

Republic of Yemen

University of science and technology

Faculty of computing and information technology

Computer science Department



# **An Automatic class attendance system**

## **Using face detection and recognition**

Done by:

Omar Hasan **A**lbukhiati

Mohammed Abdurabu Al-erwi

Fares Obad Alzobah

Supervised by:

Dr. Sadik Al-Taweel

This Project Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Computer Science

Year 2017/2018

# ACKNOWLEDGEMENT

We would like to express our deep gratitude to Dr.Sadeq Al-taweel, our project supervisor for his patient guidance, valuable and constructive suggestions during the design and implementation of this project. His willingness to give his time so generously has been much appreciated.

Our grateful thanks are also extended to the staff at the department of Computer Science for their help and effort in teaching us throughout the past four years. we would also like to extend our thanks to our friends and classmates for their opinions, and for offering us the test data.

Finally, we would like to thank our families for their support and encouragement throughout our study.

# Abstract

The world today is preening great advances in technology and engineering. Cars are being modeled to fly, robots are built to serve and ensure our comfort, and much more that we cannot keep up with what is new. All the great advances have one goal in common which is to easier for humans and to increase productivity by automating tasks, and today we present a project that contributes achieving that good.

ACAS is an automatic class attendance system that utilizes face detection and recognition techniques to take the attendance of the students. By detecting their faces using a detection algorithm and marking their presence in the faculty's LMS and in the local database of the system.

This system consists of two components, the first one is the attendance module which takes the attendance of the students based on the information provided by the admin. The second component is the web-based administration portal which enables teachers and the admin to enter info and configure the system.

# Table of contents

<b><u>Chapter 1</u></b>	<b>10</b>
1.1 <a href="#">Introduction</a> .....	11
1.2 <a href="#">Problem statements</a> .....	12
1.3 <a href="#">Objectives</a> .....	12
1.4 <a href="#">Scope and limitations:</a> .....	13
1.5 <a href="#">Methodology</a> .....	13
1.6 <a href="#">Tools</a> .....	15
1.7 <a href="#">The plan:</a> .....	16
1.8 <a href="#">The research organization:</a> .....	17
<b><u>Chapter 2</u></b>	<b>18</b>
2.1 <a href="#">Introduction</a> .....	19
2.2 <a href="#">Background</a> .....	19
2.3 <a href="#">Related work</a> .....	38
2.4 <a href="#">Proposed System</a> .....	39
2.5 <a href="#">Chapter Summery</a> .....	39
<b><u>Chapter 3</u></b>	<b>40</b>
3.1 <a href="#">Introduction</a> .....	41
3.2 <a href="#">Requirements</a> .....	41
3.2.2 <a href="#">System</a> .....	42
3.3 <a href="#">Use Case Diagram</a> .....	43
3.4 <a href="#">Entity Relationship Diagram (ERD)</a> .....	48
3.5 <a href="#">class Diagram</a> .....	49
<b><u>Chapter 4</u></b>	<b>50</b>
4.1 <a href="#">Introduction</a> .....	51
4.2 <a href="#">Architectural Design</a> .....	51
4.3 <a href="#">Database design</a> .....	52
4.4 <a href="#">Database Indexing</a> .....	53
4.5 <a href="#">Class Diagram</a> .....	55
4.6 <a href="#">Sequence diagram</a> .....	56
4.7 <a href="#">System Interface</a> .....	58
<b><u>Chapter 5</u></b>	<b>62</b>
5.1 <a href="#">Introduction</a> .....	63
5.2 <a href="#">System components</a> .....	63
5.2.1 <a href="#">The automatic attendance component</a> .....	63

5.2.2	<a href="#">working process</a> .....	65
5.2.3	<a href="#">web based administration</a> .....	68
<a href="#">Chapter 6</a>		72
6.1	<a href="#">Introduction</a> .....	73
6.2	<a href="#">Conclusion</a> .....	73
6.4	<a href="#">Future work</a> .....	74
<a href="#">References</a>		75

# Table of Figures

<a href="#"><u>Figure 1. 1 Waterfall methodology</u></a> .....	14
<a href="#"><u>Figure 1. 2 the plan</u></a> .....	16
<a href="#"><u>Figure 2. 1 Haar like Features</u></a> .....	24
<a href="#"><u>Figure 2. 2 Haar-like features with different sizes and orientation</u></a> .....	24
<a href="#"><u>Figure 2. 3 How the Haar like feature of figure 2.2 can be used to scale the eyes</u></a> .....	25
<a href="#"><u>Figure 2. 4 several classifiers combined to enhance face detection</u></a> .....	26
<a href="#"><u>Figure 2. 5 Pixel Coordinates of an integral image</u></a> .....	26
<a href="#"><u>Figure 2. 6 Values of the integral Image on a rectangle</u></a> .....	27
<a href="#"><u>Figure 2. 7 The average face, the first and last eigenface which is mainly noise</u></a> .....	29
<a href="#"><u>Figure 2. 8 The dominant 32 eigenfaces representing all the images in the set</u></a> .....	30
<a href="#"><u>Figure 2. 9 A training set consisting of M images</u></a> .....	30
<a href="#"><u>Figure 2. 10 The free space of vectors obtained by converting the N x N images</u></a> .....	32
<a href="#"><u>Figure 2. 11 The average face is represented by the blue column labelled U</u></a> .....	32
<a href="#"><u>Figure 2. 12 The covariance matrix</u></a> .....	33
<a href="#"><u>Figure 2. 13 The reduced dimensionality matrix in the low dimensional space</u></a> .....	34
<a href="#"><u>Figure 2. 14 The selected K eigenvectors that can represent the whole set of images</u></a> .....	34
<a href="#"><u>Figure 2. 15 The selected K eigenvectors being mapped in the high order</u></a> .....	35
<a href="#"><u>Figure 2. 16 Training set images as a weighted sum of the eigenvectors</u></a> .....	36
<a href="#"><u>Figure 2. 17 Weights of the eigenvectors/eigenface</u></a> .....	36
<a href="#"><u>Figure 2. 18 The PCA based Eigenfaces face recognition algorithm flow chart</u></a> .....	37
<a href="#"><u>Figure 3. 1 Use Case Diagram</u></a> .....	43
<a href="#"><u>Figure 3. 2 ERD Diagram</u></a> .....	48
<a href="#"><u>Figure 3. 3 class diagram</u></a> .....	49
<a href="#"><u>Figure 4. 1 Architectural Design</u></a> .....	51
<a href="#"><u>Figure 4. 2 Relational diagram</u></a> .....	52
<a href="#"><u>Figure 4. 3 class diagram</u></a> .....	55
<a href="#"><u>Figure 4. 4 sequence diagram (recognize faces)</u></a> .....	56
<a href="#"><u>Figure 4. 5 sequence diagram (manage &amp; configure system)</u></a> .....	57
<a href="#"><u>Figure 4. 6 login interface</u></a> .....	58
<a href="#"><u>Figure 4. 7 Home interface</u></a> .....	58
<a href="#"><u>Figure 4. 8 students interface</u></a> .....	59
<a href="#"><u>Figure 4. 9 add student interface</u></a> .....	59

<a href="#"><u>Figure 5. 1 add student interface</u></a> .....	68
<a href="#"><u>Figure 5. 2 schedule interface</u></a> .....	69
<a href="#"><u>Figure 5. 3 edit lecture interface</u></a> .....	69
<a href="#"><u>Figure 5. 4 edit lecture interface</u></a> .....	70
<a href="#"><u>Figure 5. 5 report interface</u></a> .....	70
<a href="#"><u>Figure 5. 6 report interface</u></a> .....	71
<a href="#"><u>Figure 5. 7 student report interface</u></a> .....	71

## List of Tables

<a href="#">Table 1. 1 software tools</a> .....	15
<a href="#">Table 1. 2 hardware tools</a> .....	15
<a href="#">Table 3. 1 detect faces</a> .....	44
<a href="#">Table 3. 2 recognize faces</a> .....	45
<a href="#">Table 3. 3 record attendance</a> .....	46
<a href="#">Table 3. 4 manipulate information</a> .....	47
<a href="#">Table 4. 1 admin</a> .....	53
<a href="#">Table 4. 2 students</a> .....	53
<a href="#">Table 4. 3 subject</a> .....	53
<a href="#">Table 4. 4 lecture</a> .....	54
<a href="#">Table 4. 5 studsub</a> .....	54
<a href="#">Table 4. 6 hall</a> .....	54
<a href="#">Table 4. 7 attendance</a> .....	55



## **List of abbreviation**

LMS	Learning management system
ERP	Enterprise resource planning
ERD	Entity relationship diagram
OpenCV	Open computer vision
EIRJET	International researcher journal of engineering and technology
LDA	Linear discriminate analysis
PCA	Principal component analysis

# **Chapter 1**

## **Introduction**

# Introduction

In life, technologies are advancing in a rapid phase, and because of the necessity to employ these technologies in our lives to solve problems or may be ease tasks for people, societies are going towards having the technologies in every aspect of our lives. In the faculty of computing and information technology , and as part of our responsibility, the researchers wanted to employ the technologies, we studied, in making the attendance taking in the faculty automatic by designing a system that would do that. The system would take the attendance of students in the various lecture halls, and store it in the Learning Management System (LMS), and the steps are as follows.

First the system takes the pictures with cameras installed in the halls and detects the faces and recognizes them by comparing the faces with preprocessed pictures of the students stored in a database.

Secondly, The system connects to the LMS and marks the presence and absence of the students according to the schedule that is pre-entered into the system by the system admin. The cameras installed in every lecture hall will be directly connected and mapped to the server enabling it to turn on and off the cameras accordingly .

# Problem statements

In the faculty the attendance used to be taken in papers, and then the LMS came along and teachers were required to use it to take the attendance, both papers and the LMS are facing a lot of problems and some are as follow.

## ❖ **problems with papers**

- Attendance papers may get lost as the teacher collects many papers for many levels and classes.
- The teacher may ask a student to take the attendance, and that student might present students that are not.

## ❖ **problems with the LMS**

- Some subjects do not require a computer and accessing the LMS requires one.
- Accessing the LMS during the class might be impossible due to a network problem.
- A teacher might forget to take the attendance.

# Objectives

This system aims at accomplishing a couple of goals such as:

- 1- Automating the attendance taking process.
- 2- Increasing the accuracy of the attendance by preventing the problems that traditional methods have.
- 3- Encouraging students to be punctual.
- 4- Urging teachers and students to stick to the halls assigned for classes.

## **Scope and limitations:**

### **1.4.1 Geographical scope**

The place where this system functions is the faculty of computing and information technology, or any other faculty in the university .

### **1.4.2 Functional scope**

The system should be able to:

- 1- Detect and recognize faces belonging to a specific class.
- 2- Connect with the LMS to store the attendance.
- 3- Produce reports of the attendance of a class a semester.

### **1.4.3 Limitations**

1- If the student has not enrolled in the subject on the LMS and enrolled in the ERP system, The system will not take his presence.

## **Methodology**

In software engineering the development methodologies are a lot such as the waterfall Model which is designed in a waterfall-like model, where a phase is first completed then we move to second one, in the waterfall model we can't go back to a previous phase, Also the agile method ,in which we minimize the risk by developing software in short time boxes which are called iterations, and we can alter an iteration from the beginning.

In addition to the agile method we have served other methodologies, like spired model, extreme programming, and others.

for this system we are using the waterfall model because, the system is critical and the requirements are clear at the beginning and its phases are requirements,design,implementation,verification,and maintenance.

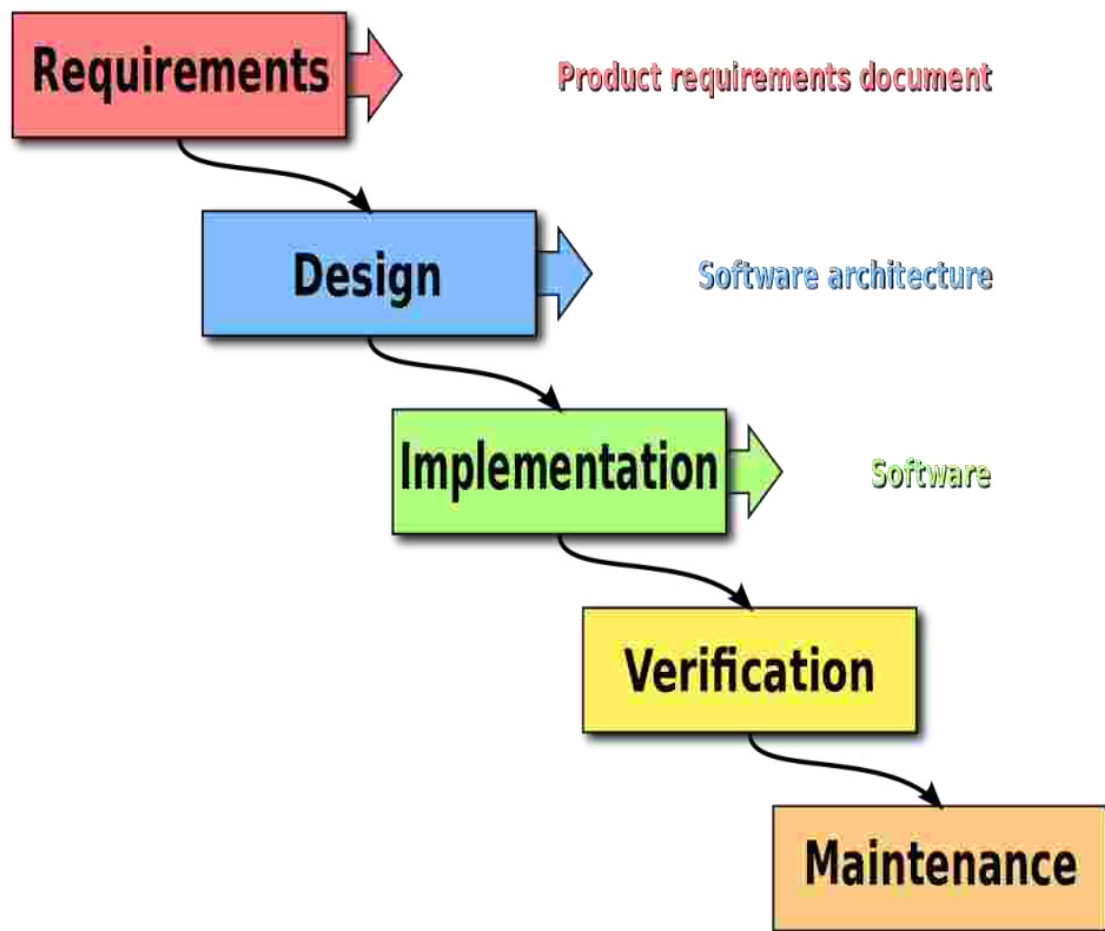


Figure 1. 1 Waterfall methodology

# Tools

## 1.1.1 software tools:

*Table 1. 1 software tools*

Tools	Usage
1- python 3.5	Language the system written in
2- windows 10	Operating system
3- MySQL	DBMS for storing the schedule and the attendance
4- openCV	Computer vision libraries
5- moodle 3.9	The lms

## 1.1.2 Hardware tools:

*Table 1. 2 hardware tools*

Tools	Usage
1- Camera	To take a pictures of students
2- Computers	To running the system .

# The plan:

In this part we specify the phases and time table in which this project is completed .we begin with the requirements and analyses phase then we move to the design and implementation and then verification

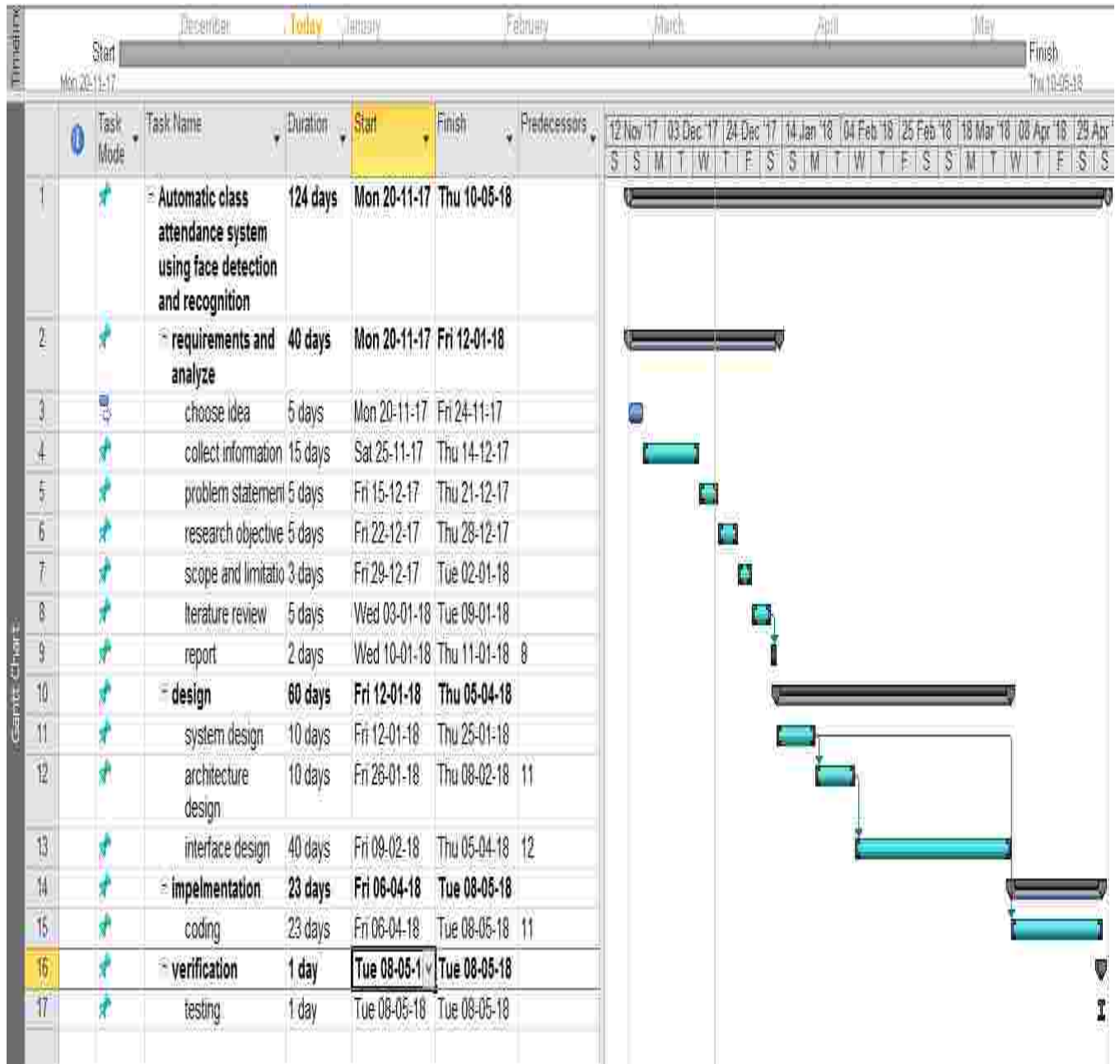


Figure 1. 2 the plan



## **The research organization:**

The research consists of six chapter:

The first chapter talks about an introduction to the research and the problems statements and the goals, and the tools used by the system.

In chapter tow we well talk about the literature review and the concepts relation to our work such .

In the third chapter we well analyze the system by giving the use case diagram along with the sue case specifications and then the ERD followed by the class diagram .In chapter four design of the system is introduced,.

the implementation of the system is given in chapter five in which we described the components of the system and provided a thorough test . Finally, the conclusion of the research and any recommendations are in chapter six.

# **Chapter 2**

## **Literature Review**

# Introduction

In this chapter, the researchers will briefly define the concept of image processing and its types. Followed by answers to some common questions we usually ask about image processing like, how are digital images represented in computer systems, and how do computers process them.

Additionally, and because the field of image processing is experiencing a lot of research focus, many face detection and recognition approaches rose, yielding many different algorithms, and a description of these approaches is given along with the proposed system.

Finally, the chosen algorithms for both face detection and recognition are explained in detail, along with the reasons why they were chosen.

## Background

### Image Processing

**Image processing** is processing of images using mathematical operations by using any form of signal processing for which the input is an image, a series of images, or a video, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Images are also processed as three-dimensional signals where the third-dimension being time or the z-axis. [1]

### Types of image processing

**Digital image processing** is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during

processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems.

In electrical engineering and computer science, **analog image processing** is any image processing task conducted on two-dimensional analog signals by analog means (as opposed to digital image processing). Basically any data can be represented in two types named as 1. Analog 2. Digital if the pictorial representation of the data represented in analog wave formats that can be named as analog image. **E.g.** television broadcasting in older days. through the dish antenna systems. Where as the digital representation or storing the data in digital form is termed as a digital image processing **e.g.** image data stored in digital logic gates. [2]

## **Image Representation in a Digital Computer**

An image is a 2-Dimensional light intensity function  $f(x, y) = r(x, y) \times i(x, y)$  Where  $r(x, y)$ , is the reflectivity of the surface of the corresponding image point.  $i(x, y)$  Represents the intensity of the incident light. A digital image  $f(x, y)$  is discretized both in spatial co-ordinates by grids and in brightness by quantization.[3] Effectively, the image can be represented as a matrix whose row, column indices specify a point in the image and the element value identifies gray level value at that point. These elements are referred to as pixels or pels. Typically following image processing applications, the image size which is used is  $256 \times 256$ , elements,  $640 \times 480$  pels or  $1024 \times 1024$  pixels. Quantization of these matrix pixels is done at 8 bits for black and white images and 24 bits for colored images (because of the three color planes Red, Green and Blue each at 8 bits). [4]

## **Steps in Digital Image Processing**

[2.8] Digital image processing involves the following basic tasks:

- Image Acquisition - An imaging sensor and the capability to digitize the signal produced by the sensor.
- Preprocessing – Enhances the image quality, filtering, contrast enhancement etc.

- Segmentation – Partitions an input image into constituent parts of objects.
- Description/feature Selection – extracts the description of image objects suitable for further computer processing.
- Recognition and Interpretation – Assigning a label to the object based on the information provided by its descriptor. Interpretation assigns meaning to a set of labeled objects.
- Knowledge Base – This helps for efficient processing as well as inters module cooperation.

## **Definition of Terms**

### **2.1.1.1 Face Detection**

Face detection is the process of identifying and locating all the present faces in a single image or video regardless of their position, scale, orientation, age and expression. Furthermore, the detection should be irrespective of extraneous illumination conditions and the image and video content. [5]

### **2.1.1.2 Face Recognition**

Face Recognition is a visual pattern recognition problem, where the face, represented as a three dimensional object that is subject to varying illumination, pose and other factors, needs to be identified based on acquired images [6].

## **Face Detection Approaches**

[8] Some of the main face detection methods are discussed here.

- 1) Knowledge based methods are developed on the rules derived from the researchers knowledge of human faces. Problem in this approach is the difficulty in translating human knowledge into well-defined rules.
- 2) Featured-based methods: Invariant features of faces are used for detecting texture, skin color. But features from such algorithm can be severely corrupted due to illumination, noise and occlusion.
- 3) Template matching: Input image is compared with predefined face template. But the performance here suffers due to variations in scale, pose and shape.

- 4) Appearance-based method: In template matching methods, the templates are predefined by experts. Whereas, the templates in appearance based methods are learned from examples in images.
- 5) Statistical analysis and machine learning techniques can be used to find the relevant characteristics of face and non-face images

## **Face Recognition Approaches [8]**

- 1) Local feature analysis method of recognition analyzes the face in terms of local features, e.g., eyes, nose, etc. referred to as LFA kernels. LFA technique offers better robustness against local variations on the facial image in carrying out a match, but does not account for global facial attributes.
- 2) Neural Network are based on learning of the faces in an example set by the machine in the `training phase and carrying out recognition in the `generalization phase. But in order to succeed in a practical set-up, the examples should sufficiently large in number to account for variations in real life situations.
- 3) Model Matching methods of face recognition (like Hidden Markov Model (HMM)[6] ) train a model for every person during model learning and choose the best matching model, given a query image. Here also a big realistic representative models is necessary for good results.
- 4) A recognition system based on sparse representation computed by  $l_1$  -minimization works with the basic idea of casting the recognition as a sparse representation problem. The main concern in this approach is the presence of sufficiently large number of features and correct computation of sparse representation. It is a robust and scalable algorithm for face recognition based on linear or convex programming.

## **Voila-Joins Algorithm** [13]

A face Detector has to tell whether an image of arbitrary size contains a human face and if so, where it is. Face detection can be performed based on several cues: skin color (for faces in color images and videos, motion (for faces in videos), facial/head shape, facial appearance or a combination of these parameters. Most face detection algorithms are appearance based without using other cues. An input image is scanned at all possible locations and scales by a sub window. Face detection is posed as classifying the pattern in the sub window either as a face or a non-face. The face/non-face classifier is learned from face and non-face training examples using statistical learning methods.

The characteristics of Viola–Jones algorithm which make it a good detection algorithm are:

- Robust – very high detection rate (true-positive rate) & very low false-positive rate always.
- Real time – For practical applications at least 2 frames per second must be processed.
- Face detection only (not recognition) - The goal is to distinguish faces from non-faces (detection is the first step in the recognition process)

The algorithm has four stages:

- 1) Haar Feature Selection.
- 2) Creating an Integral Image.
- 3) Adaboost Training.
- 4) Cascading Classifiers.

Most modern algorithms are based on the Viola Jones object detection framework, which is based on Haar Cascades.

### **❖ Haar-Cascade**

Haar like features are rectangular patterns in data. A cascade is a series of “Haar-like features” that are combined to form a classifier. A Haar wavelet is a mathematical function that produces square wave output.

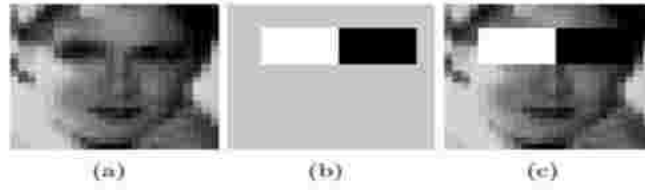


Figure 2. 1 Haar like Features

Figure 2.1 shows Haar like features, the background of a template like (b) is painted gray to highlight the pattern's support. Only those pixels marked in black or white are used when the corresponding feature is calculated [13]. Since no objective distribution can describe the actual prior probability for a given image to have a face, the algorithm must minimize both the false negative and false positive rates in order to achieve an acceptable performance. This then requires an accurate numerical description of what sets human faces apart from other objects. Characteristics that define a face can be extracted from the images with a remarkable committee learning algorithm called Adaboost. Adaboost (Adaptive boost) relies on a committee of weak classifiers that combine to form a strong one through a voting mechanism. A classifier is weak if, in general, it cannot meet a predefined classification target in error terms. The operational algorithm to be used must also work with a reasonable computational budget. Such techniques as the integral image and attentional cascades have made the Viola-Jones algorithm highly efficient: fed with a real time image sequence generated from a standard webcam or camera, it performs well on a standard PC [12].

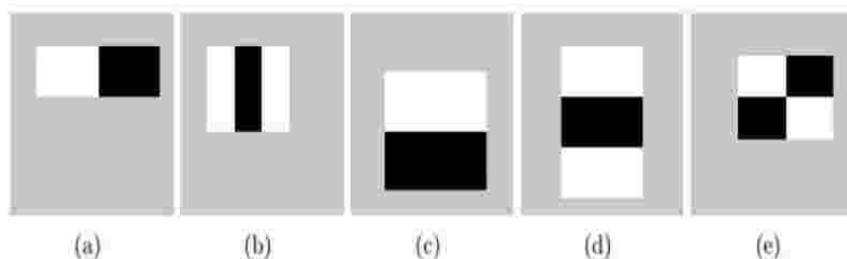


Figure 2. 2 Haar-like features with different sizes and orientation



The size and position of a pattern's support can vary provided its black and white rectangles have the same dimension, border each other and keep their relative positions. Thanks to this constraint, the number of features one can draw from an image is somewhat manageable: a  $24 \times 24$  image, for instance, has 43200, 27600, 43200, 27600 and 20736 features of category (a), (b), (c), (d) and (e) respectively as shown in figure 2.2, hence 162336 features in all.

In practice, five patterns are considered. The derived features are assumed to hold all the information needed to characterize a face. Since faces are large and regular by nature, the use of Haar-like patterns seems justified [14].

### ❖ How the Haar – like Features Work. [8]

A scale is chosen for the features say  $24 \times 24$  pixels. This is then slid across the image. The average pixel values under the white area and the black area are then computed. If the difference between the areas is above some threshold then the feature matches.

In face detection, since the eyes are of different color tone from the nose, the Haar feature (b) from Figure 2.3 can be scaled to fit that area as shown below [3],



*Figure 2. 3 How the Haar like feature of figure 2.2 can be used to scale the eyes*

One Haar feature is however not enough as there are several features that could match it (like the zip drive and white areas at the background of the image of figure 2.3). A single classifier therefore isn't enough to match all the features of a face, it is called a "weak classifier." Haar cascades, the basis of

Viola Jones detection framework therefore consist of a series of weak classifiers whose accuracy is at least 50% correct. If an area passes a single classifier, it moves to the next weak classifier and so on, otherwise, the area does not match [13].

○ **Cascaded Classifier**

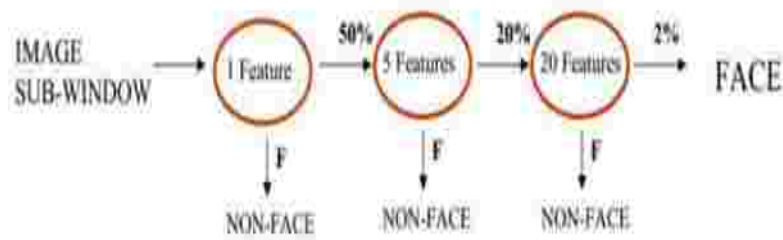


Figure 2. 4 several classifiers combined to enhance face detection

before Haar cascades, was utilized to combine a series of weak classifiers in to a strong classifier. Adaboost tries out multiple weak classifiers over several rounds, selecting the best weak classifier in each round and combining the best weak classifier to create a strong classifier. Adaboost can use classifiers that are consistently wrong by reversing their decision. In the design and development, it can take weeks of processing time to determine the final cascade sequence.

After the final cascade had been constructed, there was a need for a way to quickly compute the Haar features i.e. compute the differences in the two areas. The integral image was instrumental in this.

○ **Integral Image**

The Integral image also known as the “summed area table” developed in 1984 came in to

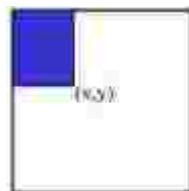


Figure 2. 5 Pixel Coordinates of an integral image

widespread use in 2001 with the Haar cascades. A summed area table is created in a single pass. This makes the Haar cascades fast, since the sum of any region in the image can be computed using a single formula.

The integral image computes a value at each pixel  $(x, y)$  as is shown in figure 2.5, that is the sum of the pixel values above and to the left of  $(x, y)$ , inclusive. This can quickly be computed in one pass through the image.

Let A, B, C, D be the values of the integral image at the corners of a rectangle as shown in figure 2.6. The sum of original image values within the rectangle can be computed.

$$Sum = A - B - C + D \quad (2.1)$$

Only three additions are required for any size of rectangle. This face detection approach minimizes computation time while achieving high detection accuracy. It is now used in many areas of computer vision.

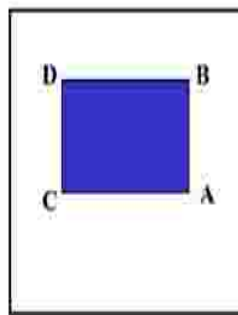


Figure 2.6 Values of the integral Image on a rectangle

The size and position of a pattern's support can vary provided its black and white rectangles have the same dimension, border each other and keep their relative positions. Thanks to this constraint, the number of features one can draw from an image is somewhat manageable: a  $24 \times 24$  image, for instance, has 43200, 27600, 43200, 27600 and 20736 features of category (a), (b), (c), (d) and (e) respectively as shown in figure 2.2, hence 162336 features in all.

In practice, five patterns are considered. The derived features are assumed to hold all the information needed to characterize a face. Since faces are large and regular by nature, the use of Haar-like patterns seems justified.

## **PCA-based Eigenface Algorithm [9]**

### **➤ Why PCA Based Eigenfaces method?**

The PCA based eigenfaces method has the best overall recognition efficiency of up to seventy percent (70%), as compared to the rest of the existing face recognition algorithms. PCA also has a low memory requirement, low computational complexity and takes less time to execute. These factors along with the limited timeframe of implementing the project informed the choice of PCA based Eigenfaces as the face recognition method.

Most systems in practice employ this technique by use of one camera that is fixed at one point in a room (say the front of a classroom). This method is limited in terms of distance of the detected faces and the quality. Our chosen method uses a mobile camera which will be hand held by the user. The captured image will thus be of high quality minimizing false positives when it comes to the recognition stage. It also improves the results by extracting the detected images within the rectangle and converting them to grayscale. This will tremendously reduce inaccuracies due to posture, illumination, background noise and different recognition results for light and dark skinned detected faces.

### **➤ PCA and its relation to Face Recognition**

Principal Component Analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of values of possibly correlated  $M$  variables (faces) into a set of values of  $K$  uncorrelated variables called principle components (eigenvectors).

The number of principal components (Eigenfaces) is always less than or equal to the number of original variables (face images) i.e.  $K \leq M$ .

This transformation is defined in such a way that the first eigenface shows the most dominant “direction/feature” of the training set of images and each succeeding component in turn shows the next most possible dominant “direction/feature”; all under the constraint that it be uncorrelated to the preceding Eigenface.

To reduce the calculations needed for finding these eigenfaces, the dimensionality of the original training set is reduced before they are calculated.

Since eigenfaces show the “directions” of data and each preceding eigenface shows less “directions” and more “noise”, only few first eigenface (say K) are selected whereas the last of the eigenfaces are discarded [7].



*Figure 2. 7 The average face, the first and last eigenface which is mainly noise*

Figure 2.7 shows the average face and the first and last eigenfaces that were generated from a collection of 30 images each of 4 people. The average face shows the smooth face structure of a generic person, the first few eigenfaces will show some dominant features of faces, and the last eigenfaces (e.g.: Eigenface 119) are mainly image noise. Figure 2.8 shows the first 32 eigenfaces. Images 32 - 119 are discarded because they are mainly noise [7].

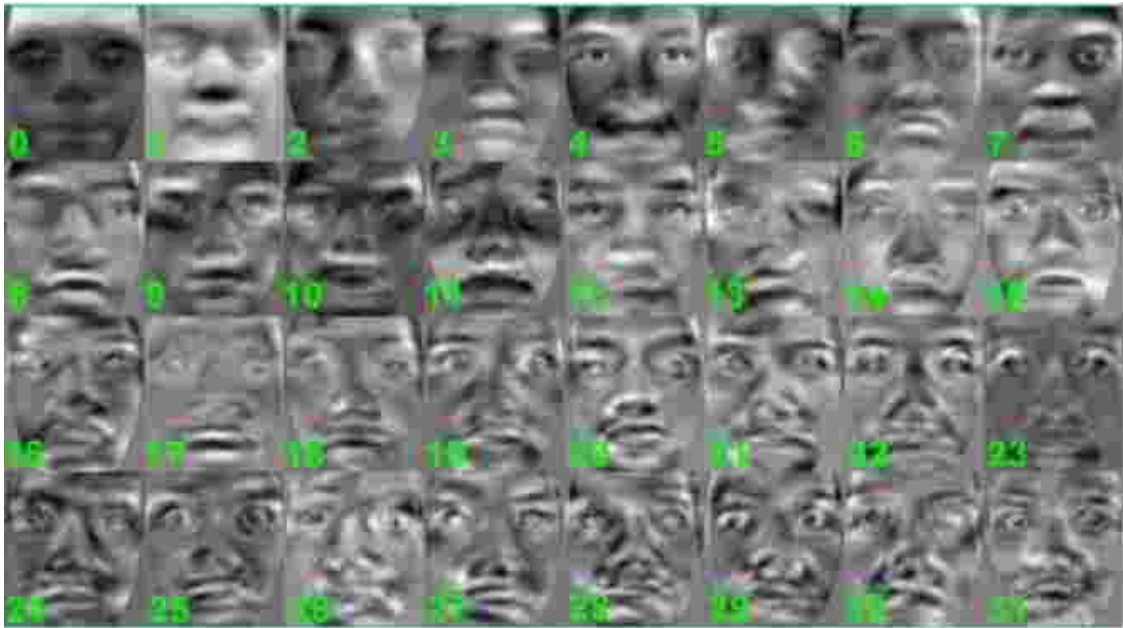


Figure 2. 8 The dominant 32 eigenfaces representing all the images in the set

Representing an image as a combination of  $K$  Eigenfaces reduces the number of values needed to recognize it from  $M$  to  $K$ . This makes the recognition process faster and free of errors caused by noise.

➤ **Working Principle of PCA Eigenface for Face Recognition. [15]**

Given a training set of  $M$  images as shown in the figure 2.9 and an unknown face all of the same size, PCA Eigenface method aims at representing the face image as a linear combination of a set of Eigenfaces/Eigenvectors as in figure 2.8.

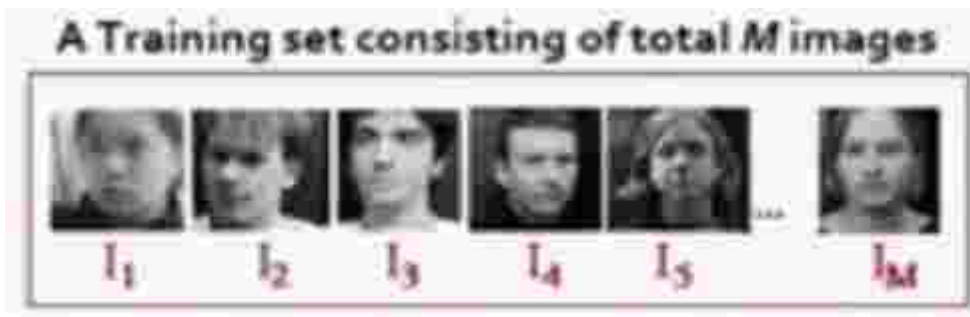


Figure 2. 9 A training set consisting of  $M$  images

These Eigenfaces (eigenvectors) are in fact the principle components of the training set of face images generated after reducing the dimensionality of the training set.

Once Eigenfaces are selected, each training set image is represented in terms of these eigenfaces. When an unknown face comes for recognition, it is also represented in terms of the selected eigenfaces.

The eigenface representation of the unknown face is compared with that of each training set face image. The distance between them is calculated. If the distance is above some specified threshold, then it recognizes the unknown face as that person. PCA Eigenfaces method considers each pixel in an image as a separate dimension. E.g. A  $50 * 50$  image = 2500 pixels thus has 2500 dimensions. This method does not work on images directly, it first converts them to a matrix (vector) form.

➤ **PCA Face Recognition Algorithm. [8]**

Step 0: Create a training set and load it.

The training set consists of total  $M$  images as shown in Figure 2.9. Where each image is of size  $N \times N$

Step 1: Convert the images in the training set to face vectors.

The images are converted in to column vectors. Let a face image be a two dimensional  $N$  by  $N$  array of 8 bit intensity values. An image may also be considered as a vector of dimension  $N$  squared. So that a typical image of size 50 by 50 becomes a vector of dimension 2500 dimensional space. An ensemble of images, then, maps to a collection of points on this huge space. All images are then converted in to column vectors of  $N$  squared rows by one. These are represented by  $T_i$  as in figure 2.10.

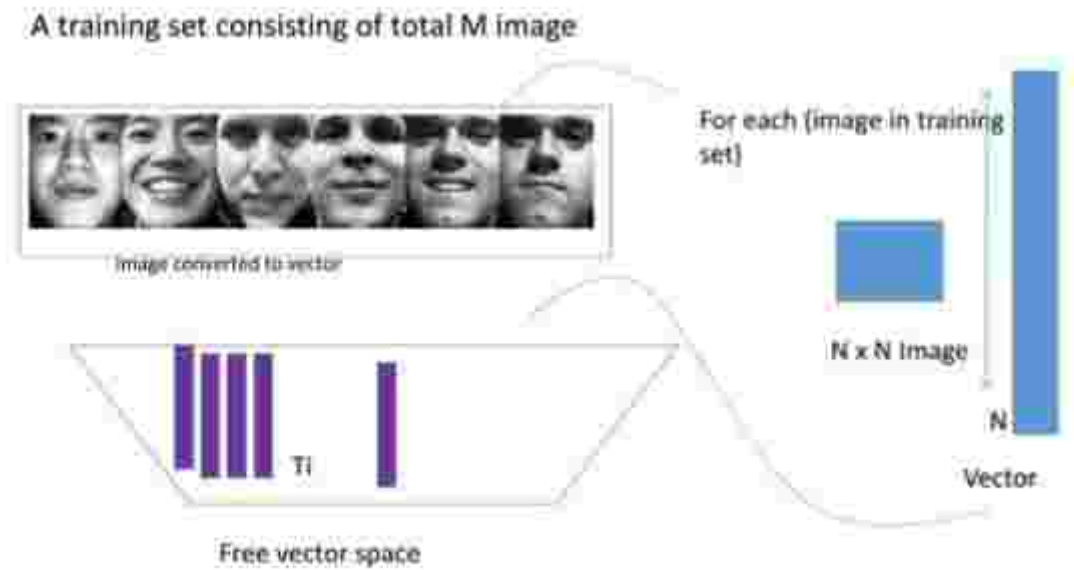


Figure 2. 10 The free space of vectors obtained by converting the  $N \times N$  images

Step 2: Normalize the face vectors i.e. remove all the common features. This is done by calculating the average face vector and then subtracting it from all the face vectors. Figure 2.11 shows the



Figure 2. 11 The average face is represented by the blue column labelled  $U$

average face ( $U$ ) which is then subtracted from all the face vectors.

Step 3: Calculate the eigenvectors from the covariance matrix. We now calculate the eigenvectors from the covariance matrix since in PCA based Eigenfaces method the principal components are obtained from the covariance matrix. The covariance matrix  $C$  is given by



$$C = A A^T \quad - (2.2)$$

Where  $A \equiv [\mathbf{O}_1, \mathbf{O}_2, \mathbf{O}_3 \dots \mathbf{O}_M]$  and is of dimension  $N^2 \times M$  - (2.3)

$$C = A A^T = N^2 \times M \times M \times N^2 = N^2 \times N^2 \quad - (2.4)$$

This is a very huge matrix as depicted in Figure 2.15 where  $N = 50, 2500 N^2$ . This would yield 2500 2500 Eigenfaces

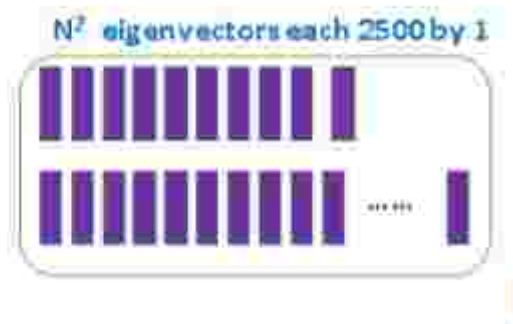


Figure 2. 12 The covariance matrix

To find K eigenvectors from M = 2500 eigenvectors with  $K < M$  would take a lot of time due to the many calculations that would need to be done and hence the need for Step 3 Dimensionality reduction.

Step 4: Reduce the dimensionality of the training set.

To reduce these calculations on the needed eigenfaces, we calculate them from a covariance matrix of reduced dimensionality.

Where

$$C = A^T A = M \times N^2 \times N^2 \times M \equiv M \times M = 100 \times 100 \quad - (2.5)$$

IF  $M = 100$

This would give 100 eigenvectors as shown in Figure 2.13.

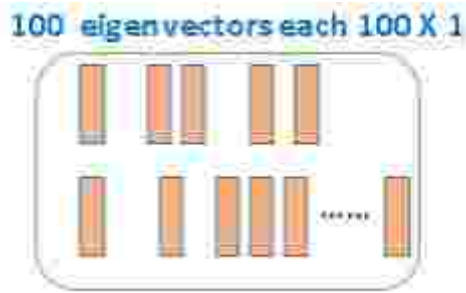


Figure 2. 13 The reduced dimensionality matrix in the low dimensional space

Step 5: Select  $K$  best eigenvectors such that  $K < M$  and can represent the whole training set. The yellow bars represent the selected  $K$  Eigen vectors that are sufficient to represent the whole training set as in Figure 2.14.

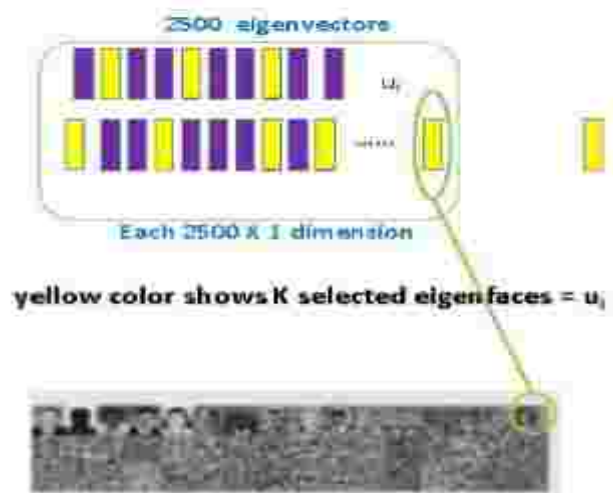


Figure 2. 14 The selected  $K$  eigenvectors that can represent the whole set of images

Step 6: Convert lower dimensional  $K$  eigenvectors into original face dimensionality as in Figure 2.15.

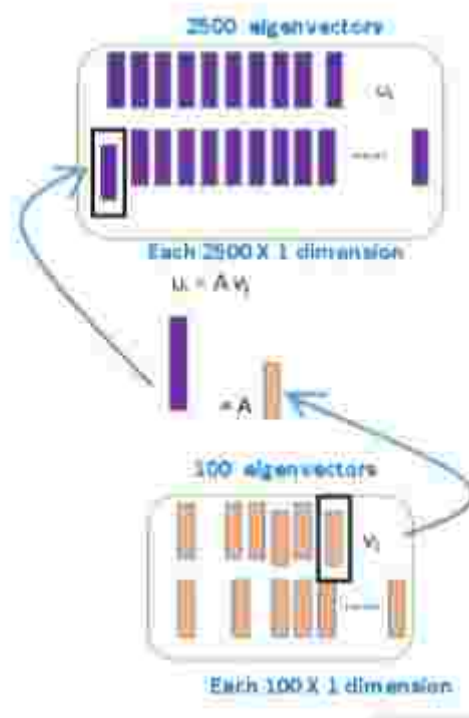


Figure 2. 15 The selected K eigenvectors being mapped in the high order dimensional space

$$U_i = AV_i \quad - (2.6)$$

$U_i$  = ith eigenvector in the higher dimensional space

$V_i$  = ith eigenvector in the lower dimensional space

Step 7: Represent each image as a linear combination of all K eigenvectors plus the mean/average face.

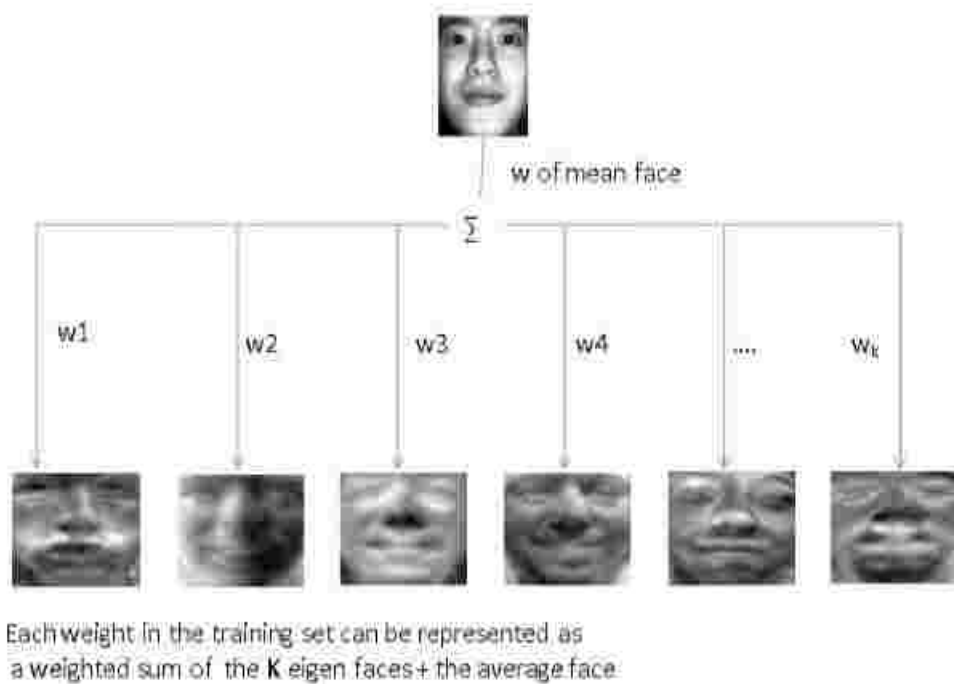


Figure 2. 16 Training set images as a weighted sum of the eigenvectors and the average face

Step 8: For each image in the training set calculate and store the associated weight vectors.

A weighted face vector  $\Omega$  which is the eigenface representation of the  $i$ th face weight vector for each face is calculated as shown in Figure 2.17

$$\Omega = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix}$$

Figure 2. 17 Weights of the eigenvectors/eigenface

The face recognition algorithm flow chart of Figure 2.18 is basically a method of checking which training image is most similar to the input image, out of the whole training set.

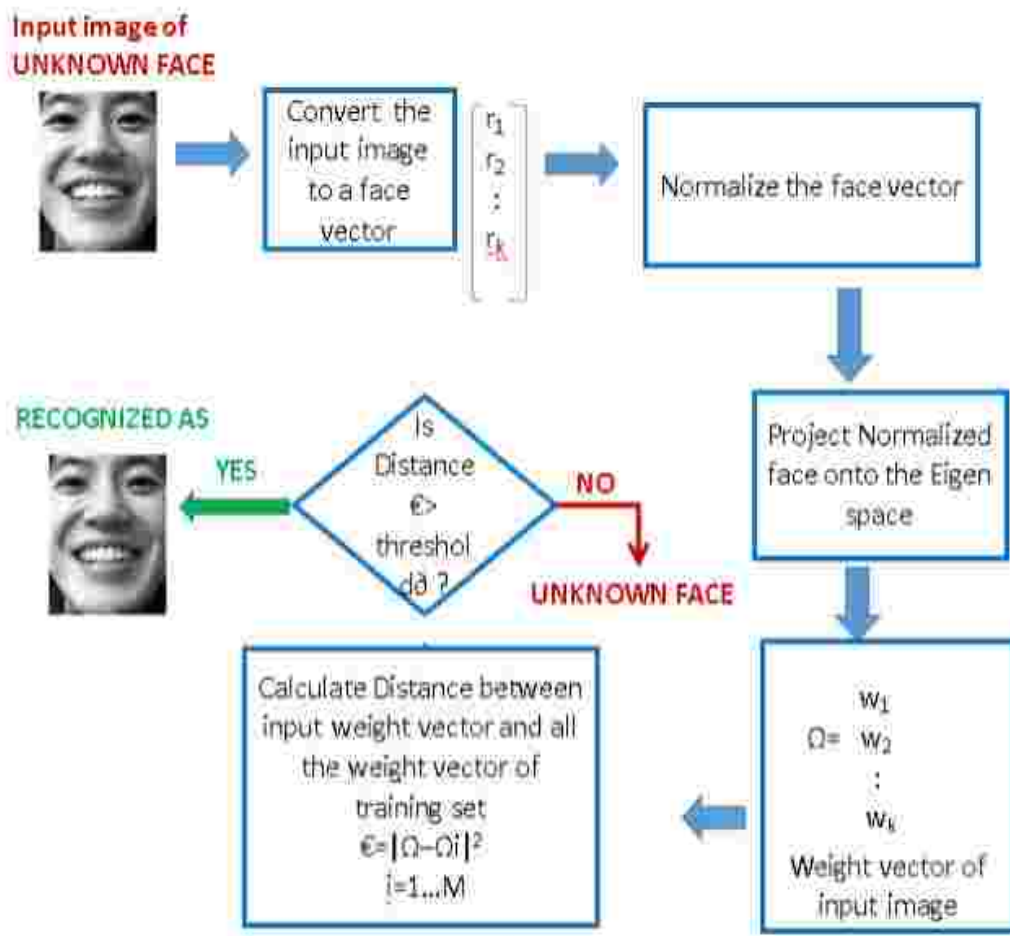


Figure 2. 18 The PCA based Eigenfaces face recognition algorithm flow chart

## A learning management system (LMS)

Is a software application for the administration, documentation, tracking, reporting and delivery of educational courses or training programs. They help the instructor deliver material to the students, administer tests and other assignments, track student progress, and manage record-keeping. LMSs are focused on online learning delivery but supporting a range of uses, acting as a platform for fully online courses, as well as several hybrid forms, such as blended learning and flipped classrooms. LMSs can be complemented by other learning technologies such as a training management system to manage instructor-led training or a Learning Record Store to store and track learning data.

## **Related work**

### **2.1.2 Class room attendance system using facial recognition system [10]**

This is a system that was implemented by Abhishek ha in ABES engineering college in Ghaziabad, the system takes the attendance by recognizing the faces on a given picture that the teacher would take of all students. For face detection LDA (liner Discriminate Analysis), which is an algorithm that detects faces based on skin color. This algorithm has many disadvantages such as the illumination and the lighting of the class. If the lighting of the class is too much, the algorithm cannot detect the faces on an image.

Also the direction of the face on the picture. For the recognition process the researcher used PCA. Finally, this system is not a real time application meaning that it only works on single picture and does not recognize faces from a live feed.

### **2.1.3 Automatic attendance system by using face recognition. [11]**

This system was certified by the international researcher journal of engineering and technology EIRJET.

The system is similar to the previous one with only a few differences like, this system used skin classification techniques for face detection which are techniques that define the skin color areas in a picture and black the rest of the picture.

The disadvantages of this system are:

- It can only detect a face from a limited distance.
- the system does not recognize properly in a poor lighting so may give false results.

### **2.1.4 Limitation**

The limitations and problems with the previously mentioned researches are listed below:

- 1) They are not real time application and only from pictures.
- 2) The algorithms take long execution times.

3) The accuracy of the algorithms is so low may produce false positives.

## **Proposed System**

The researcher propose a real time face recognition system that uses the Viola-Joins detection frame work for face detection. The Viola-Joins is haar-cascade-based, which is feature-based algorithm.

For face recognition the researchers are going to be using PCA-based-Eigenfaces algorithms. These tow algorithms were chosen based on the result of some researcher, conducted by other researcher, and the sources of the results are as follow:

For face detection, we relied on a paper by Kitili Jacson. For face recognition, we relied on a result of a comparative study conducted by Yin Shin Lin and Wain Seny Ny for National Taiwan University which concluded the PCA is the best recognition algorithm for a real time application.

Furthermore, system will interact with the LMS system to record the presence of student, so distributed system protocols are included in the system.

## **Chapter Summery**

Summary the searchers began the second chapter with an introduction then a background which talked about image processing and some other topics related to it like, steps and approaches. After that a glance of LMS is given. Followed by the algorithms that are used in the system. Finally, we previewed some related work and its disadvantages.

# **Chapter 3**

## **System Analysis**



# Introduction

In this chapter the system analysis will include the requirements of the system and also the requirements of the user both functional and nonfunctional. After that a use case diagram of the system is given along with its use case specifications. For the database are analyzed using Entry Relation Diagram (ERD)

## Requirements

In this part the researchers are going to give the requirements of both the user and the system. Each consisting of functional and nonfunctional requirements. The user requirements tend to be given by the user and include the main functions and operations the system is going to have or offer.

The system requirements are requirements the system include and they usually contain technical requirements for some user requirements and they are usually given by programmers and system developers.

### 3.1.1 User requirements

Here we list the main functional and nonfunctional requirements and they are as follow:

#### ❖ User functional requirements

1. The system should take the attendance of every student in the class.
2. The system should only take the attendance of the students that have registered in the course in the semester.
3. The attendance should be stored in the (LMS).
4. The administrator of the system should enter the information of the students into the system when start studying in the faculty.
5. The schedule should be entered by the administrator at the beginning of each semester containing the subject and the teacher and the hall for each class according to the specified table.

6. The administrator can change the schedule.
7. The teacher of a class can delay or delete a lecture .
8. The teacher of a class can change the hall in which the lecture is taken before the beginning of the lecture .

#### ❖ **User nonfunctional requirements**

1. The system should only work in the study hours.
2. The system should be connected to the university's network.
3. The system should be secure both physically and cyberly.

### **3.2.2 System Requirement**

#### ❖ **System functional requirements**

1. When the system identifies the faces on a frame it compares them with the faces of the students registered in that course and neglects the remaining.
2. The system can process feeds from several cameras at the same time.
3. The system should keep taking pictures from a camera until the end of class.
4. The system should check for new lectures every hour .
5. The system should send the attendance to the LMS at the end of each class.
6. The system is turn on and off manually by the system administrator or by the OS through a scheduled task .
7. The administrator can add or delete a lecture.
8. The system should keep the attendance in a database locally in case the LMS is down or unreachable.
9. Any new students should be entered into the system and their pictures should be entered too by the administrator.
10. The system and the LMS should communicate and sync data such as the student that are enrolled in a subject.

### ❖ System nonfunctional requirements

1. The system should process the frames fast.
2. The server has to be kept in a separate room.
3. The cameras and the wiring should be kept away from the reach of the students.
4. The web-based administration system should be easy to use .

## Use Case Diagram

The use case diagram is used to model the overall functions of the system and the roles of the actors.

Below is the use case diagram of our system as shown in Figure 3.1.

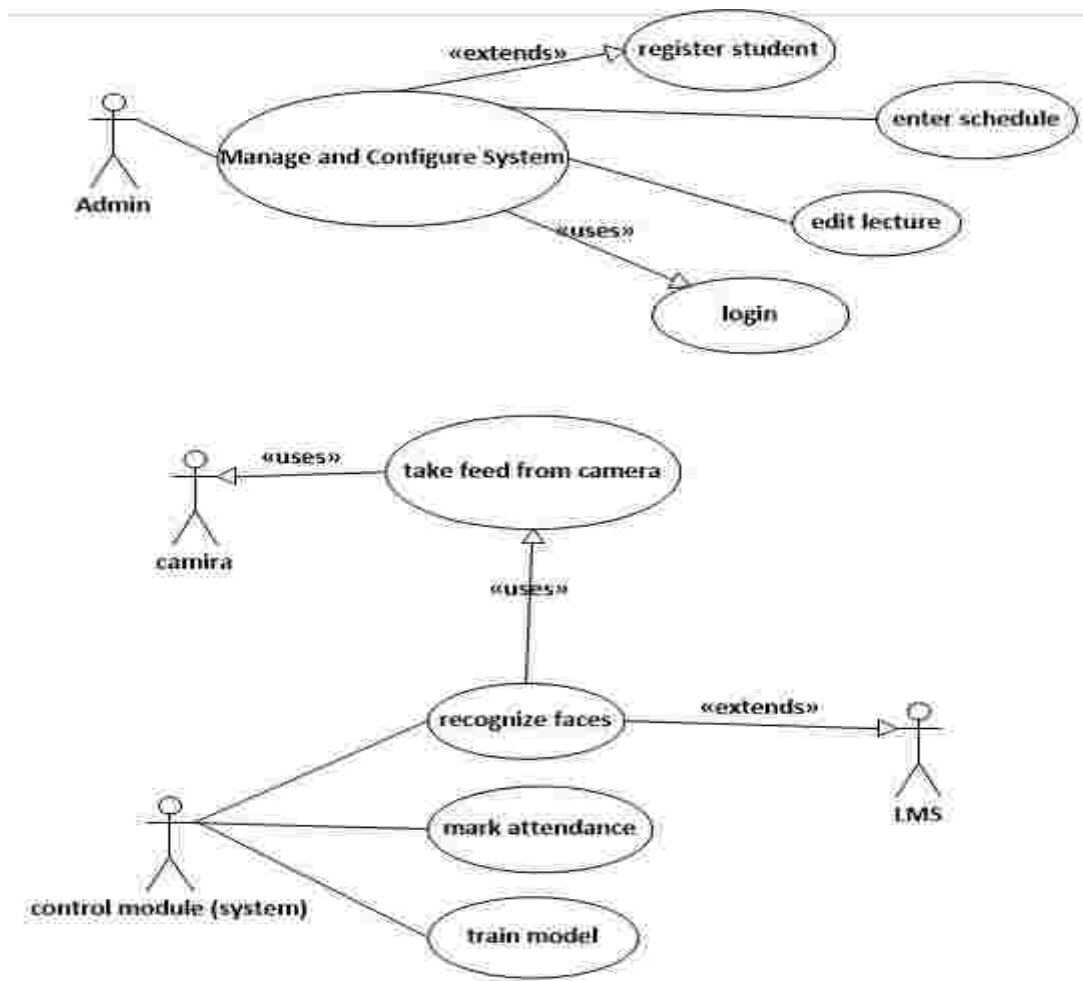


Figure 3. 1 Use Case Diagram

- **Use case entities description**

Use case entities description as shown in the tables ( Table 3.1 , Table 3.2 , Table 3.3 and Table 3.4 )

Table 3. 1 detect faces

Use case name	Detect faces
Primary actor	The control module of the system
Brief description	This use case is initiated by the system . the system starts capturing pictures from a camera. this use case runs the detection algorithm on the captured frames to detect, any faces on the frame.
Pre-condition	a camera is on and captures .
Normal flow of events	<ol style="list-style-type: none"> <li>1- The captured frames are passed to the detection Module for face detection.</li> <li>2- The frames are run on by the detection algorithm.</li> <li>3- The resulting faces are then kept in an array for recognition[E-1].</li> <li>4- This use case captures frames for the duration of the lecture</li> </ol>
Sub flow	
Alternative flow	
Post condition	
Exceptional flows	<p>[E-1 ] there are no faces on in feed.</p> <p>The system moves on.</p>

Table 3. 2 recognize faces

Use case name	recognize faces
Primary actor	The control module of the system
Brief description	This use case is initiated when detect faces use case finishes and it recognizes the faces of the students that are enrolled in the course.
Pre-condition	1- The detect faces use case is finished. 2- images are found in the frame.
Triggers	1- At the beginning of the class and students enrolled in the subject exist.
Normal flow of events	1- The control module takes the first image from the images that were given by the detected faces use case. 2- The module runs the face recognition algorithm on the image and it compares it to the images of the students enrolled in that course until it finds a match or it finishes all students with no match[E-1]. 3- When it finds a match it gets the student's label to identify the student. 4- The use case traverses through all images that were given to it. 5- A list of labels is given
Sub flow	
Alternative flow	
Post condition	
Exceptional flows	[E-1 ] when there are no hits in that database for an image it ignores the image and considers that student not enrolled in the course.

Table 3. 3 record attendance

Use case name	Record attendance
Primary actor	System control module.
Secondary actor	LMS. Local database
Brief description	This use case explains how the attendance of a lecture is stored locally and sent to CLMS .
Pre-condition	
Triggers	1-the time of the lecture is up.
Normal flow of events	1- The system stores the attendance in the local database . 2- The control module attempts to connect the lms database [E-1]. 3- The system queries any sessions for the that day [E-2]. 4- Then the attendance is marked for the subject for that day .
Sub flow	
Alternative flow	
Post condition	
Exceptional flows	[E-1 ] there is no connection with the LMS, the lecture attendance is marked as not taken in the local database and it has to be synced later.  [E-2 ] when a lecture exists and it does not have a session on clms the system moves on and the attendance is only marked on the local database.

Table 3. 4 manipulate information

Use case name	Manage and configure system.
Primary actor	Admin.
Brief description	In this use case the admin enters can add new students ,enter class lectures ,edit and delete lectures and, view reports .
Pre-condition	
Triggers	
Normal flow of events	<p>1- The admin can do the following operations:</p> <p>[A-1] add a new student.</p> <p>[A-2] alter a student's information.</p> <p>[A-3] add a schedule.</p> <p>[A-4] edit a lecture.</p> <p>[A-5] view class attendance.</p>
Alternative flow	<p>1- [A-1] add a new student the admin is prompted with a screen to add the student's info.</p> <p>2- [A-2] alter a student's info The admin can change the info of a student in the database.</p> <p>3- [A-3] add a schedule The admin is prompted with a screen to add new schedule with the necessary info such as the subject, first lecture date, starting time , hall and duration.</p> <p>4- [A-4] alter a schedule The admin can change the time of a lecture or delete a lecture or change its hall</p> <p>5- [A-5] view class attendance The admin can view the attendance of the students of a class</p>
Post condition	
Exceptional flows	

# Entity Relationship Diagram (ERD)

An ERD is a diagram that shows the relationships between the entities of a database and it is used to model and normalize a relational database.

For our system we will use a database to store the schedules of the lecture halls in the faculty and the attendance of student. Below is an ERD diagram of our system.

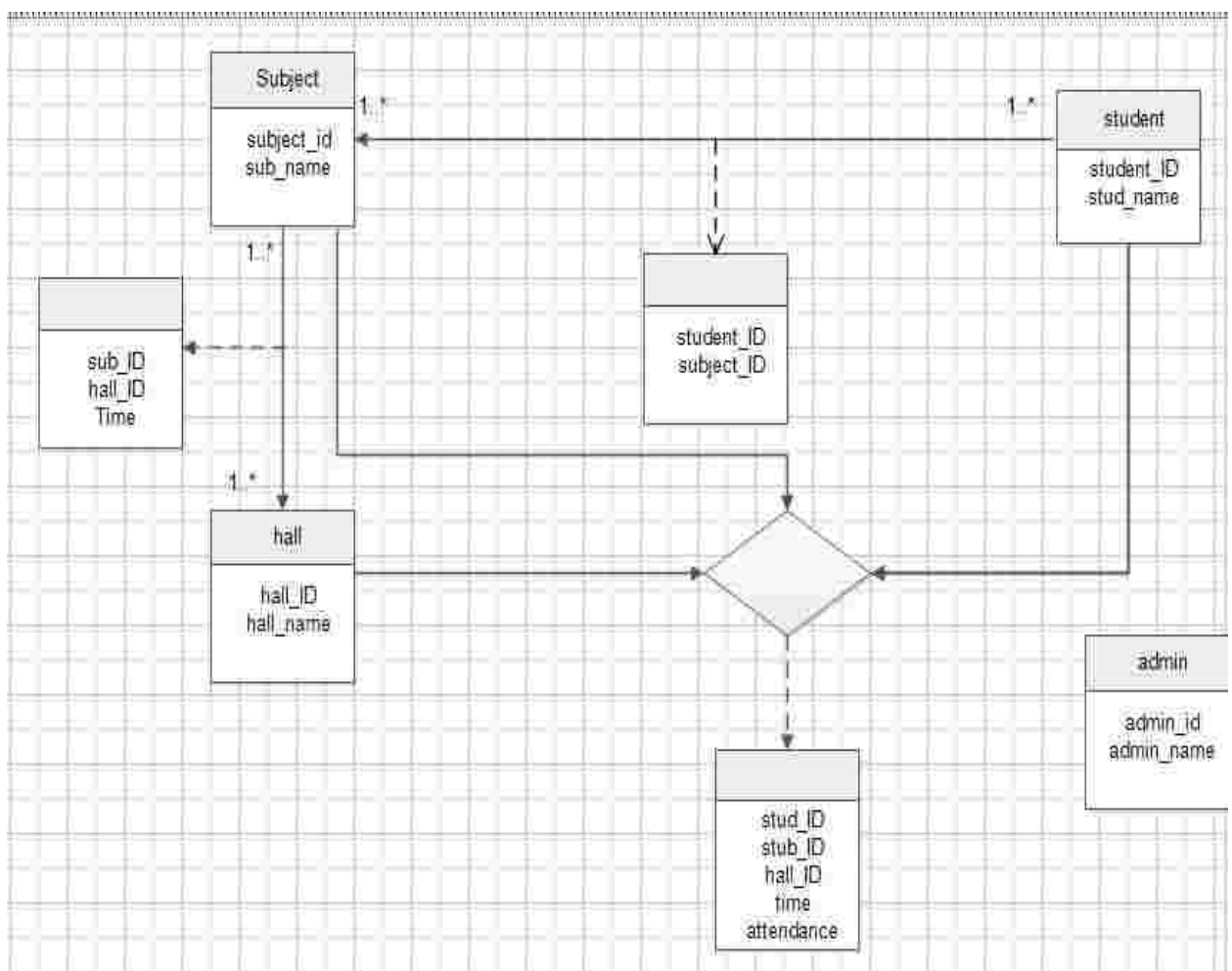


Figure 3. 2 ERD Diagram



# Class Diagram.

This project will take the object-oriented methodology and for this purpose we described the class diagram of our project as shown in figure Figure(3.3).

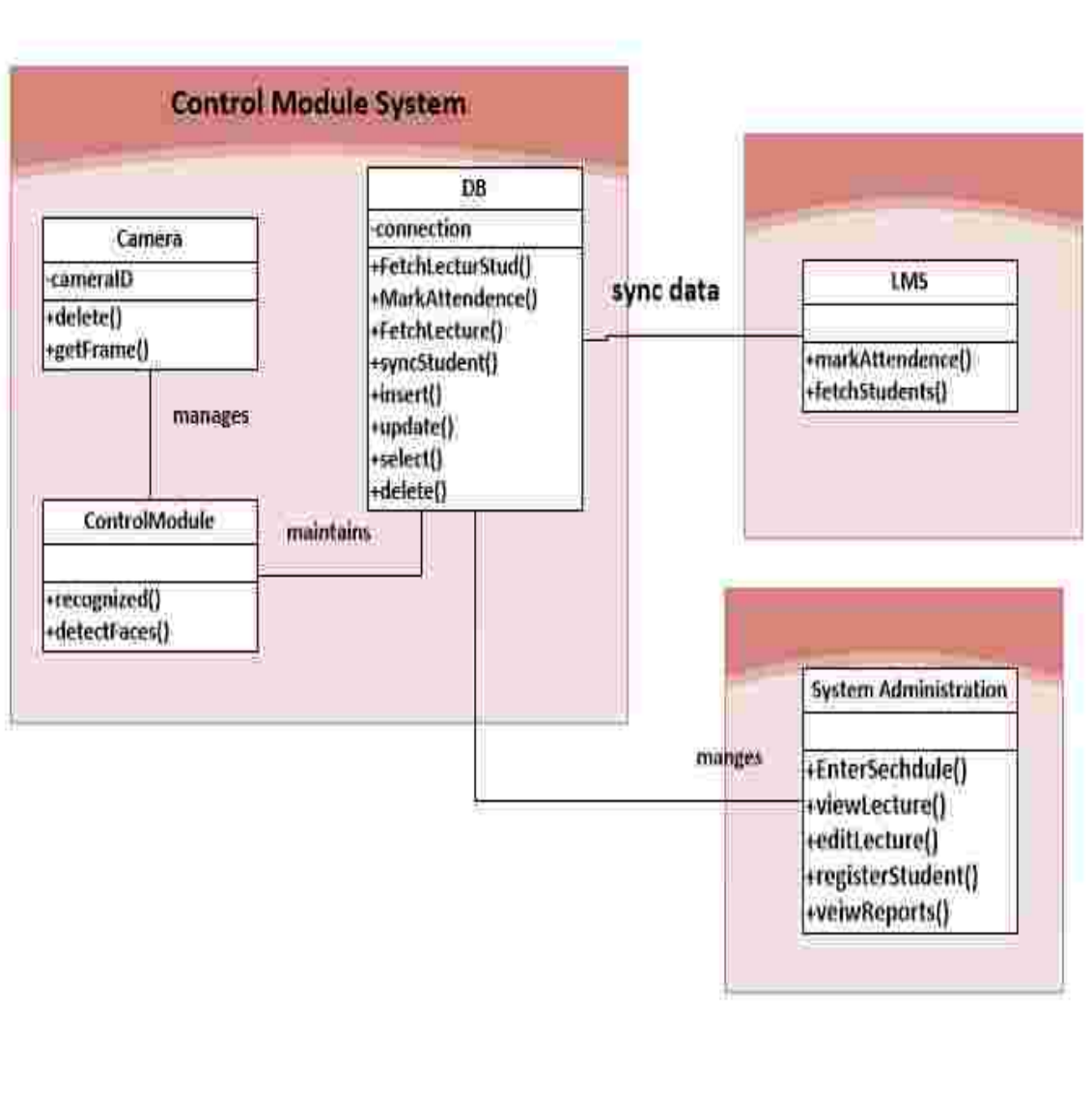


Figure 3. 3 class diagram

# **Chapter 4**

## **System design**

# Introduction

The fourth chapter comes after the analysis phase, we present the necessary design details of the diagrams of the system, first we introduce the architecture diagram of the overall the system, then the ERD of the local database followed by the class diagram. And finally the sequence diagram to show the sequence of the tasks taken by the system through time.

# Architectural Design

The style of the system architecture is the data centered architecture in the LMS, web based administration and main module components are connected to a data store and share data via the local and LMS databases. an illustration of the architecture of the system is shown in figure [4-1].

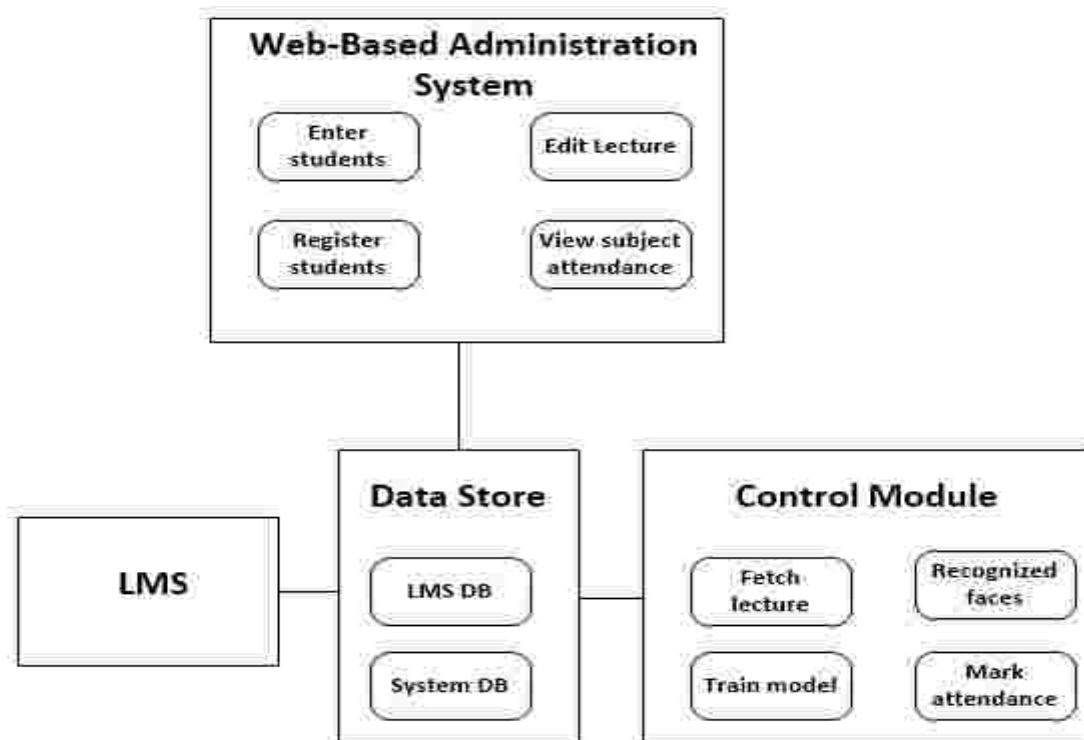


Figure 4. 1 Architectural Design

# Database design

The ERD was presented in the previous chapter. And in this chapter we presented the relational diagram of the local database as shown in figure [4-2].

## 4.3.1 Relational diagram

The Relational diagram of the ERD is shown in figure [4-2].

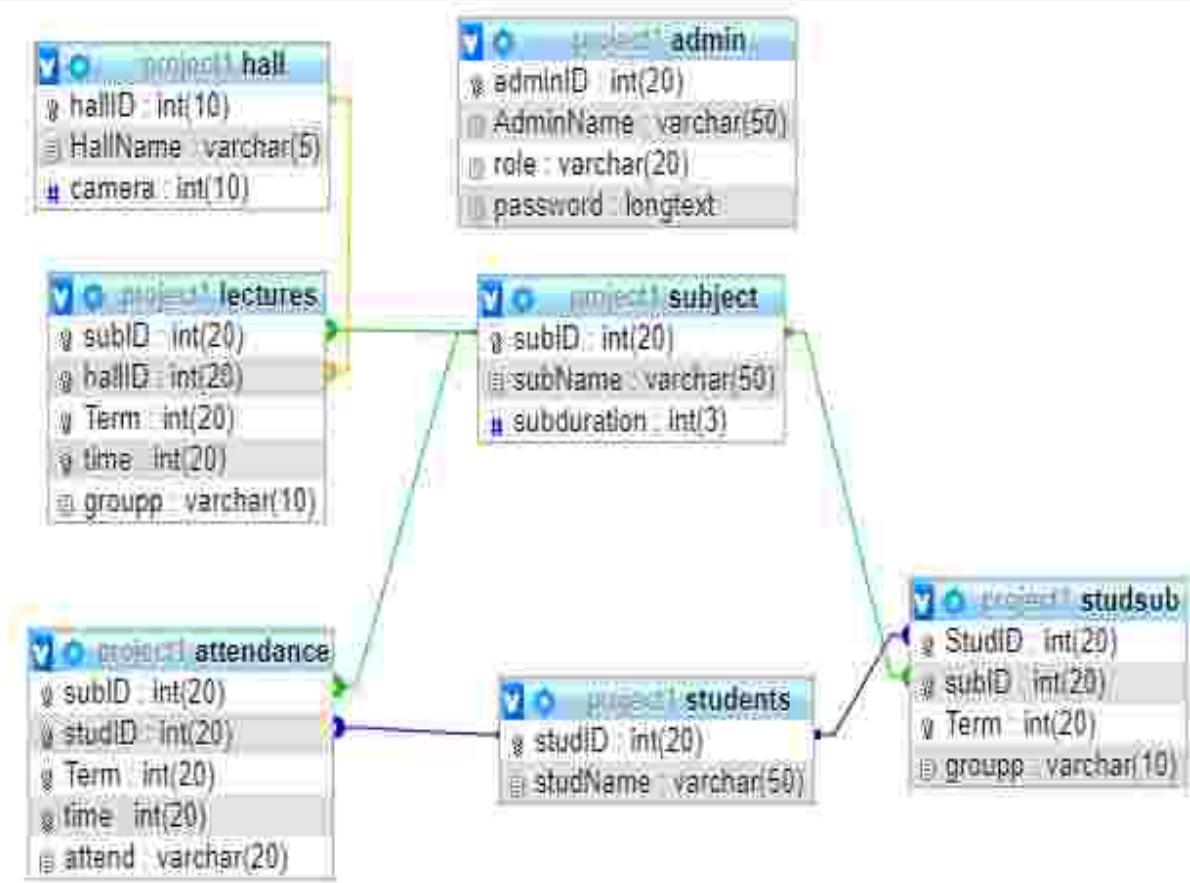


Figure 4. 2 Relational diagram

# Database Indexing

The tables of the database are illustrated in the tables [4-1], [4-2], [4-3], [4-4], [4-5], [4-6] and [4-7].

*Table 4. 1 admin*

<b>Name</b>	<b>Type</b>	<b>Length</b>
AdminID	Integer	10
AdminName	Varchar	20
Role	Varchar	7
Password	Longtext	

*Table 4. 2 students*

<b>Name</b>	<b>Type</b>	<b>Length</b>
StudID	Numeric	15
studName	Varchar	40

*Table 4. 3 subject*

<b>Name</b>	<b>Type</b>	<b>Length</b>
subID	Integer	5
subName	Varchar	5015
Subduration	Integer	1

Table 4. 4 lecture

<b>Name</b>	<b>Type</b>	<b>Length</b>
subID	Integer	15
hallID	Integer	2
Term	Integer	5
Time	Integer	5
Group	Varchar	10

Table 4. 5 studsub

<b>Name</b>	<b>Type</b>	<b>Length</b>
studID	Numeric	15
subID	Integer	5
Term	Integer	5
Group	Varchar	10

Table 4. 6 hall

<b>Name</b>	<b>Type</b>	<b>Length</b>
hallID	Integer	2
HallName	Varchar	10
Camera	Integer	2

Table 4. 7 attendance

Name	Type	Length
subID	Integer	5
hallID	Integer	2
Term	Integer	5
Time	Integer	5
Attend	Varchar	6

## Class Diagram

The next diagram is the class diagram which shows the object-oriented class design of the system as shown in figure [4-3].

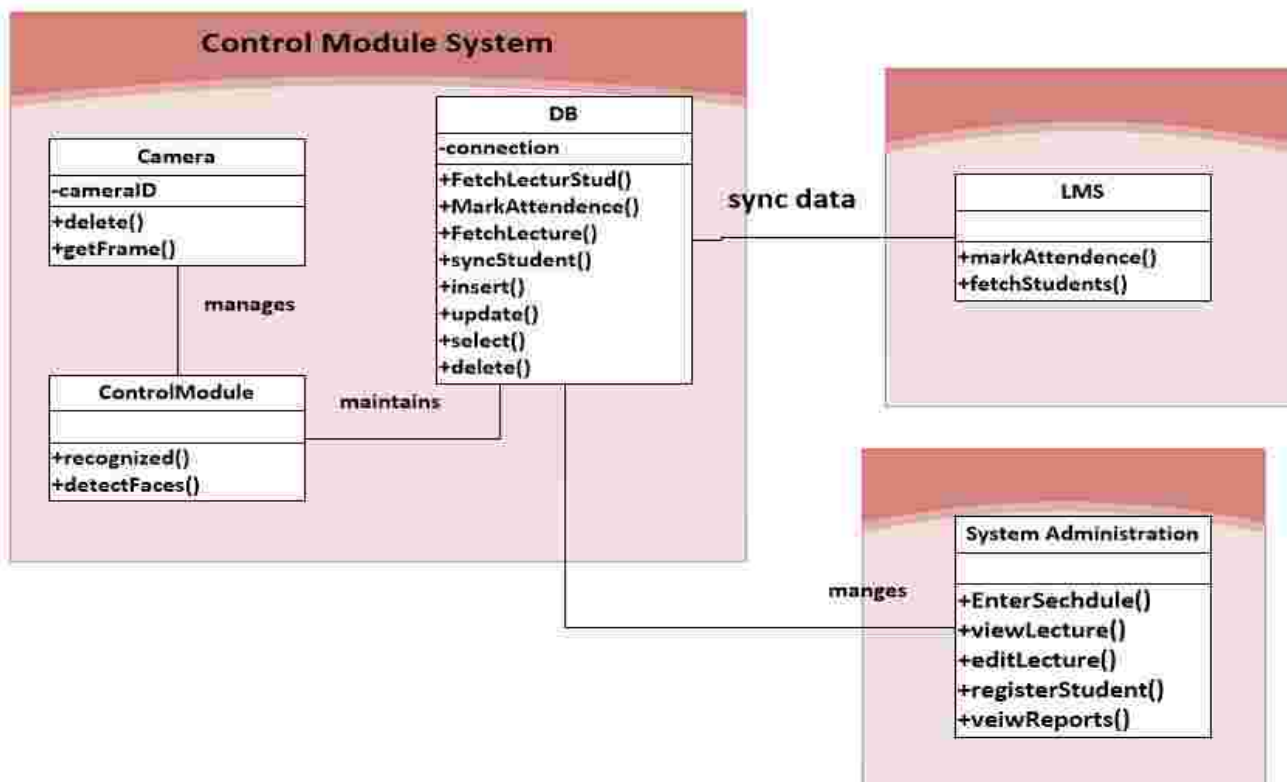


Figure 4. 3 class diagram

# Sequence diagram

The sequence diagrams of the main operations of the system are shown in figures [4-4] figures [4-5]

## 4.6.1 Recognize faces

This sequence diagram describes the flow of events of the recognized use case.

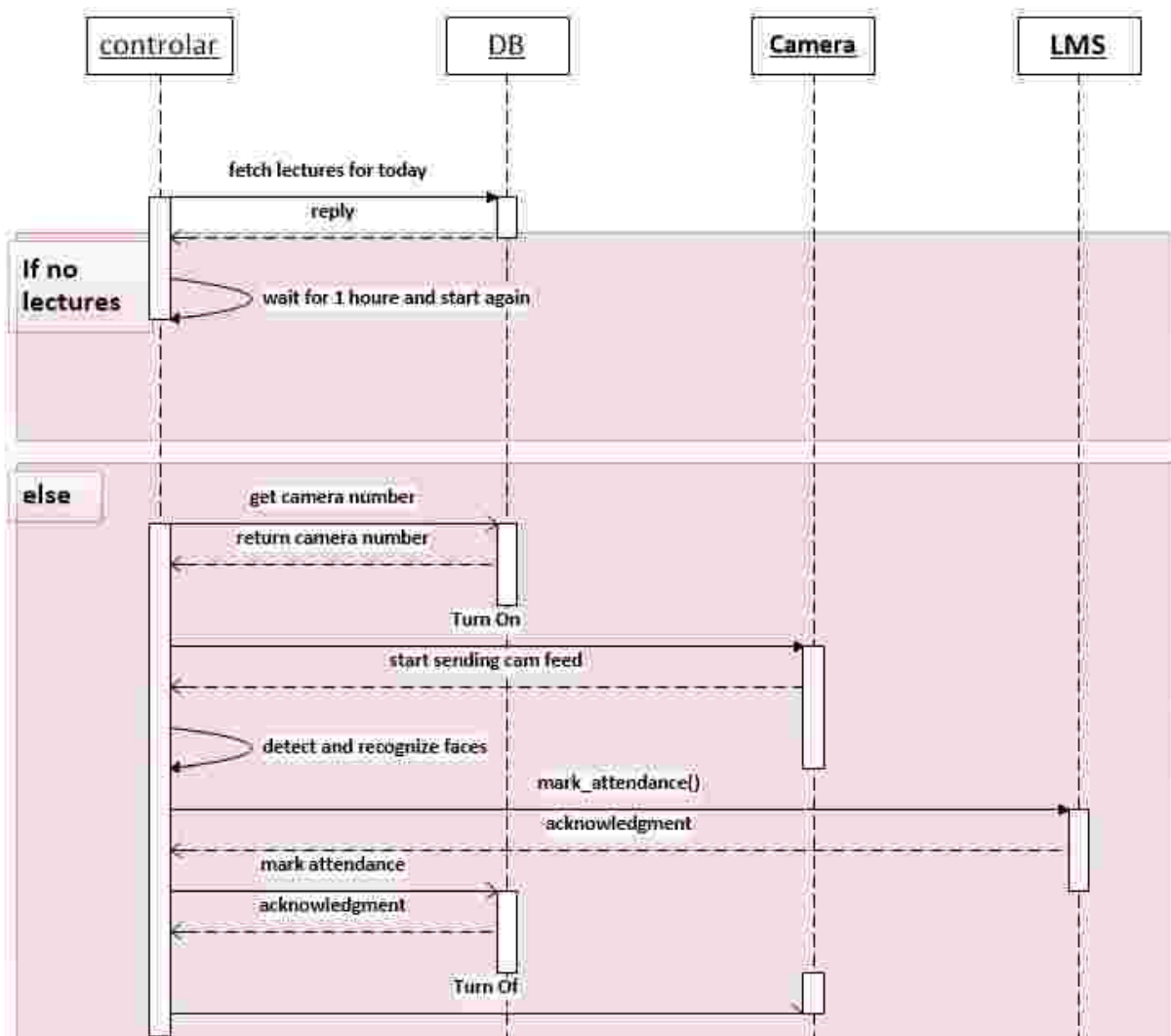


Figure 4. 4 sequence diagram (recognize faces)



## 4.6.2 Manage and Configure System

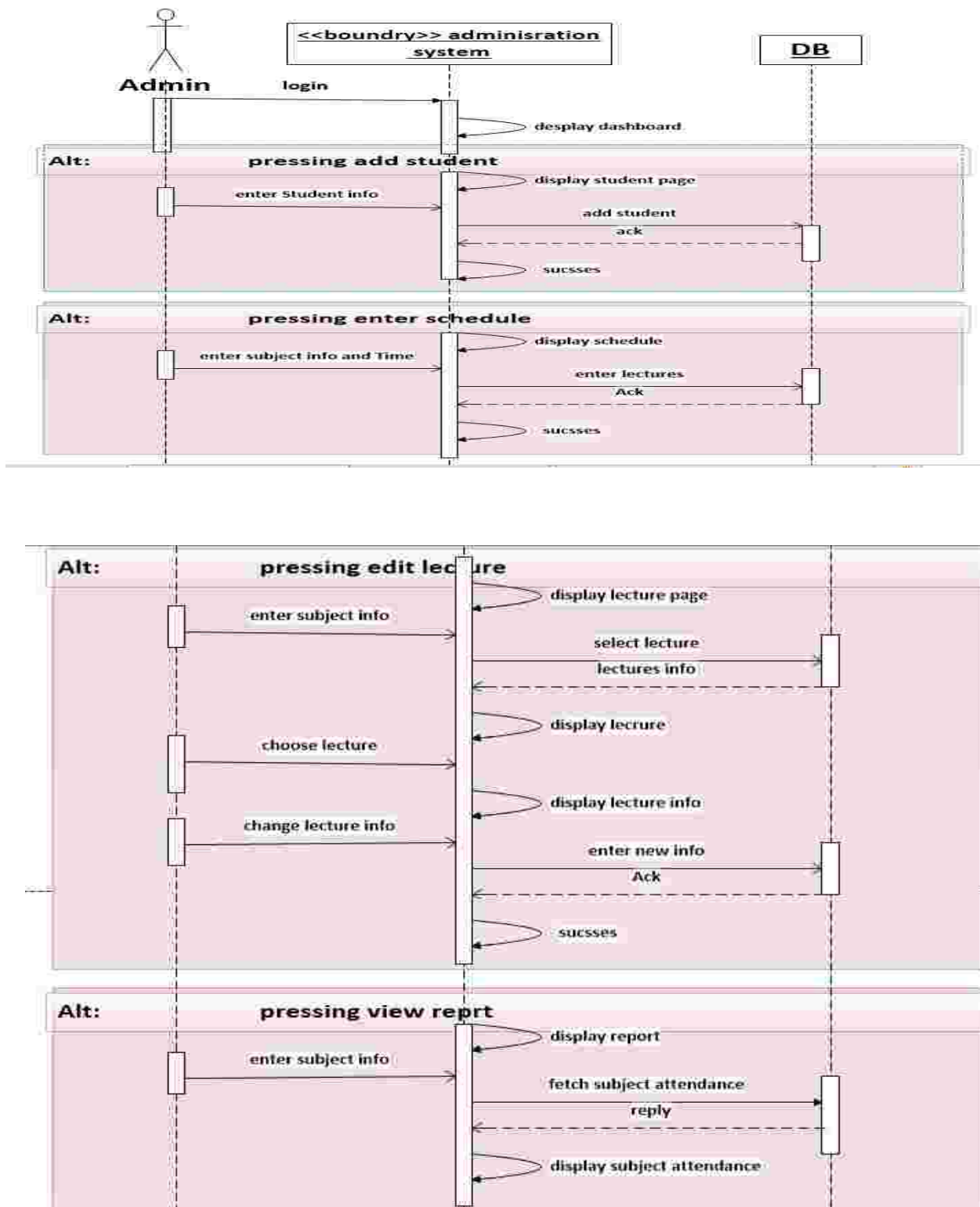


Figure 4. 5 sequence diagram (manage & configure system)

# System Interface

Finally comes the design of the interfaces of the administration system on figures [4-6]-[4-13].



Figure 4. 6 login interface

This screenshot shows the landing page of the system.

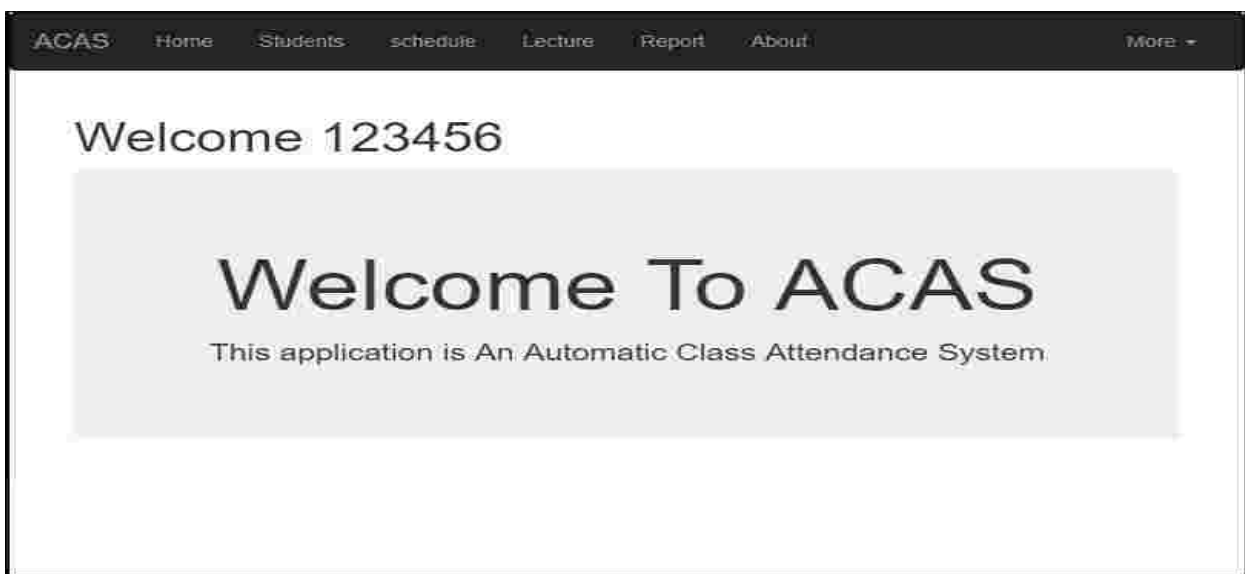


Figure 4. 7 Home interface

This screenshot shows the student page.

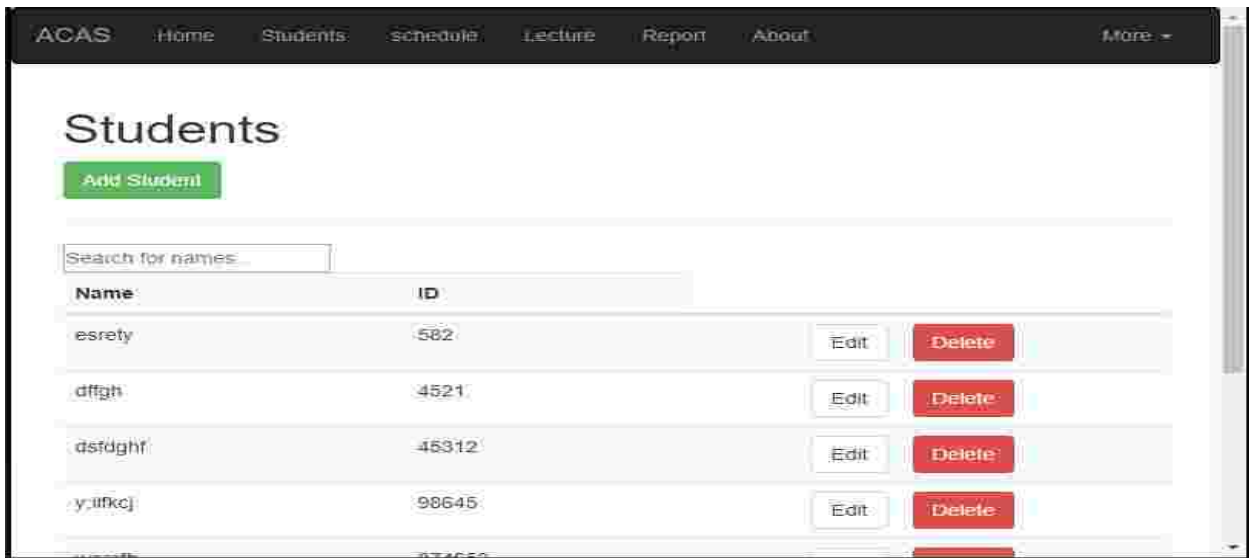


Figure 4. 8 students interface

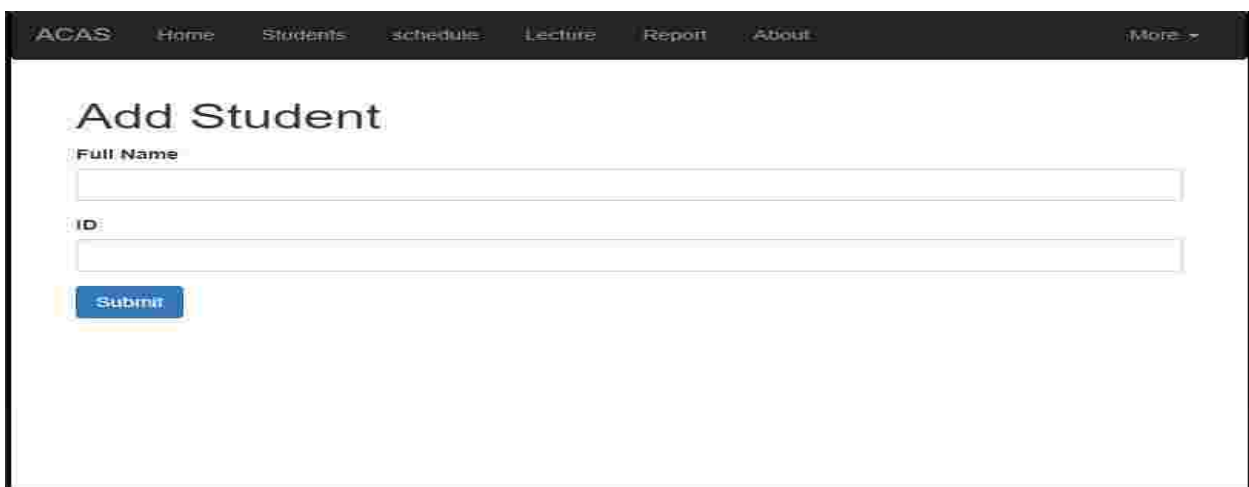


Figure 4. 9 add student interface

This page is the schedule page through which an admin can enter lecture data.

ACAS Home Students schedule Lecture Report About More

## schedule

Select subject (select one): network design

Select Hall (select one): S1

Select Time For First Lecture: 10:00-11:00

Select Starting Time: 8

Enter Number of Lectures: Enter a term

Enter Term (Hint: Like 20181): 20181

Enter Group:

Submit

Figure 4-10 schedule interface

ACAS Home Students schedule Lecture Report About More

## Lectures

Select Subject: Type Subject Name...

Enter Term: Type Term...

Enter Group: Type Group Name...

Submit

Figure 4-11 lectures interface

ACAS Home Students schedule Lecture Report About More

## Edit Lecture

Select Hall: 3-1 Select Day: yyyy-mm-dd Select Time: 8

Submit

Figure 4-12 edit lecture interface

Reports are shown in the report page.

ACAS Home Students schedule Lecture Report About More

## Report

Select Subject: Type Subject Name... Enter Term: Type Term... Enter Group: Type Group...

Submit

Figure 4-13 report interface

# **Chapter 5**

## **System implementation**

# Introduction

This system was designed to recognize student's faces and mark their attendance on LMS. Such a system definitely needs an administration portal to manage the inputs and view the outputs of the system. So in this chapter we first describe the main components of the system along with the needed software libraries and equipment. After that the working process is introduced to show how system performs its functions. Thorough test is made with screenshots.

## System components

There are several components comprising the system and they are as follow.

### **The automatic attendance component**

The core of the system is in this component which identifies the lectures and starts camera and make the attendance, For face detection and recognition we use openCV.

- **openCV**

Open computer vision is an open source python library which implements most computer vision algorithms and techniques uses the version of openCVwe used is 3.1.

- **frontal face classifier**

to detect a face the voila-joins algorithm needs to train on the object that needs to be detected after the algorithm is train on the object a classifier is saved and used every time the algorithm is started in our case we use pre-trained classifier called (frontal face .xml). this classifier describes the attributes of the human faces.

- **Face detection**

The function that implements the viola-joins algorithm is implemented like the following code

```
def detect(self, image, biggest_only=True):

    scale_factor = 1.2

    min_neighbors = 5

    min_size = (30, 30)

    biggest_only = True

    flags = cv2.CASCADE_FIND_BIGGEST_OBJECT | \

cv2.CASCADE_DO_ROUGH_SEARCH if biggest_only else \

    cv2.CASCADE_SCALE_IMAGE

    faces_coord = self.classifier.detectMultiScale(image,

        scaleFactor=scale_factor,

        minNeighbors=min_neighbors,

        minSize=min_size,

        flags=flags)

    return faces_coord
```

- **Face recognition**

Before a face is recognized the recognition algorithm needs to train a model on the images of the students. In the following code segment we show how a recognizer is created and how the train function is used.

```
images, labels, labels_dic = collect_dataset(students)
```



```
rec_lbphh = cv2.face.LBPHFaceRecognizer_create()
```

```
rec_lbphh.train(images, labels)
```

- **threads**

Through threading we were able to make the system recognize and take the attendance of several lectures at the same time. The idea is to create a thread for every lecture at the beginning of the class. The thread runs a thread function which does all the tasks starting with turning on the camera to sending the attendance to the LMS, the function that is executed as a thread is.

```
def recognition(web, window_name, end, students, lecture_info):
```

```
    images, labels, labels_dic = collect_dataset(students)
```

```
    rec_lbphh = cv2.face.LBPHFaceRecognizer_create()
```

```
    rec_lbphh.train(images, labels)
```

```
    print ("Models Trained Successfully")
```

## **working process**

The system should be started in the morning. As the system queries the database for any lectures for that day for that hour. If any lectures exist, the system connects to the LMS to synchronize the students that are enrolled in that subject and stores any new students in the database.

Then the system fetches the students enrolled in subject and train the model on those students. After that a thread is created to handle the lecture. This happens for every lecture. After that the system sleeps for an hour and start over. Secondly, and as the system is sleeping the running threads start by turning on the camera of the hall in which the subject is taught. Then the frames brought from the camera are scanned for faces using the detect function. After a face is detected on a frame the face is cut from the frame taken to the recognizer. The recognizer tries to recognize the face by looking into trained model. If a match is founded the label which is the ID of the student is stored in a list along with the time of his attendance. this process is repeated until the time of lecture finishes.

Furthermore, the attendance is taken at end of the lecture and compared with the list of the students enrolled in the subject to extract the absenteeism. After that the system connects to the LMS and sends the attendance to it. If there is a connectivity issue the system synchronizes the attendance at a later time.

A screenshot of the system while recognizing faces is shown in figure [5-1].

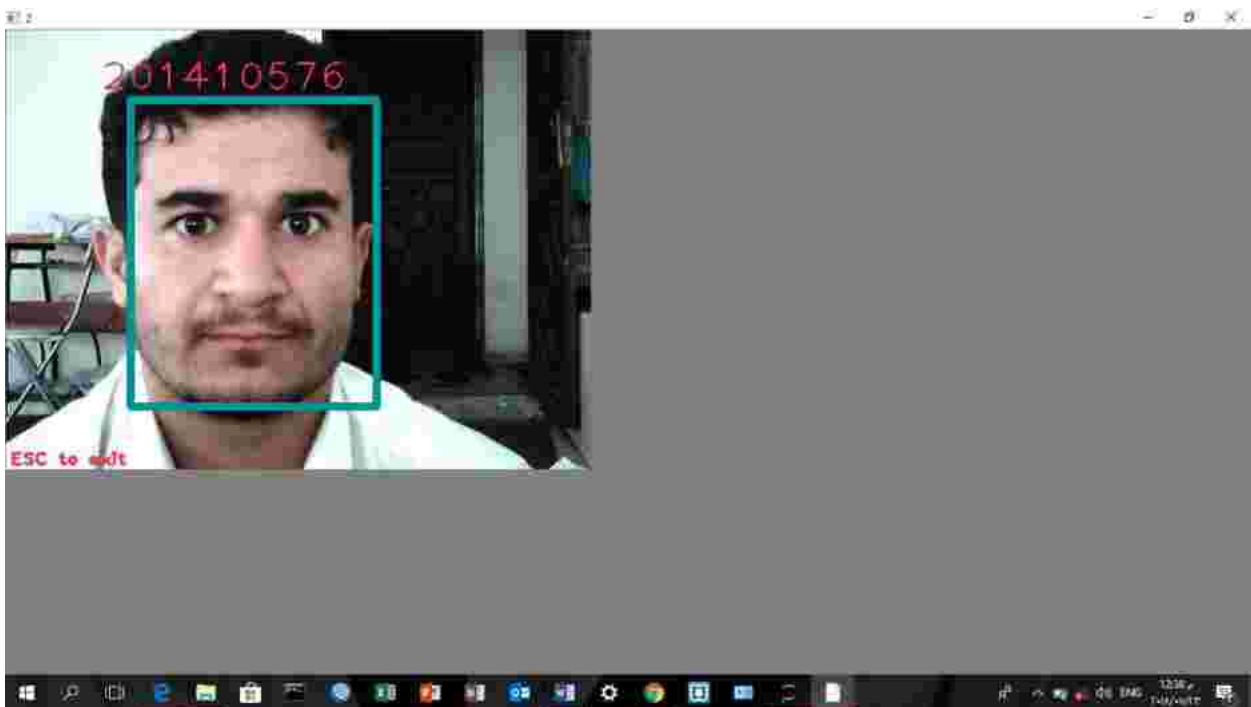


Figure 5. 1 Fares



Figure 5. 2 omer

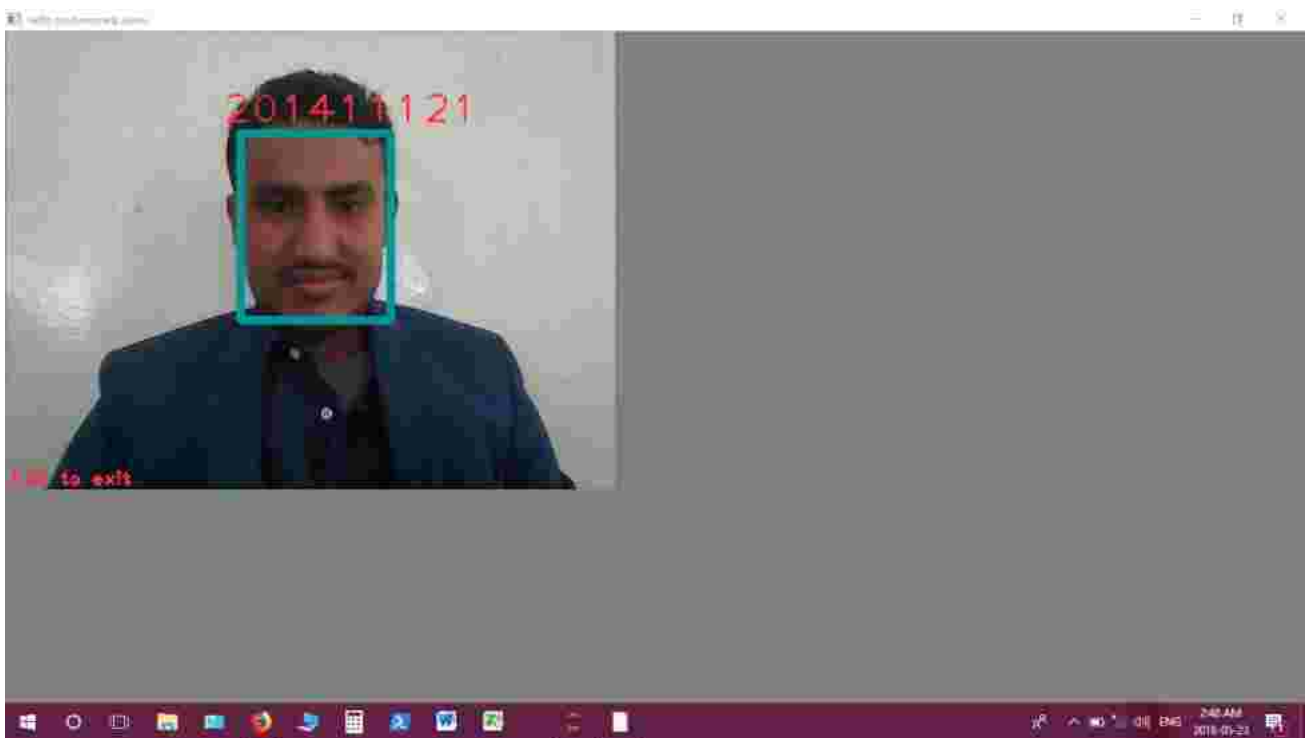


Figure 5. 3 mohammed

## web based administration

the second part of the system is the administration system which was designed to manage and configure the system through a web site built in python with flask.

- **Flask**

Flask is a micro web framework written in python and based on the working toolkit and DEJANGO template engine.

- **Tasks**

The main tasks the web portal provides are :

### 1- Add student

In this task the admin can add a new student to the system by entering the student ID number ,name and pictures for the recognition algorithm.

The pictures should be placed in the IMA folder before pressing the save button. When the student is added the system takes the student pictures and normalizes them so that algorithm would accept them when training the model. After the normalization process the new normalized pictures are placed in the people folder under the ID of the student

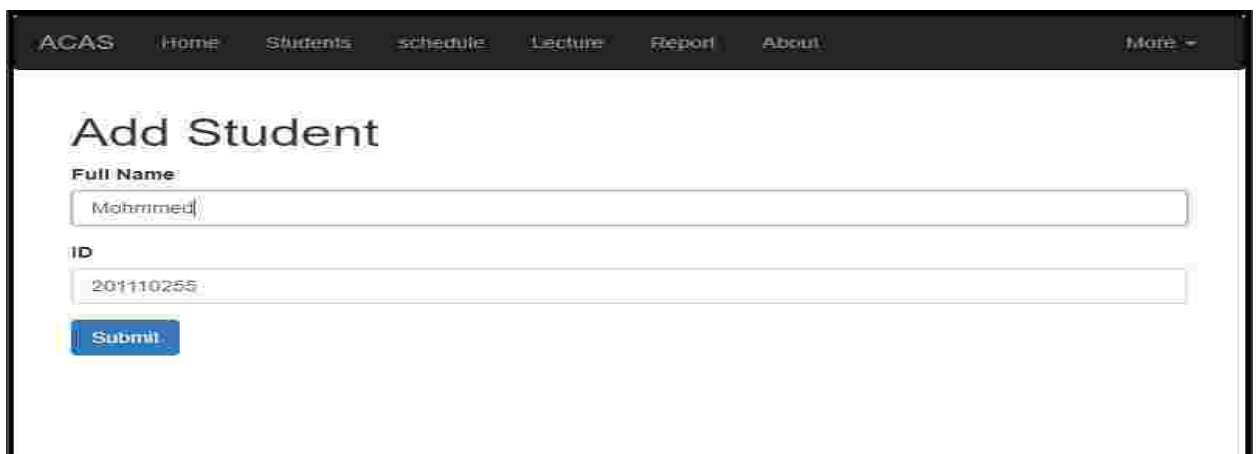


Figure 5. 4 add student interface

## 2- Add schedule

In this task enables the admin to enter the lectures of subjects at the beginning of each semester, images illustrating this task are shown in figure [5-3].

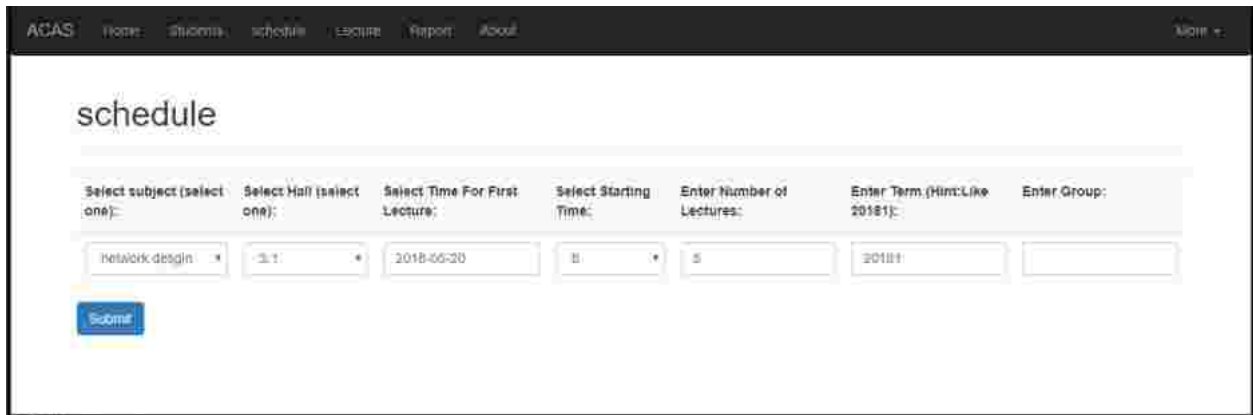


Figure 5. 5 schedule interface

## 3- Edit lecture

During the semester a teacher may want to delete a lecture or change its time or hall, this task enables the teacher to change the time or date or hall of lecture.

Screenshots are shown in figure [5-4].

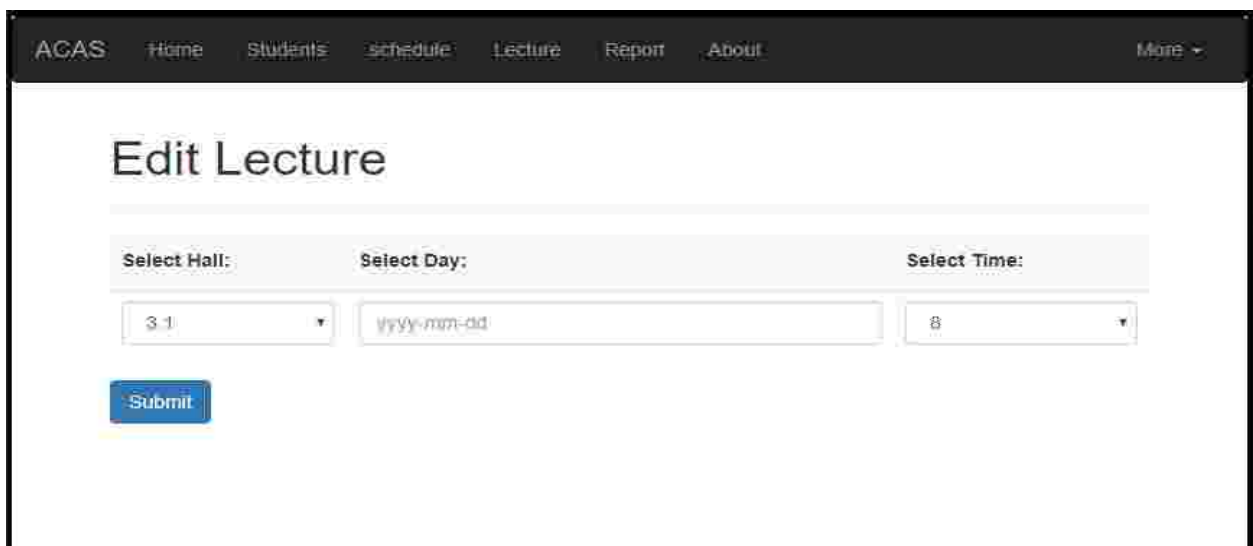


Figure 5. 6 edit lecture interface

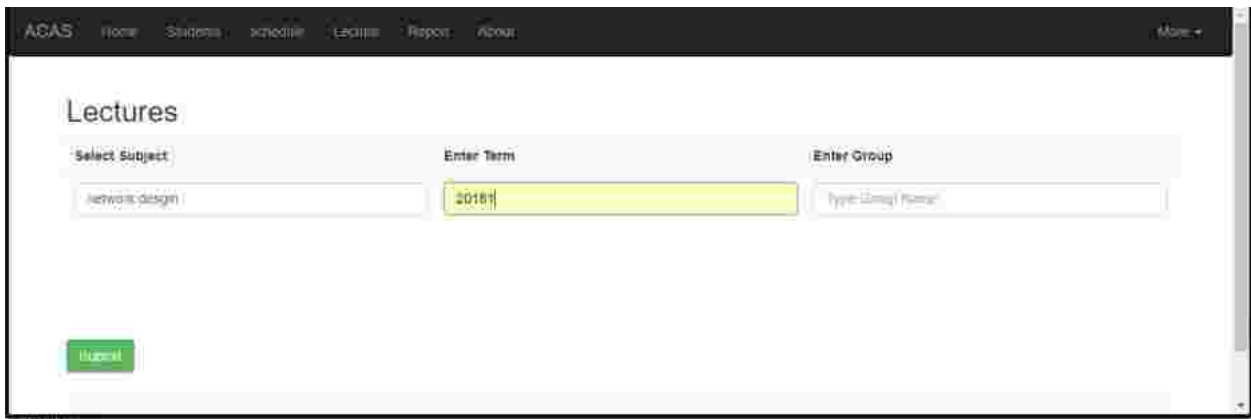


Figure 5. 7 edit lecture interface

#### 4- View reports

A teacher may want to view the attendance of subject, this task shows the students enrolled in a subject with their attendance through the semester as shown in figure [5-6].

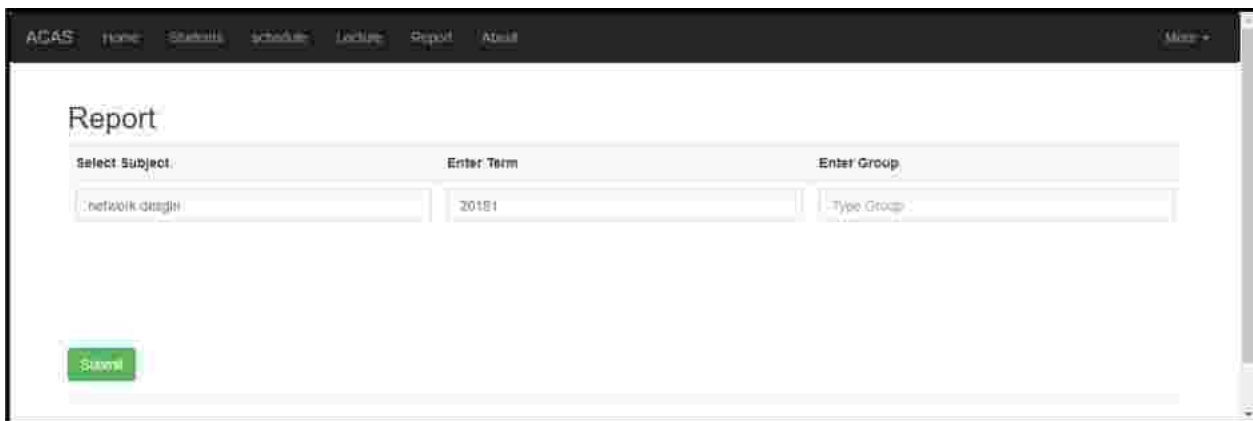


Figure 5. 8 report interface

### Report

Select Subject

Enter Term

Enter Group

Submit

Student Name	Lecture 1	Lecture 2	Lecture 3	Lecture 4	Lecture 5	Lecture 6	Lecture 7	Lecture 8	Lecture 9	Lecture 10	Lecture 11	Lecture 12	Lecture 13	Lecture 14	Lecture 15
name	absent														
name	late														
name	absent														

Figure 5. 9 report interface

### Students

Add student

Name	ID		
Sara obad	201410100	Edit	<span style="background-color: red; color: white; padding: 2px 5px;">Delete</span>
omid	201410574	Edit	<span style="background-color: red; color: white; padding: 2px 5px;">Delete</span>
Feres Alzobeh	201416679	Edit	<span style="background-color: red; color: white; padding: 2px 5px;">Delete</span>

Figure 5. 10 student report interface

# **Chapter 6**

## **Conclusion and Future Work**



# Introduction

In this chapter the researchers present a summary of the work done in this project. It also presents the recommendation and the future work that might be done to enhance and improve the project.

# Conclusion

Attendance is taking a daunting activity that waste teacher's time and consumes time from lectures.

ACAS is an automatic class attendance system that was primarily designed to automate the attendance process and stop everyone's intervention in the process. The system recognizes student's faces in their class rooms and marks their attendance in CLMS. The core of the system is the recognizer which implements a recognition algorithm and also the concurrent detection and recognition classes using threads.

This project provided a service by automating the attendance taking process, it included cool features such as marking the attendance in CLMS and providing portal for management. On the another hand the researchers found it helpful if the project used IP cameras instead of the USB cameras used now.

Also the project did have thorough reports such as per student reports and halls usage statics

Finally, any communication with the system is only through the web-based administration portal which was designed enable admins and teachers to enter necessary data such as the lectures classes and student's info.

## **Future work**

Any future work on the system should involve students in the web portal so students can view their attendance and interact with the system . Also it can provide teachers with the ability to set their own late-present policies for example a teacher may use the system to mark the students that are after 20 minutes from the start of the lecture as absent .

# References

- [1] <https://www.quora.com/What-is-image-processing-1>
- [2] <https://en.wikipedia.org>
- [3] N. Tom, *Face Detection, Near Infinity - Podcasts*, 2007.
- [4] T. Kanade, *Computer recognition of human faces*. Basel [etc.]: Birkhäuser, 1977.
- [5] A. L. Rekha and H. K. Chethan, “Automated Attendance System using face Recognition through Video Surveillance,” *Int. J. Technol. Res. Eng.*, vol. 1, no. 11, pp. 1327–1330, 2014.
- [6] ] I. Kim, J. H. Shim, and J. Yang, “Face detection,” *Face Detect. Proj. EE368 Stanf. Univ.*, vol. 28, 2003.
- [7] E. Shervin, “OpenCV Computer Vision,” 03-Oct-2010.
- [8] UNIVERSITY OF NAIROBI AUTOMATED ATTENDANCE MACHINE USING FACE DETECTION AND RECOGNITION  
PROJECT INDEX: PRJ 151  
BY NAME: KITILI JACKSON MWENDWA
- [9] [https://en.wikipedia.org/wiki/Learning\\_management\\_system](https://en.wikipedia.org/wiki/Learning_management_system).
- [10] N. Mahvish, “Face Detection and Recognition,” *Few Tutorials*, 2014. .
- [11] T. Matthew and A. Pentland, *Eigenfaces for Recognition*, vol. 3, Volume 3, Number 1 vols. Vision and Modelling Group, The Media Laboratory, MIT: *Journal of Cognitive Neuroscience*, 1991.
- [12] M. Fuzail, H. M. F. Nouman, M. O. Mushtaq, B. Raza, A. Tayyab, and M. W. Talib, “Face Detection System for Attendance of Class’ Students.”
- [13] P. Viola and M. J. Jones, “Robust real-time face detection,” *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.
- [14] Y.-Q. Wang, “An Analysis of the Viola-Jones Face Detection Algorithm,” *Image Process. Line*, vol. 4, pp. 128–148, Jun. 2014.
- [15] S. Z. Li and A. K. Jain, Eds., *Handbook of face recognition*. New York: Springer, 2005.