Face Recognition with OpenCV](http://docs.opencv.org/trunk/modules/contrib/doc/facerec/)