33

Facultyofcomputers&informatics

IT department

Studentabsence usingfacerecognition

Under Supervision of

Dr: Nabil Lashen

Facultyofcomputers&informatics

IT department

Studentabsence usingfacerecognition

Under Supervision of

Dr: Nabil Lashen

By

1- Mohamed Abdel Aziz Mohamed Radwan

2-Mahmoud Ahmed Mohamed Ahmed Elnaieb

3- Hanaa Kamal nabih

4-Asmaa Mansour mohamed

5-Hanan SalahElmohamdey

Acknowledgment

We begin by saying that we are merciful to help and assistance ***Allah*** gave us throughout our years of study in this faculty

We would like to thank and gratitude to all those who contributed to the work of this project whether it is within the scope of education (college) or outside. First, we thank our parents for motivating us to reach this degree of knowledge and thank them also for their contribution with us to achieve the highest degree of progress in this project. Secondly we would like to thank the supervisor of this project by Dr. Nabil lashen for his efforts with us throughout the duration of the study, whether in undergraduate courses or in the work of realizing the idea of this project.

Abstract

Our project is token about recording student absence using face_recognition in android studio software we combine android language with image processing

We recording student absence by first when student is first time he will register and registering his information and take a photo of himself after register he record his present in lecture the subject he has if it isn't the first time of his register he will login directly using his name and password that he has register at the first time in his information and take another photo to verify his personality

The other role is for professor as he also registers and login to calculate student absence if the percentage of absence is over limited, he will punish him.

The professor is assured that he is the real student through security

Student and professor android interface design

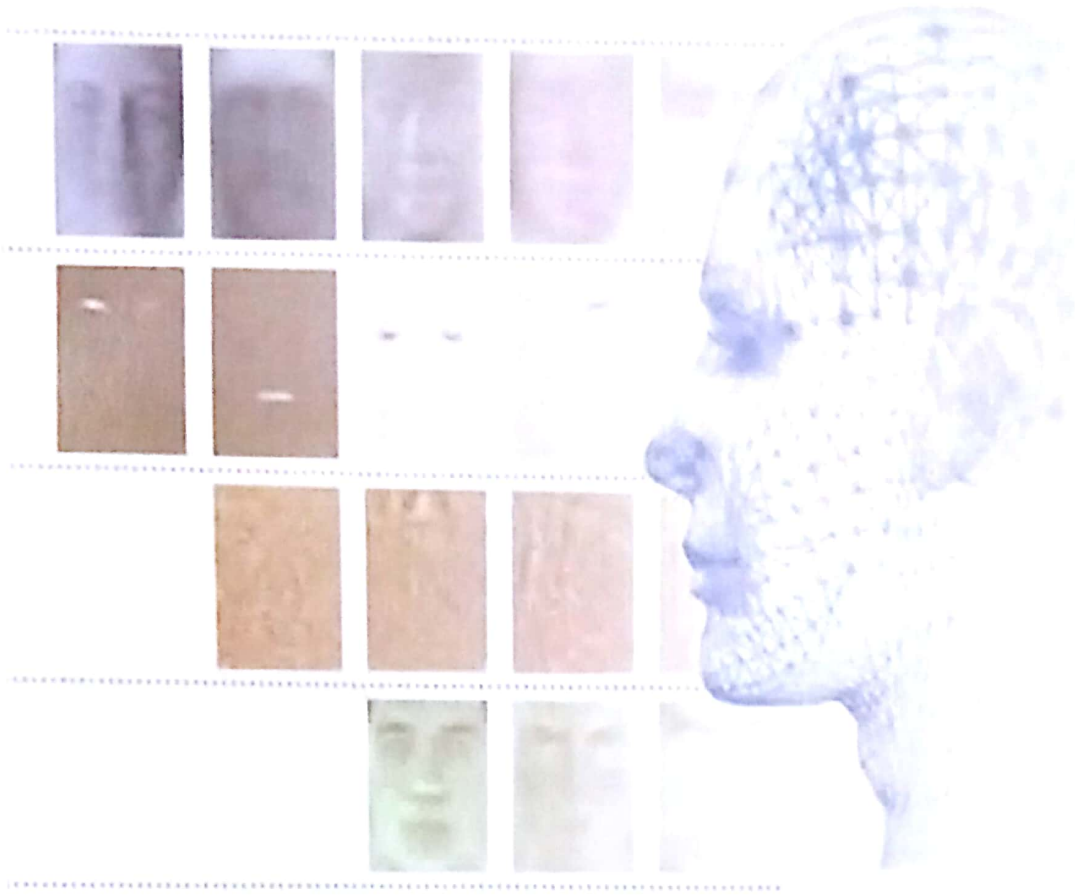
We verifying that student compare his image that he token before and stored in database

Subject also is stored in database

The partition of comparing image is using face recognition as we used Microsoft library and downloaded it in android studio

The security is done using wifi strength as we connect to college wifi to connect student with college and assured ha has his lectures

So we did that using gps that calculate wifi strength.



Contents

List of figures

Chapter One

Introduction and planning

1-1 :Problem statement

TheStudentsanddepartmentofuniversityareconcernedabouttheproblemofrecordingabsenteeism, sincethenumberofstudentsislargeandittakes a lot oftime

So, inthisproject, weworktoprovidestudentswiththemeantoregistertheirabsenceandprovide informationaboutthem.

Iteasierfor studentstoregistertheirabsencebytakephotoforstudentandrecognizingtheface.

1-2 : Theprojectgoals

Thisprojectoffersthefollowing services:

1- Students can register their absence.

2-Doctors can know the number of absences per student.

3-Provides information about students such as department, attendance and absence.

1- 3:toolsused

1- Android studio

is the official[7] integrated development environment (IDE) for the Android platform.

It was announced on May 16, 2013 at the Google I/O conference.

Android Studio was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014.

The first stable build was released in December 2014, starting from version 1.0

Based on JetBrains' IntelliJ IDEA software, Android Studio is designed specifically for Android development.

It is available for download on Windows, macOS and Linux,

and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.

2-Microsoft API

Microsoft Translator Text API, part of the Microsoft Cognitive Services API collection, is a cloud-based machine translation service supporting multiple languages that reach more than 95% of world's gross domestic product (GDP). Translator can be used to build applications, websites, tools, or any solution requiring multi-language support.
Built for business, Microsoft Translator is a proven, customizable, and scalable technology for machine translation. Microsoft Translator technology powers translation features across Microsoft products, including Office, Edge, SharePoint, Yammer, Visual Studio, Bing, and Skype. By simply integrating translation into web, desktop, or mobile applications, using industry standard REST technology, the Translator API provides a rich functionality set for any developer. Learn more about machine translation and how Microsoft Translator works.

3-my sql

is a tool to design, develop and administrate SQL database as a replacement for former MySQL GUI tools buddle. It provides comprehensive set of features including MySQL instance control, entity relationship diagrams, privileges control and reverse engineering of a current live database. Most importantly you can have all the above features under one hood un-like previous MySQL GUI tool bundle. And as always you can freely download this software without any cost regardless of the fact that MySQL tools are now developing by Oracle Corporation

4-php

is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994,

the PHP reference implementation is now produced by The PHP Development Team.

PHP originally stood for *Personal Home Page*,[4] but it now stands for the recursive acronym *PHP: Hypertext Preprocessor*.[5]

PHP code may be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks.

PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable.

The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page.

PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a *de facto* standard. Since 2014 work has gone on to create a formal PHP specification.

1-4 skills we gained from the project

- How to collect information about the problem.
- How to analyze information.
- How to select sufficient information.
- How to solve the problem.
- How to be a member of teamwork, and we learned the experience of co-operation.
- How to search about a specific information, tool, technique or technology that may help us in achieving our project by reading books, exploring the internet, or asking anyone.
-

Chapter 2

Face recognition and detection

2-1:Introduction

This project deals with the topic of face recognition techniques using digital image processing. Face recognition has always been a very challenging task for the researches. On the one hand, its applications may be very useful for personal verification and recognition. On the other hand, it has always been very difficult to implement due to all different situation that a human face can be found. Nevertheless, the approaches of the last decades have been determining for face recognition development. Due to the difficulty of the face recognition task, the number of techniques is large and diverse. In addition, the applications involve a huge number of situations.

Although we can find many other identification and verification techniques, the main motivation for face recognition is because it is considered a passive, no intrusive system to verify and identify people.

There are many other types of identification such as password, PIN (personal identification number) or token systems. Moreover, it is nowadays very instilled the usage of fingerprints and iris as a physiological identification system. They are very useful when we need an active identification system; the fact that a person has to expose their body to some device makes people feel being scanned and identified. The pause-and declare interaction is the best method for bank transactions and security areas; people feel conscious of it, as well as comfortable and safe with it. However, we do not want to interact with people that way in many other applications that required identification. For example, a store that wishes to recognize some customers or a house that has to identify people that live in there. For those application, face as well as voice verification are very desirable. It is also important that an identification technique is closer to the way human beings recognize each other.

As it has already said previously, the applications for face recognition are very varied. We can divide them into two big groups, the applications that required face identification and the ones that need face verification. The difference is that the first one uses a face to match with other one on a database; on the other hand, the verification technique tries to verify a human face from a given sample of that face.

Face recognition could be also divided into two different groups, according to their field of application. The main reason for promoting this technique is law enforcement application; however, it can also be used for commercial application. Among the law enforcement applications, some representative examples are

mug shot albums, video surveillance and shoplifting. Concerning commercial applications we can differentiate between entertainment (video games, virtual reality and training programs), smart cards (driver's license, passport and voter registration) and information security (TV parental control, cell phone and database security).

The new technologies continue to emerge with a variety of advanced features. It also affects the demand for self-recognition system. One of them is face recognition. Face recognition is a complex and difficult problem that is important for surveillance and security, telecommunications, digital libraries, and human-computer intelligent interactions

Face is our primary focus of interaction with the society. This can be observed in interaction among animals as well as between animals and humans. Face communicates identity, emotion, race, and age. It is also quite useful for judging gender, size and perhaps even character of the person. It has often been observed that human ability to recognize faces is quite remarkable. Faces are complex visual stimuli that are not easily described by shapes and patterns; yet people have the ability to recognize faces even after years of separation. Faces are so important in human interaction that no other avenue to person identification is as convincing as face recognition. There are several aspects of recognizing human identity and processing facial information that make the problem very vague. Recognition of a person's identity is not necessarily a function of viewing the person's face in isolation.

Face recognition is closely related to face detection, face tracking and facial expression analysis. The ability to recognize people by their facial characteristics is nothing but the Face recognition. Just as the human task of face recognition is neither clearly defined nor clearly differentiated from related tasks, automatic face recognition is not a single defined problem. We can define automatic face recognition as a pattern recognition task performed specifically on faces. It can be described as classifying a face as either "known" or "unknown", after comparing it with stored known faces.

Basically human face recognition procedure consists of two stages. The first stage is where the face detection process takes place very rapidly in humans except in certain circumstances where the object is located at a far distance. The second stage is recognition stage which is recognizing the face as individuals face

2-2: Importance of face recognition

In today's networked world, the need to maintain the security of information or physical property is becoming both increasingly important and increasingly difficult. From time to time we hear about the crimes like credit card frauds, network intrusions, or security breaches. The criminals are taking advantage of a fundamental flaw in the conventional access control systems. The systems do not grant access by "who we are", but by "What we have", such as ID cards, keys, passwords etc. None of these means actually define us. Rather, they merely are means to authenticate us. It goes without saying that if someone steals duplicates or acquires these identity means, he or she will be able to access our data or our personal property anytime and anywhere.

2-3: Face Recognition Methods

In the beginning of the 1970's, face recognition was treated as a 2D pattern recognition problem. The distances between important points where used to recognize known faces, e.g. measuring the distance between the eyes or other important points or measuring different angles of facial components. But it is necessary that the face recognition systems to be fully automatic. Face recognition is such a challenging yet interesting problem that it has attracted researchers who have different backgrounds: psychology, pattern recognition, neural networks, computer vision, and computer graphics.

The following methods are used to face recognition.

1. Holistic Matching Methods

2. Feature-based (structural) Methods

3. Hybrid Methods

1. Holistic Matching Methods:

In holistic approach, the complete face region is taken into account as input data into face catching system. One of the best example of holistic methods are Eigenfaces (most widely used method for face recognition), Principal Component Analysis, Linear Discriminant Analysis and independent component analysis etc.

- **Holistic example**

The first successful demonstration of machine recognition of faces was made by Turk and Pentland in 1991 using Eigenfaces. Their approach covers face recognition as a two-dimensional recognition problem. The flowchart in Figure 1 illustrates the different stages in an Eigen face based recognition system.
(1) The first stage is to insert a set of images into a database, these images are names as the training set and this is because they will be used when we compare images and when we create the Eigenfaces.
(2) The second stage is to create the Eigenfaces. Eigenfaces are made by extracting characteristic features from the faces. The input images are normalized to line up the eyes and mouths. They are then resized so that they have the same size. Eigenfaces can now be extracted from the image data by using a mathematical tool called Principal Component Analysis (PCA).
(3) When the Eigenfaces have been created, each image will be represented as a vector of weights.
(4) The system is now ready to accept entering queries.
(5) The weight of the incoming unknown image is found and then compared to the weights of those already in the system. If the input image's weight is over a given threshold it is considered to be unidentified. The identification of the input image is done by finding the image in the database whose weights are the closest to the weights of the input image. The image in the database with the closest weight will be returned as a hit to the user of the system

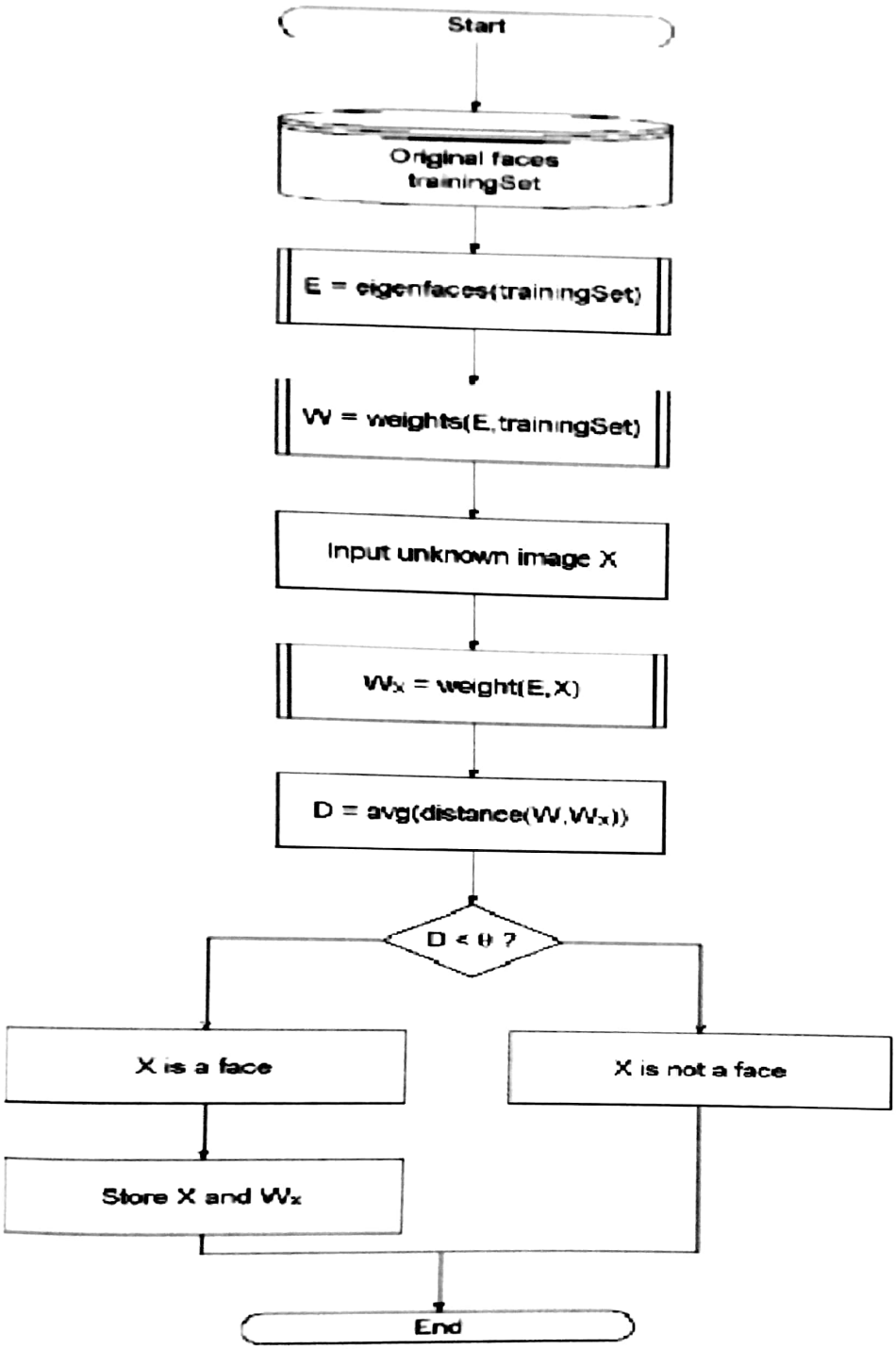


Figure 2-1: Flow chart of the Eigen face-based algorithm.

2. Feature-based (structural) Methods:

In this methods local features such as eyes, nose and mouth are first of all extracted and their locations and local statistics (geometric and/or appearance) are fed into a structural classifier. A big challenge for feature extraction methods is feature "restoration", this is when the system tries to retrieve features that are invisible due to large variations, e.g. head Pose when we are matching' a frontal image with a profile image.[5] Distinguishes between three different extraction methods:

I. Generic methods based on edges, lines, and curves

II. Feature-template-based methods

III. Structural matching methods that take into consideration geometrical Constraints on the features.

3. Hybrid Methods: Hybrid face recognition systems use a combination of both holistic and feature extraction methods. Generally 3D Images are used in hybrid methods. The image of a person's face is caught in 3D, allowing the system to note the curves of the eye sockets, for example, or the shapes of the chin or forehead. Even a face in profile would serve because the system uses depth, and an axis of measurement, which gives it enough information to construct a full face. The 3D system usually proceeds thus: Detection, Position, Measurement, Representation and Matching. **Detection** - Capturing a face either a scanning a photograph or photographing a person's face in real time.

Position - Determining the location, size and angle of the head. **Measurement** - Assigning measurements to each curve of the face to make a template with specific focus on the outside of the eye, the inside of the eye and the angle of the nose. **Representation** - Converting the template into a code - a numerical representation of the face and **Matching** - Comparing the received data with faces in the existing database. In Case the 3D image is to be compared with an existing 3D image, it needs to have no alterations. Typically, however, photos that are put in 2D, and in that case, the 3D image need a few changes. This is tricky, and is one of the biggest challenges in the field today.

✓ Eigen face

The problem with the image representation we are given is its high dimensionality. Two-dimensional $p \times q$ grayscale images span a $m = pq$-dimensional vector space, so an image with 100×100 pixels lies in a $10,000$-dimensional image space already. The question is: Are all dimensions equally useful for us? We can only make a decision if there's any variance in data, so what we are looking for are the components that account for most of the information. The Principal Component Analysis (PCA) was independently proposed by Karl Pearson (1901) and Harold Hoteling (1933) to turn a set of possibly correlated variables into a smaller set of uncorrelated variables. The idea is, that a high-dimensional dataset is often described by correlated variables and therefore only a few meaningful dimensions account for most of the information. The PCA method finds the directions with the greatest variance in the data, called principal components.

Algorithmic Description

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a random vector with observations $x_i \in \mathbb{R}^d$.

1. Compute the mean μ

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

2. Compute the the Covariance Matrix S

$$S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T,$$

3. Compute the eigenvalues λ_i and eigenvectors ν_i of S

$$S\nu_i = \lambda_i\nu_i, i = 1, 2, \ldots, n$$

4. Order the eigenvectors descending by their eigenvalue. The k principal components are the eigenvectors corresponding to the k largest eigenvalues.

The k principal components of the observed vector x are then given by:

$$y = W^T(x - \mu)$$

where $W = (\nu_1, \nu_2, \ldots, \nu_k)$.

The reconstruction from the PCA basis is given by:

$$x = Wy + \mu$$

where $W = (\nu_1, \nu_2, \ldots, \nu_k)$.

The Eigenfaces 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 neighbor between the projected training images and the projected query image.

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

$$X^TX\nu_i = \lambda_i\nu i$$

and get the original eigenvectors of $S = XX^T$ with a left multiplication of the data matrix:

$$XX^T(X\nu_i) = \lambda_i(X\nu_i)$$

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] for the derivation and proof of the equations.

✓ Fisherfaces

The Principal Component Analysis (PCA), which is the core of the Eigenfaces method, finds a linear combination of features that maximizes the total variance in data. While this is clearly a powerful way to represent data, it doesn't consider any classes and so a lot of discriminative information *may* be lost when throwing components away. Imagine a situation where the variance in your data is generated by an external source, let it be the light. The components identified by a PCA do not necessarily contain any discriminative information at all, so the projected samples are smeared together and a classification becomes impossible (see http://www.bytefish.de/wiki/pca_lda_with_gnu_octave for an example).

The Linear Discriminant Analysis performs a class-specific dimensionality reduction and was invented by the great statistician Sir R. A. Fisher. He successfully used it for classifying flowers in his 1936 paper *The use of multiple measurements in taxonomic problems* [Fisher36]. In order to find the combination of features that separates best between classes the Linear Discriminant Analysis maximizes the ratio of between-classes to within-classes scatter, instead of maximizing the overall scatter. The idea is simple: same classes should cluster tightly together, while different classes are as far away as possible from each other in the lower-dimensional representation. This was also recognized by Belhumeur, Hespanhaand Kriegman and so they applied a Discriminant Analysis to face recognition in [BHK97].

Algorithmic Description

Let $\mathbf{X}$ be a random vector with samples drawn from $\mathbf{c}$ classes:

$$\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_c\}$$
$$\mathbf{X}_i = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$$

The scatter matrices $\mathbf{S_B}$ and *S_{W}* are calculated as:

$$\mathbf{S_B} = \sum_{i=1}^{c} \mathbf{N}_i (\mu_i - \mu)(\mu_i - \mu)^T$$

$$\mathbf{S_W} = \sum_{i=1}^{c} \sum_{\mathbf{x}_j \in \mathbf{X}_i} (\mathbf{x}_j - \mu_i)(\mathbf{x}_j - \mu_i)^T$$

, where μ is the total mean:

$$\mu = \frac{1}{\mathbf{N}} \sum_{i=1}^{\mathbf{N}} \mathbf{x}_i$$

And μ_i is the mean of class $\mathbf{i} \in \{\mathbf{1}, \ldots, \mathbf{c}\}$:

$$\mu_i = \frac{1}{|X_i|} \sum_{x_j \in X_i} x_j$$

Fisher's classic algorithm now looks for a projection W, that maximizes the class separability criterion:

$$W_{opt} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|}$$

Following [BHK97], a solution for this optimization problem is given by solving the General Eigenvalue Problem:

$$\begin{aligned} S_B v_i &= \lambda_i S_w v_i \\ S_W^{-1} S_B v_i &= \lambda_i v_i \end{aligned}$$

There's one problem left to solve: The rank of S_W is at most $(N - c)$, with N samples and c classes. In pattern recognition problems the number of samples N is almost always samller than the dimension of the input data (the number of pixels), so the scatter matrix S_W becomes singular (see [RJ91]). In [BHK97] this was solved by performing a Principal Component Analysis on the data and projecting the samples into the $(N - c)$-dimensional space. A Linear Discriminant Analysis was then performed on the reduced data, because S_W isn't singular anymore.

The optimization problem can then be rewritten as:

$$\begin{aligned} W_{pca} &= \arg\max_W |W^T S_T W| \\ W_{fld} &= \arg\max_W \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|} \end{aligned}$$

The transformation matrix W, that projects a sample into the $(c - 1)$-dimensional space is then given by:

$$W = W_{fld}^T W_{pca}^T$$

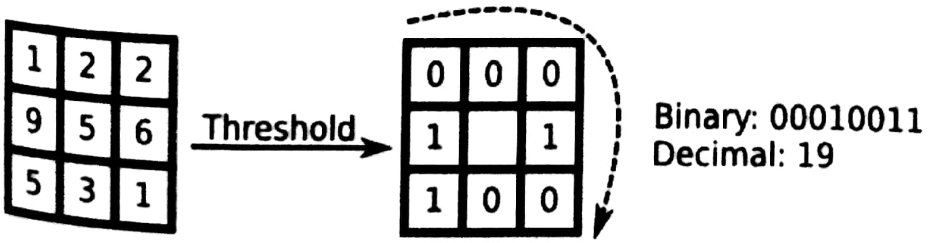
✓ local binary

Eigenfaces and Fisherfaces take a somewhat holistic approach to face recognition. You treat your data as a vector somewhere in a high-dimensional image space. We all know high-dimensionality is bad, so a lower-dimensional subspace is identified, where (probably) useful information is preserved. The Eigenfaces approach maximizes the total scatter, which can lead to problems if the variance is generated by an external source, because components with a maximum variance over all classes aren't necessarily useful for classification (see http://www.bytefish.de/wiki/pca_lda_with_gnu_octave). So to preserve some discriminative information we applied a Linear Discriminant Analysis and optimized as described in the Fisherfaces method. The Fisherfaces method worked great... at least for the constrained scenario we've assumed in our model.

Now real life isn't perfect. You simply can't guarantee perfect light settings in your images or 10 different images of a person. So what if there's only one image for each person? Our covariance estimates for the subspace *may* be horribly wrong, so will the recognition. Remember the Eigenfaces method had a 96% recognition rate on the AT&T Facedatabase? How many images do we actually need to get such useful estimates? Here are the Rank-1 recognition rates of the Eigenfaces and Fisherfaces method on the AT&T Facedatabase

So in order to get good recognition rates you'll need at least 8(+-1) images for each person and the Fisherfaces method doesn't really help here. The above experiment is a 10-fold cross validated result carried out with the facerec framework at: https://github.com/bytefish/facerec. This is not a publication, so I won't back these figures with a deep mathematical analysis. Please have a look into [KM01] for a detailed analysis of both methods, when it comes to small training datasets.

So some research concentrated on extracting local features from images. The idea is to not look at the whole image as a high-dimensional vector, but describe only local features of an object. The features you extract this way will have a low-dimensionality implicitly. A fine idea! But you'll soon observe the image representation we are given doesn't only suffer from illumination variations. Think of things like scale, translation or rotation in images - your local description has to be at least a bit robust against those things. Just like SIFT, the Local Binary Patterns methodology has its roots in 2D texture analysis. The basic idea of Local Binary Patterns is to summarize the local structure in an image by comparing each pixel with its neighborhood. Take a pixel as center and threshold its neighbors against. If the intensity of the center pixel is greater-equal its neighbor, then denote it with 1 and 0 if not. You'll end up with a binary number for each pixel, just like 11001111. So with 8 surrounding pixels you'll end up with 2^8 possible combinations, called *Local Binary Patterns* or sometimes referred to as *LBP codes*. The first LBP operator described in literature actually used a fixed 3 x 3 neighborhood just like this:



2-4: Face Recognition Applications

Face recognition is also useful in human computer interaction, virtual reality, database recovery, multimedia, computer entertainment, information security e.g. operating system, medical records, online banking , Biometric e.g. Personal Identification - Passports, driver licenses , Automated identity verification - border controls , Law enforcement e.g. video surveillances , investigation , Personal Security - driver monitoring system, home video surveillance system.

Face Identification: Face recognition systems identify people by their face images. Face recognition systems establish the presence of an authorized person rather than just checking whether a valid identification (ID) or key is being used or whether the user knows the secret personal identification numbers (Pins) or passwords. The following are example. To eliminate duplicates in a nationwide voter registration system because there are cases where the same person was assigned more than one identification number. The face recognition system directly compares the face images of the voters and does not use ID numbers to differentiate one from the others. When the top two matched faces are highly similar to the query face image, manual review is required to make sure they are indeed different persons so as to eliminate duplicates.

Access Control: In many of the access control applications, such as office access or computer logon, the size of the group of people that need to be recognized is relatively small. The face pictures are also caught under natural conditions, such as frontal faces and indoor illumination. The face recognition system of this application can achieve high accuracy without much co-operation from user. The following are the example. Face recognition technology is used to monitor continuously who is in front of a computer terminal. It allows the user to leave the terminal without closing files and logging out. When the user leaves for a predetermined time, a screen saver covers up the work and disables the mouse & keyboard. When the user comes back and is recognized, the screen saver clears and the previous session appears as it was left. Any other user who tries to logon without authorization is denied.

Image database investigations: Searching image databases of licensed drivers benefit recipients, missing children, immigrants and police bookings.

General identity verification: Electoral registration, banking, electronic commerce, identifying newborns, national IDs, passports, employee IDs.

Security: Today more than ever, security is a primary concern at airports and for airline staff office and passengers. Airport protection systems that use face recognition technology have been implemented at many airports around the world.

Surveillance: Like security applications in public places, surveillance by face recognition systems has a low user satisfaction level, if not lower. Free lighting conditions, face orientations and other divisors all make the deployment of face recognition systems for large scale surveillance a challenging task. The following are some example of face based surveillance. To enhance town center surveillance in Newham Borough of London, this has 300 cameras linked to the closed circuit TV (CCTV) controller room. The city council claims that the technology has helped to achieve a 34% drop in crime since its facility. Similar systems are in place in Birmingham, England. In 1999 Visions was awarded a contract from National Institute of Justice to develop smart CCTV technology.

2-5: Generic framework

Face recognition algorithm can be divided into the following two functional modules:-

Face image detector finds the locations of human faces in a normal picture, and a Face recognizer determines who the person is. Both the face detector and the face recognizer follow the same framework. They have a feature extractor that transforms the pixels of the facial image into a useful vector representation, and a pattern recognizer that searches the database to find the best match of the input image. In the face detection phase, the pattern recognizer categorizes the incoming feature vector according to one of the two image classes: "face" images or "non-face" images. On the other hand, the recognizer classifies the feature vector (assuming it is from a "face" image) as a known or unknown face.

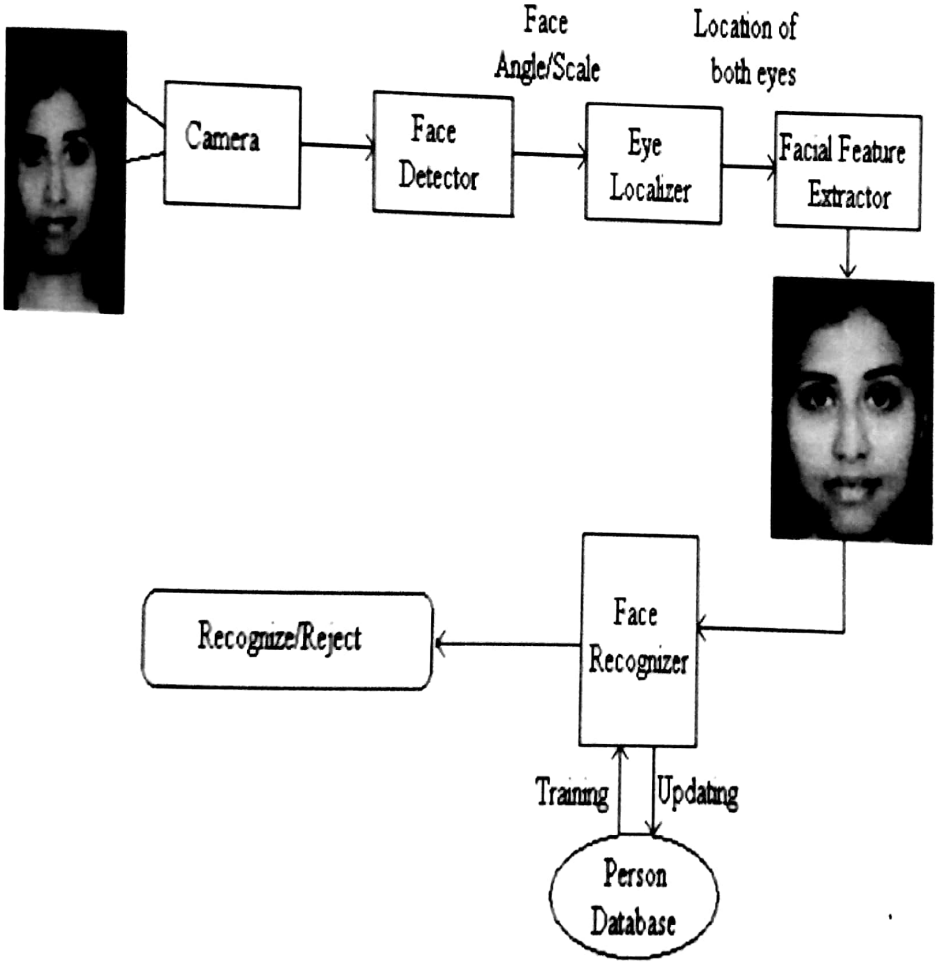


Figure 2-2: Generic framework for face detection and recognition system

2-6: Face detection

Face detection is a computer technology that determines the locations and sizes of human faces in arbitrary (digital) images.

FACE DETECTION ALGORITHM

Face detection algorithm is based on eye analogue segmentation

1) Convert the image into gray level image if it is colored.

2) Discard the light pixels of the image via thresholding and assign them the value of 255 (The value of white pixel is assigned as we want to concentrate on dark portions).

3) Apply the Laplacian high pass filter to sharpen the image and remove the isolated specks from the image.

4) Label the 8-connected segments of the resultant image.

5) Compute the centroid, area, width and height of the labeled segments.

6) Select the components that satisfy several predefined constraints. These constraints use many threshold variables that were decided after analysis of a large number of images. E.g. the segment area to image area ratio should lie within a specified range. The segment's major to minor axis ratio should be greater than MINMJMIRATIO and less than MAXMJMIRATIO. All the threshold variables and constraints are invariant of the size and illumination of the image which makes the algorithm robust.

7) Make pairs of remaining eye analogue segments. Take each pair and check whether or not they are eyes. Several constraints are used for this purpose. E.g. Horizontal distance between the eye centroids should be less than MaxXdistance threshold. Vertical distance between the eye centroids should be less than MaxYdistance threshold. Discard the pairs that don't satisfy these constraints.

8) If more than one pair is still left, select the best pair .If only one pair is left it is detected as eye. If there is no pair, no face is detected in the image.

9) Calculate the rotation angle (say A degrees) according to the difference of Y coordinates of the centroids of the detected eyes. Accordingly rotate the image by A degrees so that Y difference of eye centroids is zero. This is an indication of straight face. Rotational independence is achieved by this step.

10) Calculate the face rectangle boundaries by using difference of X coordinates of the centroids (diffx) of eyes. There are some predefined constants (itop, ibot, ileft, iright) that help in specifying the face rectangle. Thus by considering only the significant portions of the face for the recognition purpose we achieve robustness against temporary facial features such as a beard. E.g. top = diffx * itop Bottom = diffx * ibot Left = diffx * ileft Right = diffx *iright 11) Rescale the image according to predefined standard using nearest neighbor approach, bilinear approach or cubic approach. Thisstep is performed to achieve scaling independence.

Figure 2-3: (a) Original image (b) detected face image

✓ Our project by project

event activity

```
public class Event extends ActionBarActivity
implements
    TextToSpeech.OnInitListener {
  TextToSpeech t1;
  String sp="Guest";
  @Override
protected void onCreate(Bundle
savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_event);
    Intent intent = getIntent();
    String stringData="Guest";
    String KEY="person";
   t1=new TextToSpeech(this, this);
if (null != intent) {
stringData= intent.getStringExtra(KEY)    }
if(stringData==null){
stringData="Guest";}
    String toSpeak ="Welcome "+stringData;
sp=stringData;
t1.speak(toSpeak, TextToSpeech.QUEUE_FLUSH,
null);
   TextView text1=(TextView)
findViewById(R.id.textview1);
text1.setText(""+stringData);
speakOut();Thread timerThread = new Thread(){
public void run(){
try{
sleep(8000);//Come to MainActivity after some
time
}catch(InterruptedException e){
e.printStackTrace()  }finally{
            Intent intent = new
Intent(Event.this,MainActivity.class);
startActivity(intent);   } });
timerThread.start();}
```

```
  @Override
public void onInit(int status) {
if (status == TextToSpeech.SUCCESS) {
int result = t1.setLanguage(Locale.US);
if (result == TextToSpeech.LANG_MISSING_DATA
            || result == TextToSpeech.LANG_NOT_SUPPORTED)
Log.e("TTS", "This Language is not supported");
       } else {
speakOut();}
    } else {
Log.e("TTS", "Initilization Failed!");}}
private void speakOut()
  String text = "Welcome "+sp+" Have a good Day";
t1.speak(text, TextToSpeech.QUEUE_FLUSH, null);
}@Override
public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.menu_main, menu);
return true;
  }
@Override
public void onPause(){
if(t1 !=null){
t1.stop();
t1.shutdown();  }
super.onPause()}
  @Override
public void onBackPressed() {
startActivity(new Intent(this, MainActivity.class));}
}
```

SplashScreen :

```
public class SplashScreen extends Activity {
  @Override
protected void onCreate(Bundle savedInstanceState) {
```

```
    // TODO Auto-generated method stub
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_splash_screen);
    Thread timerThread = new Thread(){
public void run(){
try{
sleep(3000);
}catch(InterruptedException e){
e.printStackTrace();
}finally{
            Intent intent = new
Intent(SplashScreen.this,MainActivity.class);
startActivity(intent);            }}};
timerThread.start();}
@Override
protected void onPause() {
    // TODO Auto-generated method stub
super.onPause();
finish();      }
final UUID faceId=UUID.fromString(stringData);
Log.e("id","id is = "+faceId);


    Button buttonsubmit=(Button)
findViewById(R.id.buttonsubmit);
    Button buttoncancel=(Button)
findViewById(R.id.buttoncancel);
speakOut();
form activity:
public class FormActivity extends
AppCompatActivity implements
    TextToSpeech.OnInitListener {
private ProgressDialog detectionProgressDialog;
  TextToSpeech t1;
  @Override
protected void onCreate(Bundle
savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_form);
    String stringData="Guest";
    String KEY="Id";
if (null != intent) {
stringData= intent.getStringExtra(KEY),}
final UUID faceId=UUID.fromString(stringData);
Log.e("id","id is = "+faceId);
    Button buttonsubmit=(Button)
findViewById(R.id.buttonsubmit);
    Button buttoncancel=(Button)
findViewById(R.id.buttoncancel);
speakOut();
buttonsubmit.setOnClickListener(new
View.OnClickListener() {
        @Override
public void onClick(View v) {


        EditText name=(EditText)
findViewById(R.id.name);
        EditText companyname=(EditText)
findViewById(R.id.companyname);


        EditText designation=(EditText)
findViewById(R.id.designation);
final String Name=name.getText().toString();
final String
Companyname=companyname.getText().toString();
final String Designation=designation.getText().toString();
int l=Name.length();
if(l !=0){
createperson(faceId, Name, Companyname, Designation);
}
else {
Toast.makeText(getApplicationContext(), "Enter your
name",
                Toast.LENGTH_LONG).show()}}});
buttoncancel.setOnClickListener(new
View.OnClickListener() {
```

```
        @Override
public void onClick(View v) {
            Intent intent = new
Intent(FormActivity.this,MainActivity.class);
startActivity(intent)}});
detectionProgressDialog = new ProgressDialog(this);}
@Override
public void onInit(int status) {
if (status == TextToSpeech.SUCCESS) {
int result = t1.setLanguage(Locale.US);
if (result == TextToSpeech.LANG_MISSING_DATA
                || result ==
TextToSpeech.LANG_NOT_SUPPORTED) {
Log.e("TTS", "This Language is not supported");
        } else {
speakOut();}
      } else {
Log.e("TTS", "Initilization Failed!");}}
private void speakOut() {
     String text = "Sorry, We Dont have you in our
records. Please register";
t1.speak(text, TextToSpeech.QUEUE_FLUSH, null);}
   @Override
public void onPause(){
if(t1 !=null){
t1.stop();
t1.shutdown();}
super.onPause();}
private FaceServiceClient faceServiceClient =
new
FaceServiceClient("e4c5fc86f5c046459fcd99ff89ee33a2")
;//Insert API Key
Public void createperson(UUID faceId,String
Namee,String Companynamee,String Designation){
Log.e("Inside createperson", "Inside createperson");
Log.e("Namee","Namee "+Namee);
final String groupId="Persongroupid";//Insert
Persongroupid here
final String Name=Namee;
final String
UserData=Designation+";"+Companynamee+";";
```

```
final UUID[] Ids = new UUID[1];
Ids[0] = faceId;
AsyncTask<Void, String, CreatePersonResult>
createpersonTask =
new AsyncTask<Void, String, CreatePersonResult>() {
protected CreatePersonResult doInBackground(Void...
Params) {
try{
publishProgress("CreatingPerson...");
                CreatePersonResult result =
faceServiceClient.createPerson(groupId, Ids, Name,
UserData);
if (result == null)   publishProgress("CreatingPerson
Finished. Nothing Found");
return null; }
publishProgress(String.format("CreatingPersonFinished.
));
return result;
}catch (Exception e) {
publishProgress("createperson failed");
return null;}}
@Override
protected void onPreExecute() {
            //TODO: show progress dialog
detectionProgressDialog.show();}
          @Override
protected void onProgressUpdate(String... progress) {
            //TODO: update progress
detectionProgressDialog.setMessage(progress[0]);}
Log.e("onPostExecute","onPostExecute"+result.personId
);
```

```
            //TODO: update face frames
detectionProgressDialog.dismiss();
if (result == null) return;
displayPersonResult(result);}};
createpersonTask.execute()}
public void displayPersonResult(CreatePersonResult
res){
Log.e("displayPersonResult", "displayPersonResult"
);
    UUID PersonId=res.personId;
Log.e("PersonId is", "PersonId is " + PersonId);
train();}
@Override
public void onBackPressed() {
startActivity(new Intent(this, MainActivity.class)); }
public void train(){
Log.e("Inside Traning", "Inside Traning");
final String
PersonGroupId="Persongroupid";//Insert
Persongroupid here
    AsyncTask<Void, String, TrainingStatus>
trainTask =
new AsyncTask<Void, String, TrainingStatus>() {
protected TrainingStatus doInBackground(Void...
Params) {
publishProgress("Traning...");
                TrainingStatus result =
faceServiceClient.trainPersonGroup(PersonGroupId)
;
if (result == null){
publishProgress("Traning Finished. Nothing Found");
return null;}
publishProgress(
String.format("Traning Finished."));
return result;
}catch (Exception e) {
publishProgress("Traning failed")   return null;}}
```

```
            @Override
protected void onPreExecute() {
                //TODO: show progress dialog
detectionProgressDialog.show();}
            @Override
protected void onProgressUpdate(String... progress) {
                //TODO: update progress
detectionProgressDialog.setMessage(progress[0])
            @Override
protected void onPostExecute(TrainingStatus result) {
                //TODO: update face frames
detectionProgressDialog.dismiss();
if (result == null) return;
displayTrainResult(result)              }};
trainTask.execute();}
public void displayTrainResult(TrainingStatus res){
Log.e("In displayTrainResult", "In displayTrainResult");
if (res != null) {
        String Id=res.id;
        String Status=res.status;
        String StartTime=res.startTime;
        String EndTime=res.endTime;
Log.e("displayTrainResult", "Is is "+Id+" Traning Status
"+Status+" StartTime "+StartTime+" EndTime "+EndTime);
        Intent intent = new
Intent(FormActivity.this,MainActivity.class);
startActivity(intent);
    }

  }
}
```

Chapter 3

System analysis and database

3-1: Introduction

This chapter will navigate the software analysis phase .the analysis starts with the problem definition.

Formally, system analysis is thedissection of a system into its component pieces for purp2oses ofstudying how those component pieces interact and work. With respect to information system development, system analysis is the survey and planning of the system and project, the study and analysis of the existing business and information system, and the definition of business requirements and priorities for a new or improved system. There are several popular or emerging strategies for system analysis. This technique can be used in combination with

One another

3-2: static context diagrams:

Rather than simply depicting the list of actors, which does not provide any additional information with regard to a textual list, we can draw a diagram that we will call *static context diagram* .To do this, simply use a class diagram in which each actor is linked to a central class representing the system by an association, which enables the number of instances of actors connected the system at a given time to be specified.

Even though this is not a traditional UML diagram, we have found this kind of "context diagram" very useful in our practical experience.

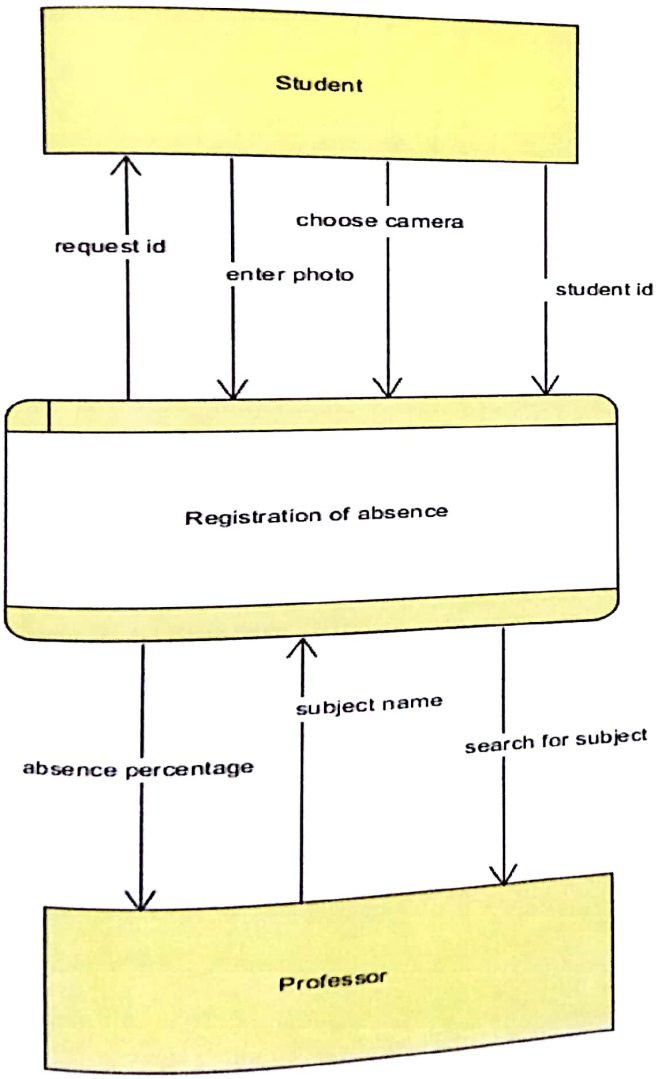


Figure 3-1: context diagram

3-2-1: Data flow diagram (level 0 dfd)

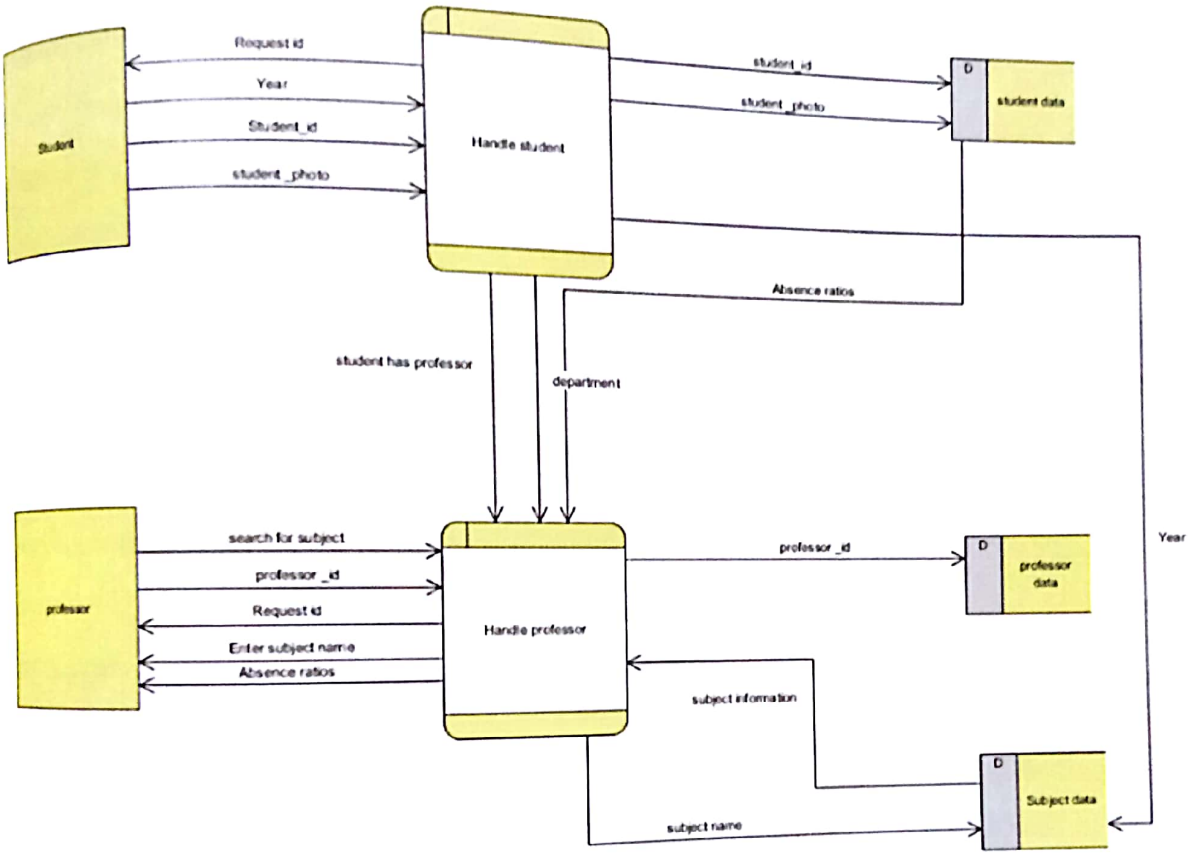


Figure 3-2: Data flow diagram (dfd)

3-3: use cases

A *use case* represents the specialization of a sequence of actions, including variants that system can perform interacting with actors of the system. A use case models a service offered by the system .it expresses the actor/system interactions and yields an observable result of value to an actor.

We can prepare a preliminary list of use cases of the registerabsence project, in order of actors.

We can prepare a preliminary list of use cases of the medicine management system project, in order of actors.

1- Students

-register into system

-enter id

-enter year

-browse camera

-take photo

2-professor

-register into system

-enter id

-search for subject

-request absence ratio

3- Verification system

-verify data

Creating use case diagrams

We are now going to give concrete expression to our identification of use cases by realizing UML diagrams, aptly called use case diagram shows the relationships among actors and the system, and use cases. We can easily obtain a preliminary diagram, show the use cases (ellipses) inside the MMS (box) and linked by association (lines) to their primary actors. such as figure 3-3 :

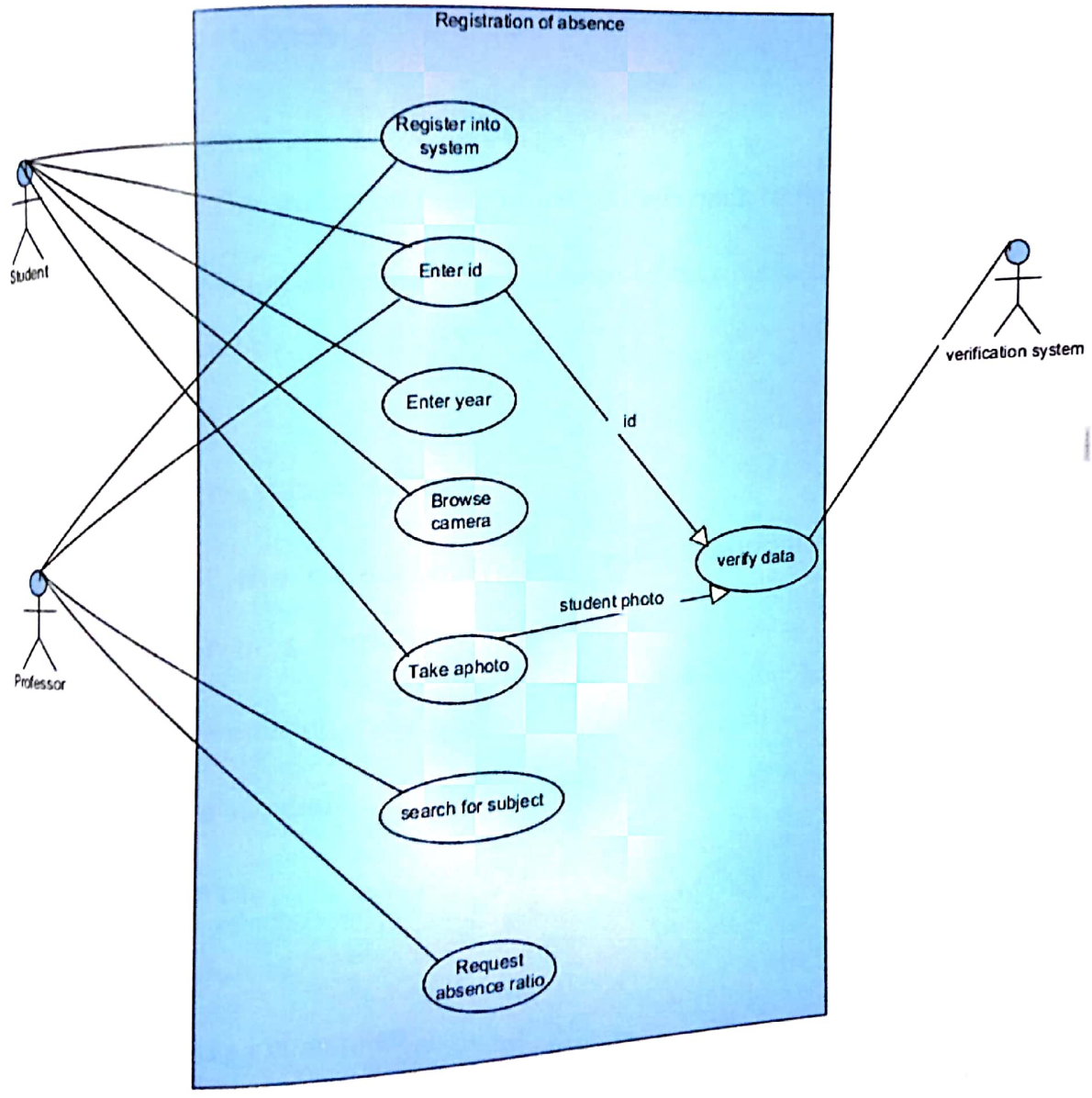


Figure 3-3: Use cases

3-4:database

✓ database definition

Database is a system which facilitates shared access to data in database, and which maintain the reliability, security, and integrity of the database by controlling access to it and supervising updates.

✓ Purpose of database

There is not so much the storage of the information as its quick retrieval

In other words, you must structure your database so that it can be queried quickly and efficiently.

✓ Benefits of the database

The advantage of the database system over traditional, based methods of record-keeping can be summarized as following:-

-Redundancy can be reducing.

Inconsistency can be avoided

The data can be shared.

Standards can be enforced.

Security restrictions can be applied.

Integrity can be maintained.

Speed: the machine can retrieve and change data far faster than a human can.

Currency: Accurate, up-to-date information is available on demand at any time.

Conflicting requirements can be balanced.

3.5: Main elements of entity relationship diagram (ERD)

1-Entity Definition

An entity is a thing of significance about which the user needs information

Example1:

For MMS, the following are examples of different entities:

Professor

Student

Subjects

student_has_professor

Department

Subject

Student_has_subject

Year

2-Attribute definition

-specific pieces of information which need to be known.

-An entity should have attributes.

Attribute Optional

Mandatory attributes:

-A mandatory attribute means that a value must be stored for this attribute for each entity instance (NOT NULL).

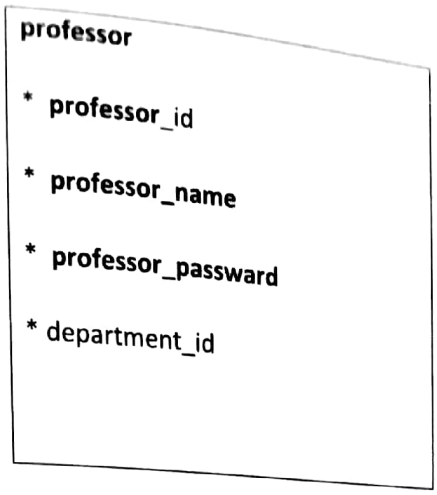
-A mandatory attribute will be tagged with *.

Optional Attributes:

- A value may be stored for each entity instance (or may be NULL)

-will be tagged with 0.

Example 2



3-6: Relationship Types:

1-one-to-one (1:1):

Have a degree of one and only one in the both direction.

Are rare.

2-Many-to-One (M: 1 or 1: M):

Have a degree of one or more in one direction and a degree of one and only one in the other direction.

Are very common.

Many-to-Many (M: M):

ave a degree of one or more in both directions.

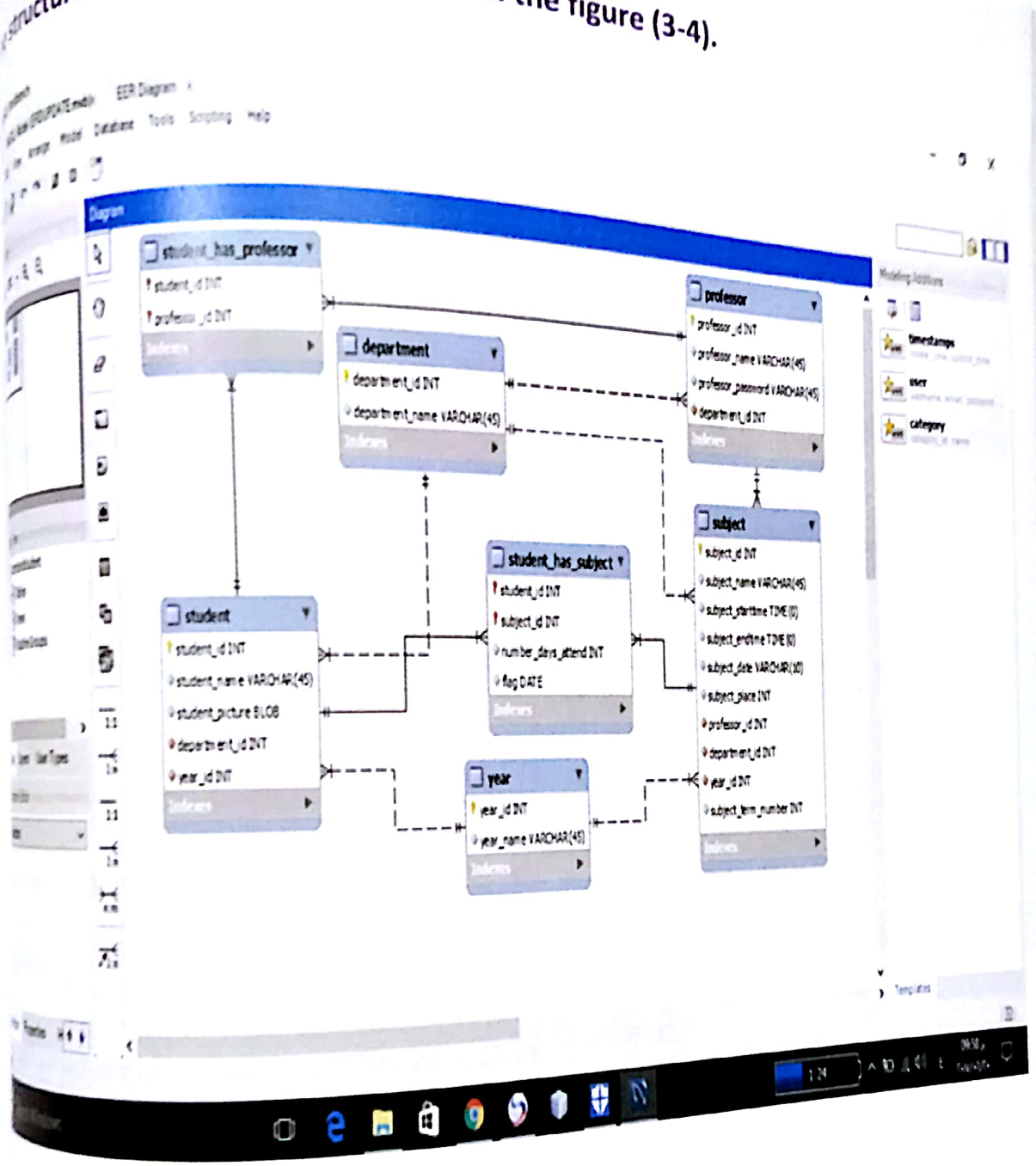
tity Relationship Diagram (ERD)

his document is an entity relationship diagram, for system to manage electronic sources. An ERD is a model that identifies the concepts or entities that exist in a stem

nd the relationship between those entities. An ERD is often used a way to visualize a ational database .each entity represents a database table, and relationship lines epresent the keys in one table that point to specific records in related tables.

he project presents a multimedia database for (medicine management system).this atabase stores the complete data about (Pharmacy's information, medicine, doctor, order, egister, patient, firm owner).Our system gives many operations on the database such as earching for (medicine, pharmacy ,firm owner) ,finding the nearest pharmacy, giving order r (doctor, firm owner) . We will use the workbench program to design the ER diagram.

structure of the program is shown in the figure (3-4).



e 3-4: Erd

Database Table

first aim of our application is to design general database system for any user in the pany, when the application run for the first time the user must provide the application equired data to build the system related to its company.

data base will design as follows:

ach company we create a separate database .Most of these tables are created in the g model, the table name are taken from the data added,

e. if the developer wants to design database for doctor then the table will named or.

made up of seven tables:

rofessor

tain all information about professor such as

fessor _id, professor_name , professor_password)

udent

tain all information about student such as

dent_id, student _name, student _picture)

bject

tain all information about subject such as

ject_id, subject_name, subject_starttime, subject_endtime, date,

subject_term_number)

4-department

(department_id, department name)

5-student_has_subject

Contain (student_ id, subject_id, number_day_attent, flage date)

6-year

Contain (year_id, year_name)

7-student_has_professor

Contain (student_ id ,professor_id)

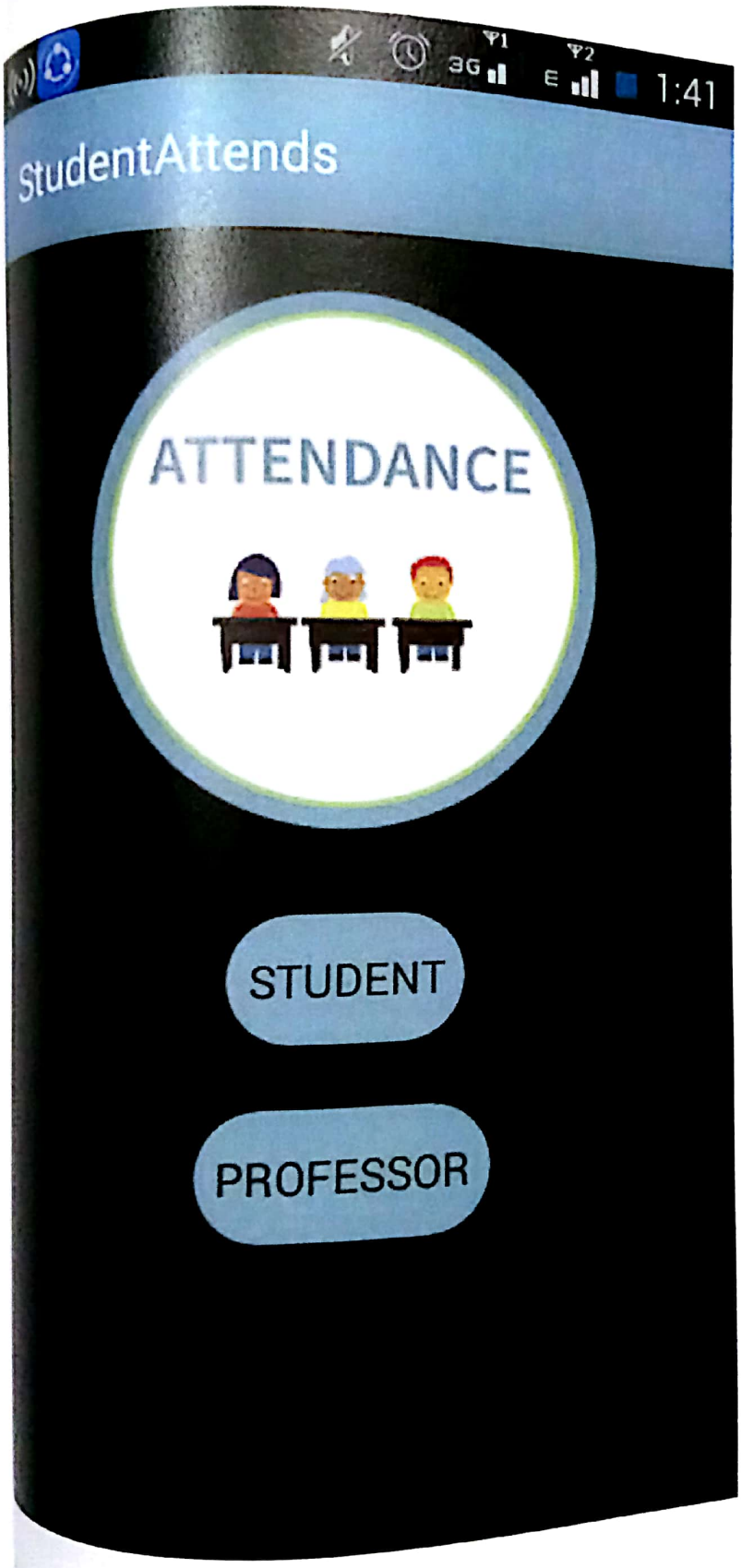
❖ filling data in database

After providing the system with required data, the system is dealing with this data, so we must have the ability to operate these data by some operations like search, request order, register for user. We should provide information about each table

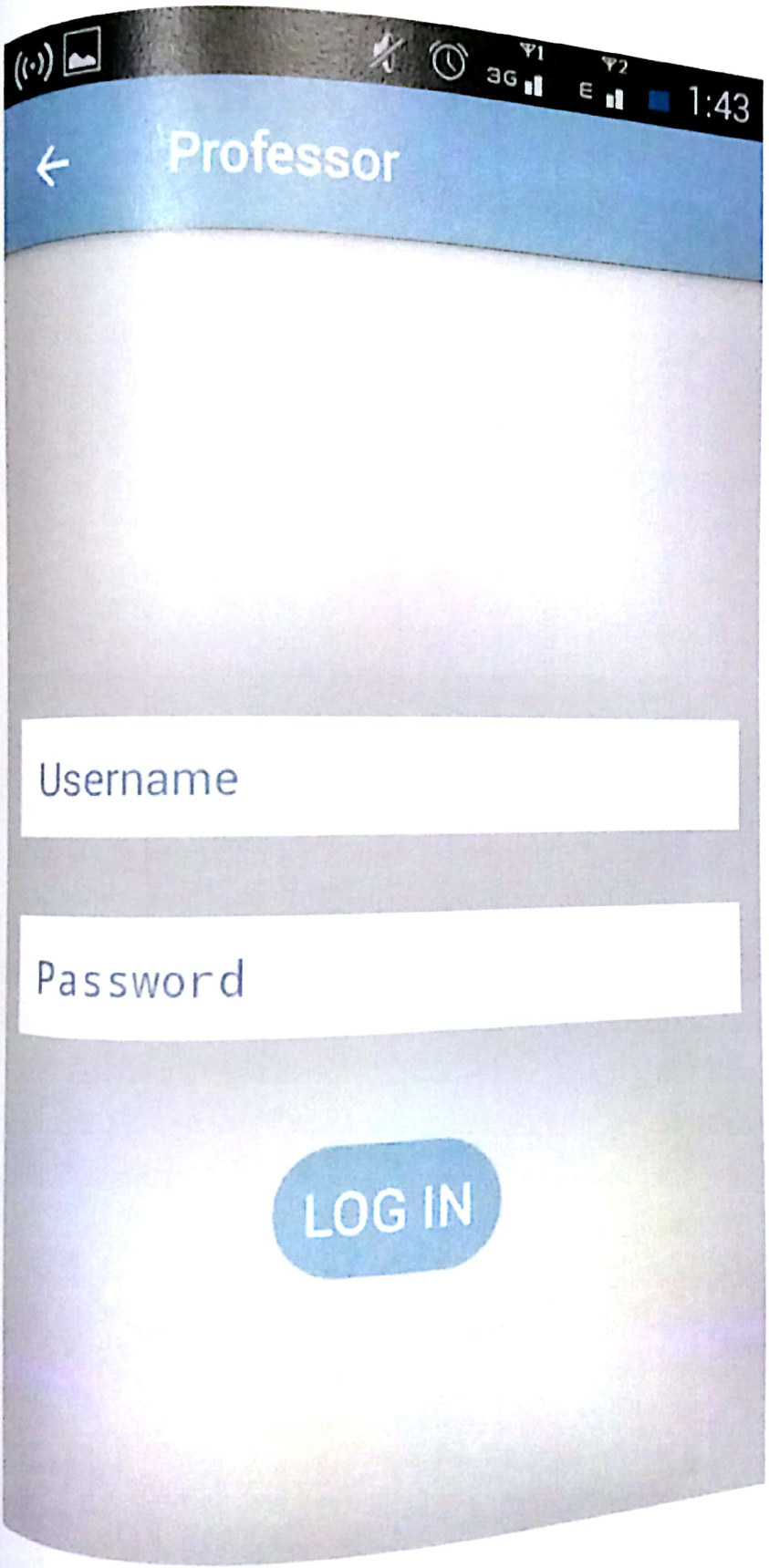
Chapter 4

Our Interface of project

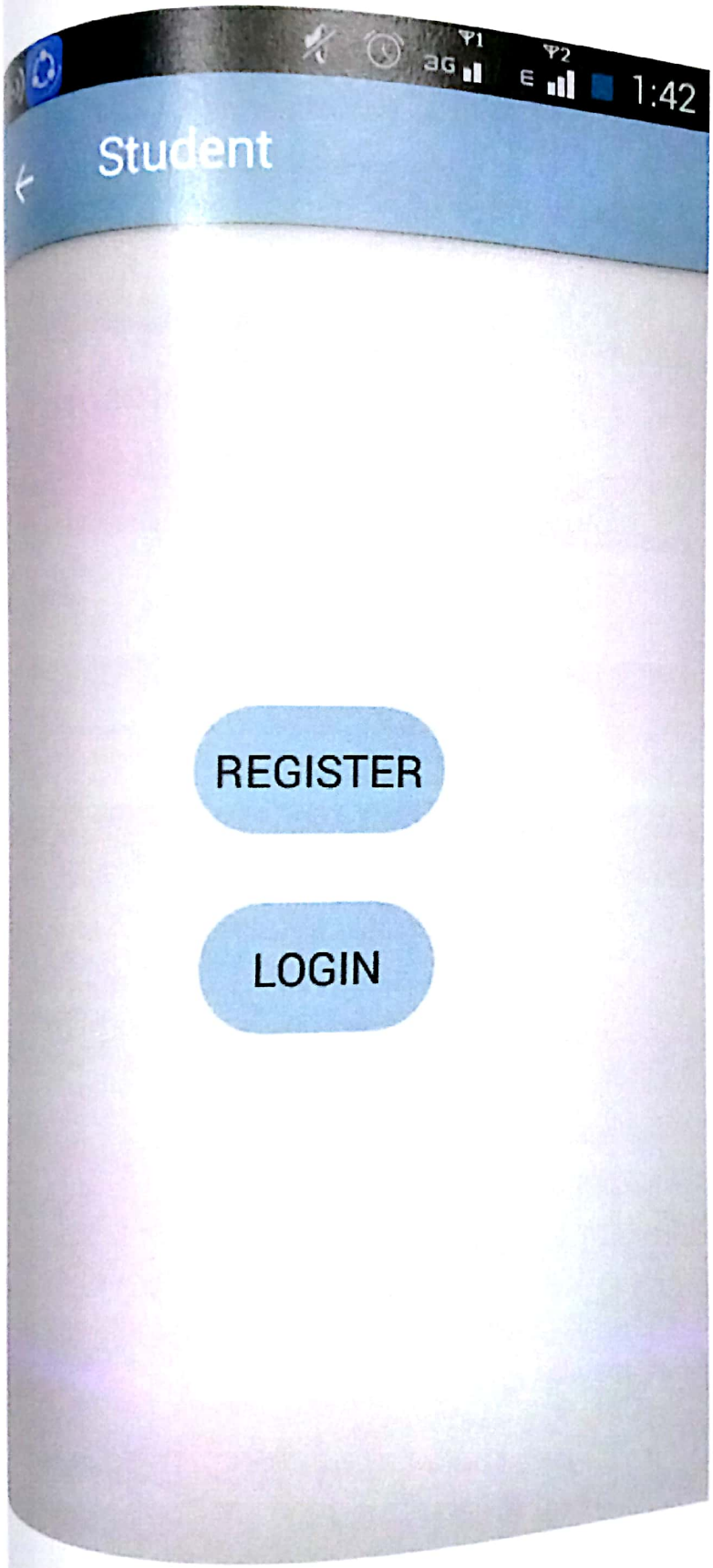
Our home page:



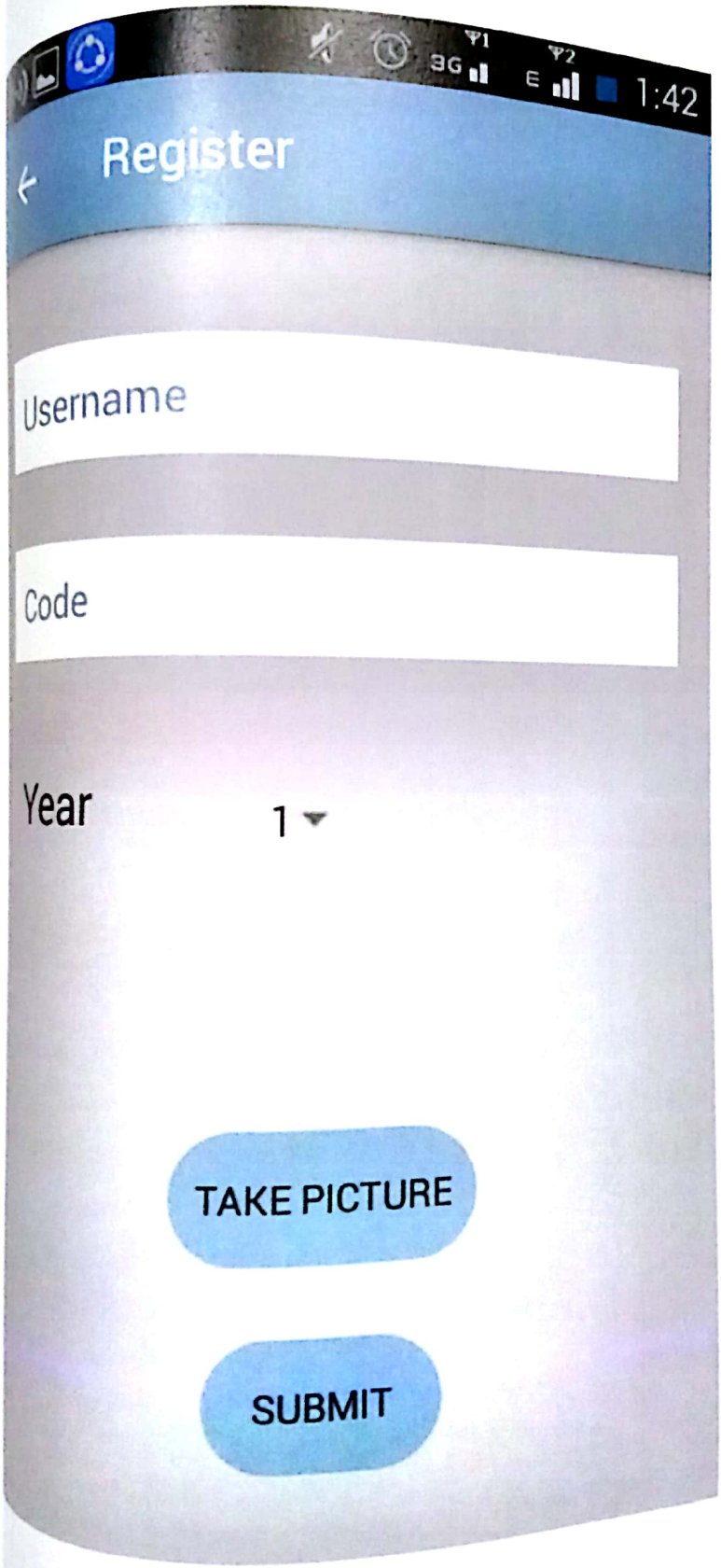
rofessor page:

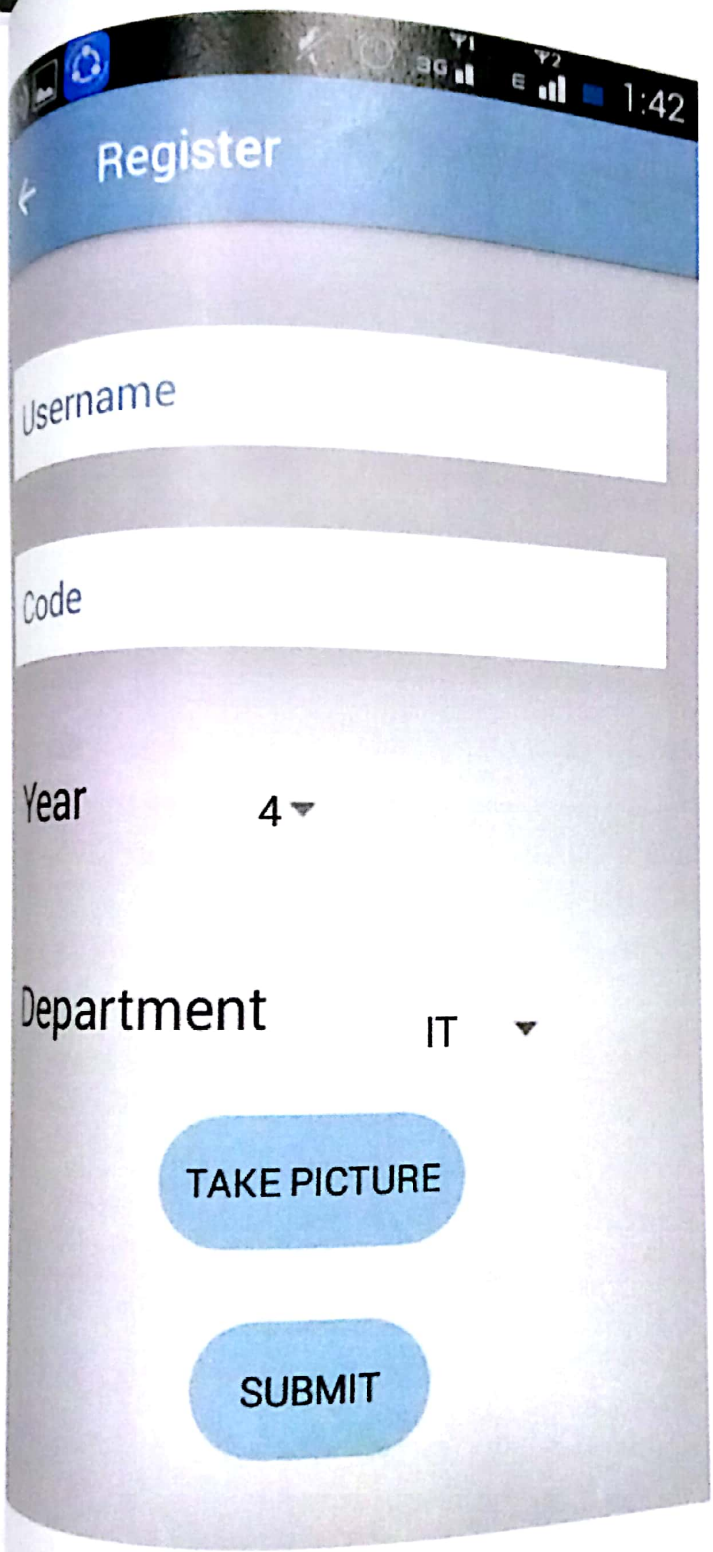


udent's page:

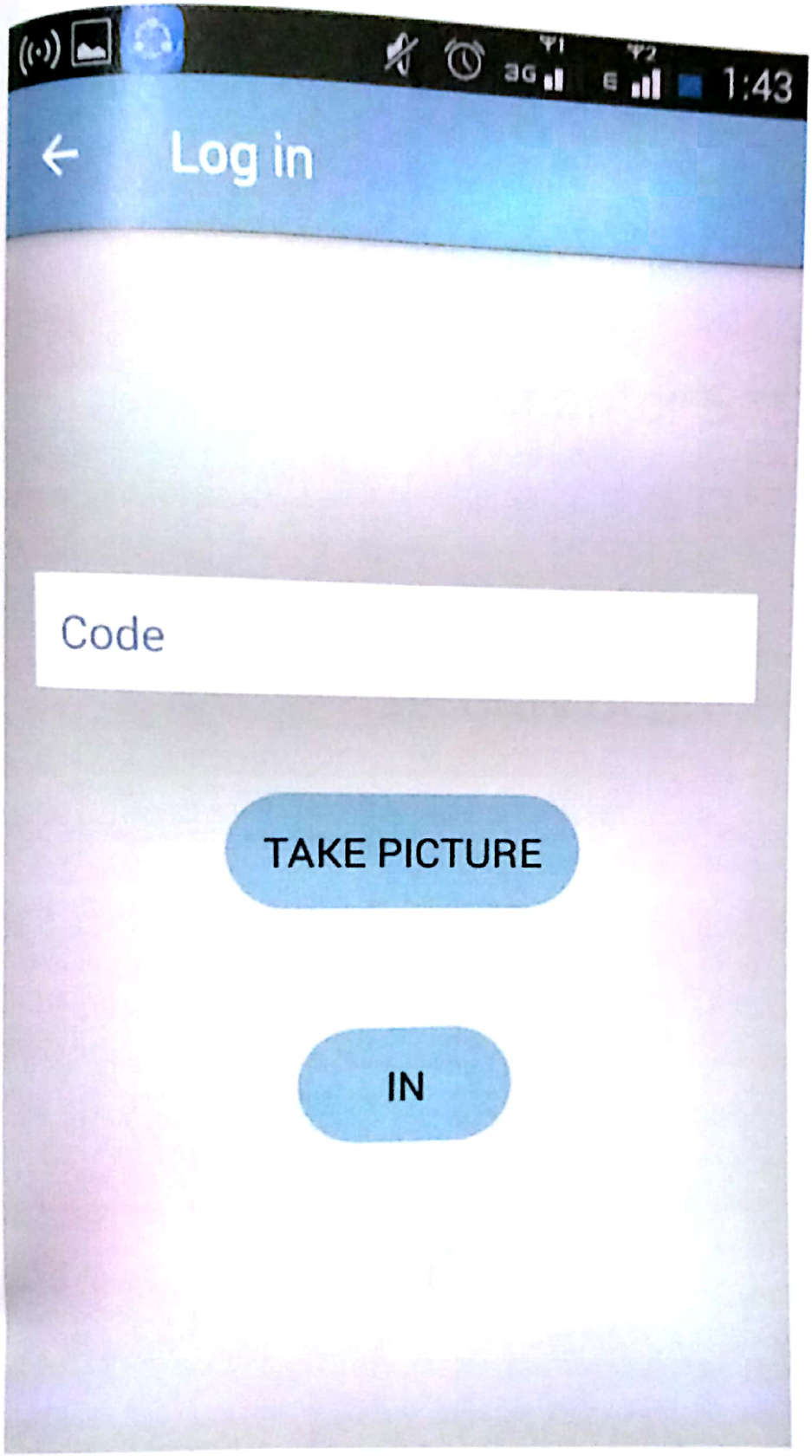


egister Student page:




1:42
Register
Username
Code
Year
4
Department
IT
TAKE PICTURE
SUBMIT

Login Student page:



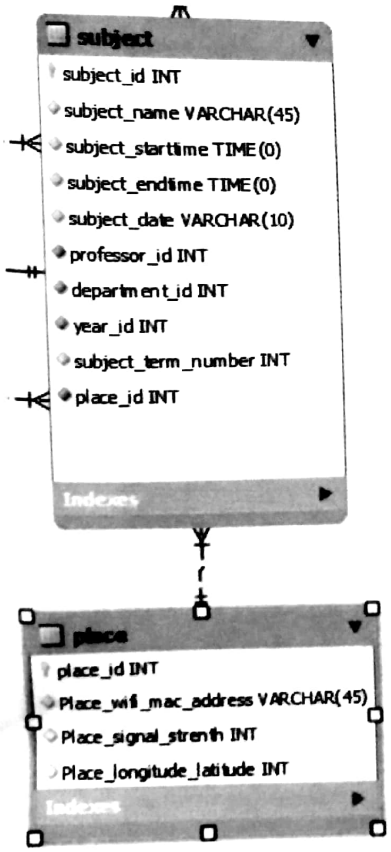
Chapter 5

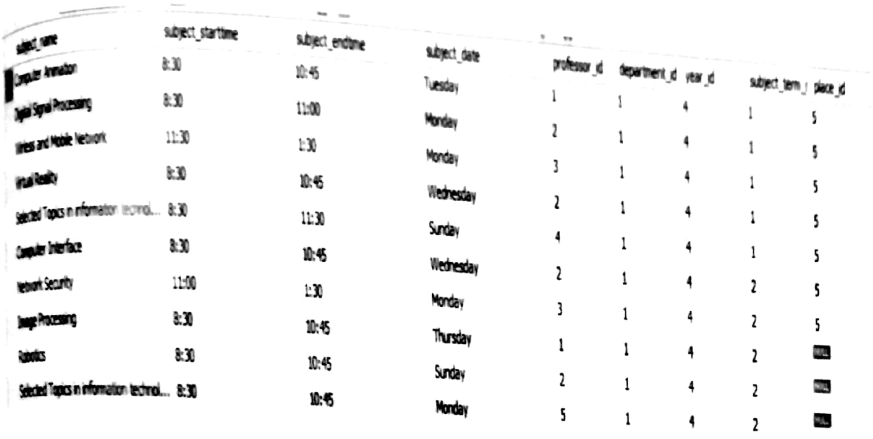
Security of project

Security chapter of how to verify that the student is in college and in his lectures

Security

Security in our program is to make sure that the student is really in lecture and to make sure of this we use three methods We use one of this methods to make sure that student is in right place of lecture and in the time of lecture. to do not make student attend in another time of lecture and make attend we make rules to prevent that as no one has to attend twice in the same lecture in the same day and make offer to professor to update place of lecture and time that put in database in subject table and in place table that in subject table to make sure we are in the time we use two columns the subject_starttime and the subject_endtime and the third is the subject day to make sure that the student do not attend in false day and start time and end time to make sure that the really time of lecture as in the table or the professor update it we use place_id as foreign key of the table place in table subject to make sure that the time and date and place of the lecture .





subject_name	subject_starttime	subject_endtime	subject_date	professor_id	department_id	year_id	subject_term_i	place_id
Computer Animation	8:30	10:45	Tuesday	1	1	4	1	5
Digital Signal Processing	8:30	11:00	Monday	2	1	4	1	5
Wires and Mobile Network	11:30	1:30	Monday	3	1	4	1	5
virtual Reality	8:30	10:45	Wednesday	2	1	4	1	5
Selected Topics in information technol...	8:30	11:30	Sunday	4	1	4	1	5
Computer Interface	8:30	10:45	Wednesday	2	1	4	2	5
Network Security	11:00	1:30	Monday	3	1	4	2	5
Image Processing	8:30	10:45	Thursday	1	1	4	2	NULL
Robotics	8:30	10:45	Sunday	2	1	4	2	NULL
Selected Topics in information technol...	8:30	10:45	Monday	5	1	4	2	NULL

Our tools that used for security are:

1. Use the wifi signal strength range.
2. Use the wifi mac address of the router in the lecture place.
3. Use the longitude and latitude of the student place.

Explaining of security steps:

1. using the wifi signal strength range that used when we know the range is excellent in where and good and week in where place of lecture and that will help us in determine lecture place in right place and where student exist in lecture we will use it with this code

```
WifiManager wifiManager = (WifiManager) context.getSystemService(Context.WIFI_SERVICE);
int numberOfLevels = 5;
WifiInfo wifiInfo = wifiManager.getConnectionInfo();
int level = WifiManager.calculateSignalLevel(wifiInfo.getRssi(), numberOfLevels);
```

The method calculateSignalLevel is built in method in android that take the number of levels to consider in the calculated level.Andthe power of the signal measured in RSSI. And calculate the rang and we compare by the level in database
In database and use it as that place_id insubject we store at place_signal_strength

2. using the wifi mac address of the lecture router that is good idea if the wifi rang do not out of lecture place that make sure that the student in the lecture place is the only have the right to attend we use the same package wifi manager and wifiInfo class to make that and we take the mac address and store in database by id

We use code in store mac address

```
public String getMacId() {

    WifiManager wifiManager = (WifiManager) getSystemService(Context.WIFI_SERVICE);
    WifiInfo wifiInfo = wifiManager.getConnectionInfo();
    return wifiInfo.getBSSID();
}
```

We use column wifi_macAddress to store it and different of it this methode is good and helpful but if the rang is covered the outside of the lecture the student can attend without being in and if we have not routers or access points in all lecture places that do not work

3. using the longitude and latitude that we use by android and android provide two provider that as gps provider and network provider the gps get the longitude and latitude by the gps sensor in the mobile but that have error and the second way to get the longitude and latitude by network provider and that use cell of mobile phone and accurate as possible as that can and we can know the student place in lecture by it that will but in database in the column place_ longitude_latitude to make sure that the place of the student we must store the place of lecture of the longitude and latitude that the three methods and we use in our project we use code for the third that location manager package and use location listener that override four method onLocationChange that read the location change of the longitude and latitude and onStatusChange that make sure the cell do not close as close the network onProvierEnable that make sure that the provider is enable onProvierDisable that make sure that the provider is disable

Code of eigen face

```
eRecognitionAppActivity:

ate void show Toast(String message, int duration} (

uration != Toast.LENGTH_SHORT && duration != Toast.LENGTH_LONG(

w new IllegalArgumentException;()

mToast != null && mToast.getView().isShown(()

oast.cancel(); // Close the toast if it is already open

oast = Toast.makeText(this, message, duration;(

oast.show;()


ate void addLabel(String string} (

ng label = string.substring(0, 1).toUpperCase(Locale.US) +
ng.substring(1).trim().toLowerCase(Locale.US ;(

 Make sure that the name is always uppercase and rest is lowercase

agesLabels.add(label); // Add label to list of labels

g.i(TAG, "Label: " + label;(

ainFaces(); // When we have finished setting the label, then retrain faces

ate void trainFaces} ()

images.isEmpty(()

turn; // The array might be empty if the method is changed in the
nClickListener

at imagesMatrix = new Mat((int) images.get(0).total(), images.size(),
ages.get(0).type;(()

r (int i = 0; i < images.size(); i(++

ages.get(i).copyTo(imagesMatrix.col(i)); // Create matrix where each image
epresented as a column vector

g.i(TAG, "Images height: " + imagesMatrix.height() + " Width: " +
agesMatrix.width() + " total: " + imagesMatrix.total;(()

     Train the face recognition algorithms in an asynchronous task, so we
o not skip any frames

useEigenfaces(

w NativeMethods.TrainFacesTask(imagesMatrix).execute;()

ise}


<String> uniqueLabelsSet = new HashSet<>(imagesLabels); // Get all
ique labels

ng[] uniqueLabels = uniqueLabelsSet.toArray(new
ng[uniqueLabelsSet.size()]); // Convert to String array, so we can read the
ues from the indices

[] classesNumbers = new int[uniqueLabels.length;[

r (int i = 0; i < classesNumbers.length; i(++

lassesNumbers[i] = i + 1;
```

```
//Create incrementing list for each unique label starting at 1


int[] classes = new int[imagesLabels.size;[()

for (int i = 0; i < imagesLabels.size(); i} (++

String label = imagesLabels.get(i;(

for (int j = 0; j < uniqueLabels.length; j} (++

if (label.equals(uniqueLabels[j} (([

classes[i] = classesNumbers[j]; // Insert corresponding number

break;

{

{

{


Mat vectorClasses = new Mat(classes.length, 1, CvType.CV_32S);
// CV_32S == int

vectorClasses.put(0, 0, classes); // Copy int array into a vector


new NativeMethods.TrainFacesTask(imagesMatrix,
vectorClasses).execute;()

{

{

private void showLabelsDialog} ()

Set<String> uniqueLabelsSet = new HashSet<>(imagesLabels); //
Get all unique labels

if (!uniqueLabelsSet.isEmpty()) { // Make sure that there are any
labels

 //          Inspired by:
http://stackoverflow.com/questions/15762905/how-can-i-
display-a-list-view-in-an-android-alert-dialog

AlertDialog.Builder builder = new
AlertDialog.Builder(FaceRecognitionAppActivity.this;(

builder.setTitle("Select label;(":

builder.setPositiveButton("New face", new
DialogInterface.OnClickListener} ()

@               Override

public void onClick(DialogInterface dialog, int which} (

dialog.dismiss;()

showEnterLabelDialog;()

{

;({
```

```
uilder.setNegativeButton("Cancel", new
ialogInterface.OnClickListener() {
            Override
@
ublic void onClick(DialogInterface dialog, int which) {
ialog.dismiss();
mages.remove(images.size() - 1); // Remove last image
}
{
uilder.setCancelable(false); // Prevent the user from closing the dialog


tring[] uniqueLabels = uniqueLabelsSet.toArray(new
tring[uniqueLabelsSet.size()]); // Convert to String array for
rrayAdapter
rrays.sort(uniqueLabels); // Sort labels alphabetically
inal ArrayAdapter<String> arrayAdapter = new
rrayAdapter<String>(FaceRecognitionAppActivity.this ,
ndroid.R.layout.simple_list_item_1, uniqueLabels) {
@           Override
ublic @NonNull View getView(int position, @Nullable View
onvertView, @NonNull ViewGroup parent) {
extView textView = (TextView) super.getView(position, convertView,
arent);
f (getResources().getBoolean(R.bool.isTablet(
extView.setTextSize(20); // Make text slightly bigger on tablets
ompared to phones
lse
extView.setTextSize(18); // Increase text size a little bit
eturn textView;
}
{
istView mListView = new ListView(FaceRecognitionAppActivity.this);
istView.setAdapter(arrayAdapter); // Set adapter, so the items
ctually show up
uilder.setView(mListView); // Set the ListView
inal AlertDialog dialog = builder.show(); // Show dialog and store in
inal variable, so it can be dismissed by the ListView
istView.setOnItemClickListener(new
dapterView.OnItemClickListener() {
```

```
@Override
public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
dialog.dismiss();
addLabel(arrayAdapter.getItem(position);((
{
:((
{       else
showEnterLabelDialog(); // If there is no existing labels, then ask the
user for a new label
{
private void showEnterLabelDialog() {
AlertDialog.Builder builder = new
AlertDialog.Builder(FaceRecognitionAppActivity.this;(
builder.setTitle("Please enter your name;(":
final EditText input = new EditText(FaceRecognitionAppActivity.this;(
input.setInputType(InputType.TYPE_CLASS_TEXT;(
builder.setView(input;(
builder.setPositiveButton("Submit", null;(
//        Set up positive button, but do not provide a listener,
so we can check the string before dismissing the dialog
builder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {
dialog.dismiss();
images.remove(images.size() - 1); // Remove last image
        }
    });
builder.setCancelable(false); // User has to input a name
    AlertDialog dialog = builder.create();
    // Source: http://stackoverflow.com/a/7636468/2175837
dialog.setOnShowListener(new DialogInterface.OnShowListener() {
        @Override
public void onShow(final DialogInterface dialog) {
        Button mButton = ((AlertDialog)
dialog).getButton(AlertDialog.BUTTON_POSITIVE);
mButton.setOnClickListener(new View.OnClickListener() {
```

```
)Override
ublic void onClick(View view} (
ring string = input.getText().toString().trim:()
)string.isEmpty()) { // Make sure the input is valid
                    If input is valid, dismiss the dialog and add the
el to the array
alog.dismiss:()
ddLabel(string:(




(
    {
    Show keyboard, so the user can start typing straight away
ialog.getWindow().setSoftInputMode(WindowManager.LayoutParam
.SOFT_INPUT_STATE_VISIBLE;(
ialog.show;()
@Override
rotected void onCreate(Bundle savedInstanceState) {
uper.onCreate(savedInstanceState);
etWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_S
CREEN_ON);
etContentView(R.layout.activity_face_recognition_app);
nToolbar = (Toolbar) findViewById(R.id.toolbar);
etSupportActionBar(mToolbar); // Sets the Toolbar to act as the
ctionBar for this Activity window
    DrawerLayout drawer = (DrawerLayout)
indViewById(R.id.drawer_layout);
    ActionBarDrawerToggle toggle = new ActionBarDrawerToggle
    (this, drawer, mToolbar, R.string.navigation_drawer_open,
.string.navigation_drawer_close);
drawer.addDrawerListener(toggle);
oggle.syncState();
    RadioButton mRadioButtonEigenfaces = (RadioButton)
indViewById(R.id.eigenfaces);
nRadioButtonEigenfaces.setOnClickListener(new
iew.OnClickListener() {
    @Override
ublic void onClick(View v) {
howToast(getResources().getString(R.string.eigenfaces),
oast.LENGTH_SHORT),
eEigenfaces = true;
Eigenfaces();
    }
});
```

```
//Set radio button based on value stored in shared preferences
prefs = PreferenceManager.getDefaultSharedPreferences(this.(
useEigenfaces = prefs.getBoolean("useEigenfaces", false.(
mRadioButtonEigenfaces.setChecked(useEigenfaces.(
tinydb = new TinyDB(this); // Used to store ArrayLists in the shared
preferences
mThresholdFace = (SeekBarArrows)
findViewById(R.id.threshold_face;(
mThresholdFace.setOnSeekBarArrowsChangeListener(new
SeekBarArrows.OnSeekBarArrowsChangeListener} ()
@           Override
public void onProgressChanged(float progress} (
Log.i(TAG, "Face threshold: " +
mThresholdFace.progressToString(progress;((
faceThreshold = progress;
faceThreshold = mThresholdFace.getProgress(); // Get initial value
mThresholdDistance = (SeekBarArrows)
findViewById(R.id.threshold_distance;(
mThresholdDistance.setOnSeekBarArrowsChangeListener(new
SeekBarArrows.OnSeekBarArrowsChangeListener} ()
@           Override
public void onProgressChanged(float progress} (
Log.i(TAG, "Distance threshold: " +
mThresholdDistance.progressToString(progress;((
distanceThreshold = progress;
{
;({      distanceThreshold = mThresholdDistance.getProgress(); // Get
initial value
findViewById(R.id.clear_button).setOnClickListener(new
View.OnClickListener() {
        @Override
public void onClick(View v) {
Log.i(TAG, "Cleared training set");
images.clear(); // Clear both arrays, when new instance is created
imagesLabels.clear();
showToast("Training set cleared", Toast.LENGTH_SHORT);
        }
    });
```

```
ublic void onClick(View v} (

g.i(TAG, "Gray height: " + mGray.height() + " Width: " +
Gray.width() + " total: " + mGray.total;(()

(mGray.total() == 0(

turn;

ze imageSize = new Size(200, 200.0f / ((float) mGray.width() / (float)
Gray.height ;((()

//          Scale image in order to decrease computation time

mgproc.resize(mGray, mGray, imageSize;(

Log.i(TAG, "Small gray height: " + mGray.height() + " Width: " +
mGray.width() + " total: " + mGray.total;(()

//          SaveImage(mGray;(

Mat image = mGray.reshape(0, (int) mGray.total()); // Create column
vector

Log.i(TAG, "Vector height: " + image.height() + " Width: " +
image.width() + " total: " + image.total;(()

images.add(image); // Add current image to the array

//          Calculate normalized Euclidean distance

new NativeMethods.MeasureDistTask(useEigenfaces, new
NativeMethods.MeasureDistTask.Callback} ()

@               Override

public void onMeasureDistComplete(Bundle bundle} (

float minDist =
bundle.getFloat(NativeMethods.MeasureDistTask.MIN_DIST_FLOAT;(

if (minDist != -1) {

int minIndex =
bundle.getInt(NativeMethods.MeasureDistTask.MIN_DIST_INDEX_INT
);    float faceDist =
bundle.getFloat(NativeMethods.MeasureDistTask.DIST_FACE_FLOAT);

if (imagesLabels.size() > minIndex) { // Just to be sure

Log.i(TAG, "dist[" + minIndex + "]: " + minDist + ", face dist: " +
faceDist + ", label: " + imagesLabels.get(minIndex));
String minDistString = String.format(Locale.US, "%.4f", minDist);

                String faceDistString = String.format(Locale.US,
"%.4f", faceDist);

if (faceDist < faceThreshold && minDist < distanceThreshold) // 1.
Near face space and near a face class

showToast("Face detected: " + imagesLabels.get(minIndex) + ".
Distance: " + minDistString, Toast.LENGTH_LONG);

else if (faceDist < faceThreshold) // 2. Near face space but not near a
known face class

showToast("Unknown face. Face distance: " + faceDistString + ".
Closest Distance: " + minDistString, Toast.LENGTH_LONG);

else if (minDist < distanceThreshold) // 3. Distant from face space and
near a face class

showToast("False recognition. Face distance: " + faceDistString + ".
Closest Distance: " + minDistString, Toast.LENGTH_LONG);
```

face-lib.cpp:

```
Eigenfaces eigenfaces;

Fisherfaces fisherfaces;


using namespace std;

using namespace cv;

using namespace Eigen;


#ifdef __cplusplus

extern "C} "

#endif


JNIEXPORT void JNICALL
Java_com_lauszus_facerecognitionapp_NativeMethods_TrainFaces

)JNIEnv, jobject, jlong addrImages, jlong addrClasses} (

Mat *pImages = (Mat *) addrImages; // Each images is represented as a
column vector

Mat *pClasses = (Mat *) addrClasses; // Classes are represented as a
vector


LOG_ASSERT(pImages->type() == CV_8U, "Images must be an 8-bit
matrix;("

MatrixXi images;

cv2eigen(*pImages, images); // Copy from OpenCV Mat to Eigen matrix


//    Facebase *pFacebase;

if (pClasses == NULL) { // If classes are NULL, then train Eigenfaces

eigenfaces.train(images); // Train Eigenfaces

LOGI("Eigenfacess numComponents: %d",
eigenfaces.numComponents;(

//        pFacebase = &eigenfaces;

{    else}

LOG_ASSERT(pClasses->type() == CV_32S && pClasses->cols == 1,
"Classes must be a signed 32-bit vector;("

VectorXi classes;

cv2eigen(*pClasses, classes); // Copy from OpenCV Mat to Eigen vector

LOG_ASSERT(classes.minCoeff() == 1, "Minimum value in the list must
be 1;("

fisherfaces.train(images, classes); // Train Fisherfaces

LOGI("Fisherfaces numComponents: %d", fisherfaces.numComponents;(

//        pFacebase = &fisherfaces;

{
```