ABSTRACT

- **Title of the Project**
  Face Recognition Attendance System

- **Goal of the Project**
  Maintaining the attendance is essential in every foundation for checking the performance of students as well as employee. Each organization has its own technique. Traditionally student's attendance is taken physically on register or sheet, given by the employee in class. The previous approach in which manually takes and maintains the attendance records was very inconvenient task.
  Face Recognition Attendance System can capture images of users and save that all image records in database. When student or employee come in front of camera. Then system will check whether this face copies are saved in database or not. If the face is similar to database image, then system marks the attendance of that student or employee.

- **Software Requirements**
  1. OpenCV Framework
  2. Python 3.0 or higher
  3. Windows 8 or higher
  4. MySQL

- **Hardware Requirements**
  1. A Standalone computer (i3 5th Gen, 4GB Ram or higher)
  2. Camera to capture images

INTRODUCTION

In many Institution and Organization, the attendance is a very important factor to maintain the record of lectures, salary and work hours etc. Most of the institutes and organizations follow the manual method using old paper and file method and some of them have shifted to biometric technique. The current method that colleges use is that the professor passes a sheet or make roll calls and mark the attendance of the students and this sheet further goes to the admin department with updates the final excel sheet. This process is quite hectic and time consuming. Also, for professors or employees at institutes or organizations the biometric system serves one at a time. So, why not shift to an automated attendance system, which works on face recognition technique? Be it a classroom or entry gates it will mark the attendance of the students, professors, employees, etc. The main elements of this technology are as follows:

Face Detection:

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. Face detection also refers to the psychological process by which humans locate and attend to faces in a visual scene. Face detection can be regarded as a specific case of object-class detection. Face-detection algorithms focus on the detection of frontal human faces. It is analogous to image detection in which the image of a person is matched bit by bit. Image matches with the image stores in database. Any facial feature changes in the database will invalidate the matching process. Firstly, the possible human eye regions are detected by testing all the valley regions in the gray-level image. Then the genetic algorithm is used to generate all the possible face regions which include the eyebrows, the iris, the nostril and the mouth corners.

Face Recognition:

A facial recognition system is a computer application capable of identifying or verifying a person from a digital image. One of the ways to do this is by comparing selected facial features from the image and a facial database. It is typically used in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems. For example, an algorithm may analyses the relative position, size, and/or shape of the eyes, nose, cheekbones, and jaw. These features are then used to search for other images with matching features. Other algorithms normalize a gallery of face images and then compress the face data, only saving the data in the image that is useful for face recognition.

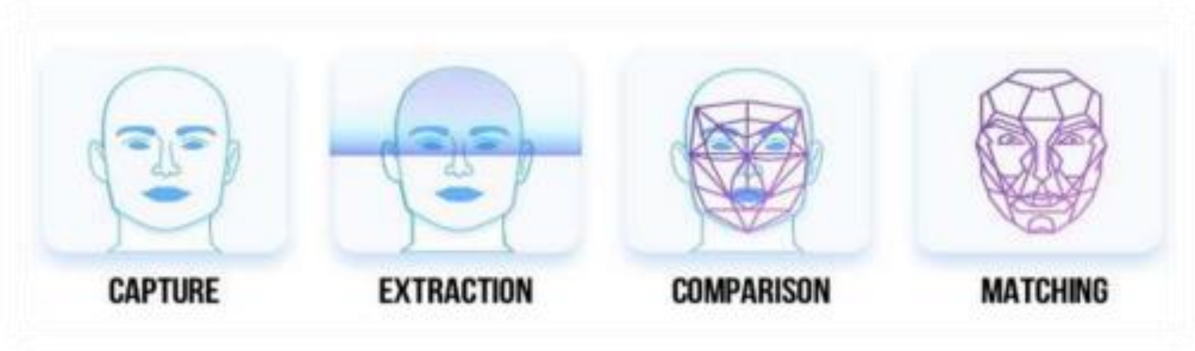
2.1 Background

Preserving the attendance is very crucial in all the institutes for checking the overall performance of students. Each institute has its very own method in this regard. A few are taking attendance manually using the old paper or document-based approach and some have adopted techniques of automated attendance the use of few biometric techniques. There are many computerized methods to be had for this reason i.e. biometric attendance. Face recognition appears to offer several advantages over other biometric methods, a few of which are outlined here: Almost all these technologies require some voluntary action by the user, i.e., the user needs to place his hand on a hand-rest for fingerprinting or hand geometry detection and has to stand in a fixed position in front of a camera for iris or retina for identification.

2.2 Problem Statement

Maintaining the attendance is essential in every foundation for checking the performance of students as well as employee. Each organization has its own technique. Traditionally student's attendance is taken physically on register or sheet, given by the employee in class. Physically recorded participation can be effortlessly controlled. Besides, it is exceptionally hard to confirm one by one student in a substantial classroom environment with disseminated branches whether the verified students are really reacting or not.

The previous approach in which manually takes and maintains the attendance records was very inconvenient task. Traditionally, student's attendances are taken manually by using attendance sheet given by the faculty members in class, which is a time consuming event. Moreover, it is very difficult to verify one by one student in a large classroom environment with distributed branches whether the authenticated students are actually responding or not. The ability to compute the attendance percentage becomes a major task as manual computation produces errors, and also wastes a lot of time. This method could easily allow for impersonation and the attendance sheet could be stolen or lost. An automatic attendance management system using biometrics would provide the needed solution.



SOFTWARE & HARDWARE REQUIREMENT ANALYSIS

3.1 Methods

OpenCv:

OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it is free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform.

Caffe:

Caffe is a well-known and widely used machine-vision library that ported Matlab's implementation of fast convolutional nets to C and C++. Caffe is not intended for other deeplearning applications such as text, sound or time series data. Caffe perform image classification 6 with convolutional nets, which represent the state of the art. In contrast to Caffe, Deeplearning4j offers parallel GPU support for an arbitrary number of chips, as well as many, seemingly trivial, features that make deep learning run more smoothly on multiple GPU clusters in parallel.

MATLAB:

MATLAB combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly. MATLAB toolboxes are professionally developed, rigorously tested, and fully documented. MATLAB apps let you see how different algorithms work with your data. Iterate until you've got the results you want, then automatically generate a MATLAB program to reproduce or automate your work. Scale your analyses to run on clusters, GPUs, and clouds with only minor code changes. There’s no need to rewrite your code or learn big data programming and out-of-memory techniques.

3.2 Working Environment

Python:

Python is an interpreted high-level programming language for general purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple 7 programming paradigms, including object oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems.

Tkinter:

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

OpenCV:

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is crossplatform and free for use under the open-source BSD license. OpenCV supports the deep learning frameworks TensorFlow, Torch/PyTorch and Caffe.

3.3 Requirements to run the application

Supportive Operating Systems: The supported Operating Systems for client include: Windows 2010, windows 2008, windows 2007.

3.3.1 Software Requirements:

The Software Requirements in this project include:

a. Python
b. OpenCV framework
c. MS Excel

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed. 8 OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

3.3.2 Hardware Requirements:

The most common set of requirements defined by any operating system or software application is the physical computer resources, known as hardware.

Components	**Minimum**	**Recommended**
Processor	Intel Core i3 2nd generation	Intel Core i7 5th generation
RAM	4GB	8GB
Camera	HD 720p Webcam	Full HD 1080p Webcam
Disk	128GB	512GB

DESIGN PHASE

ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system

SEQUENCE DIAGRAM

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

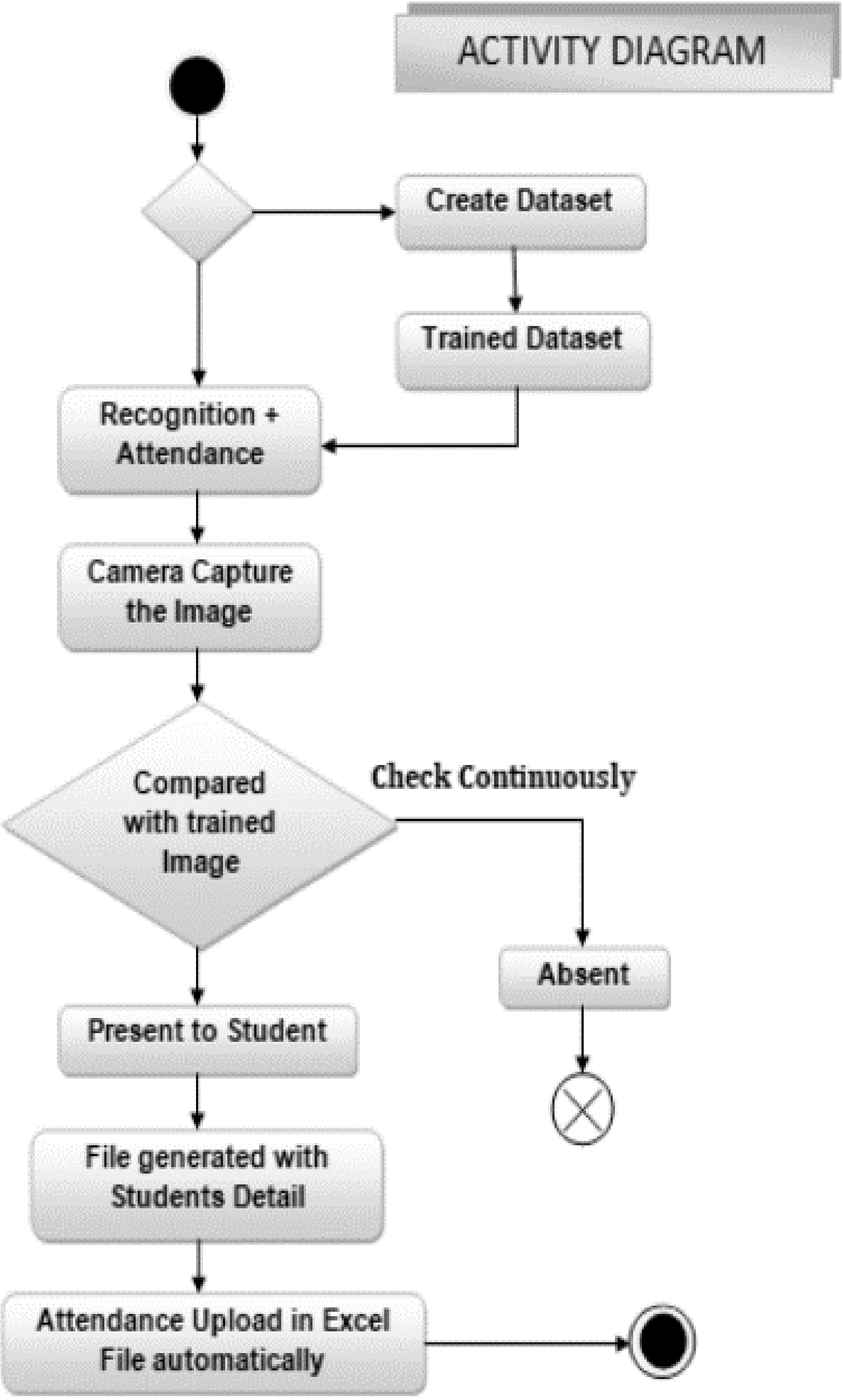
FLOW-CHART DIAGRAM

A flow chart is a graphical or symbolic representation of a process. Each step in the process is represented by a different symbol and contains a short description of the process step. The flow chart symbols are linked together with arrows showing the process flow direction. It represents a workflow.

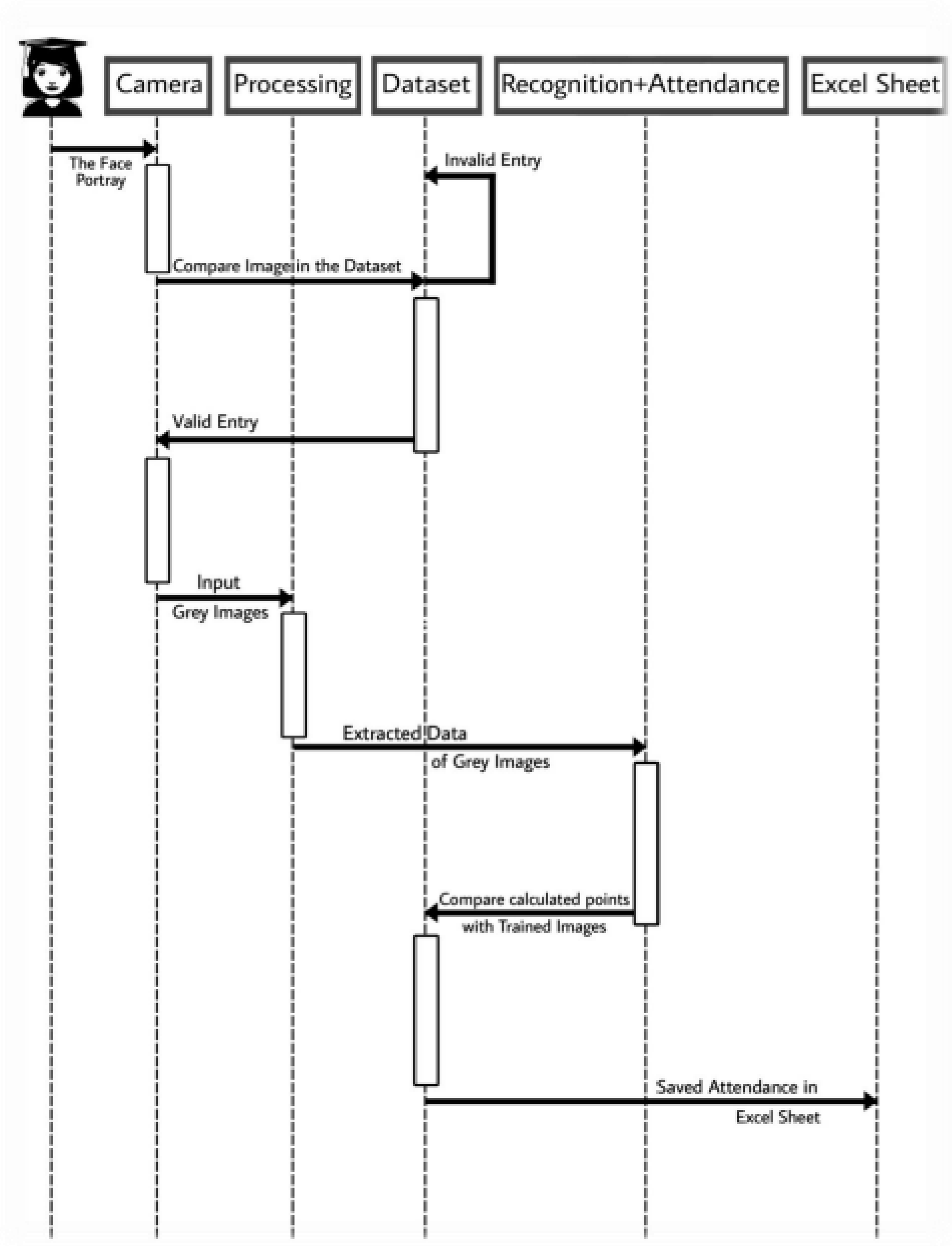
CLASS DIAGRAM

A class diagram in the Unified Modelling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects

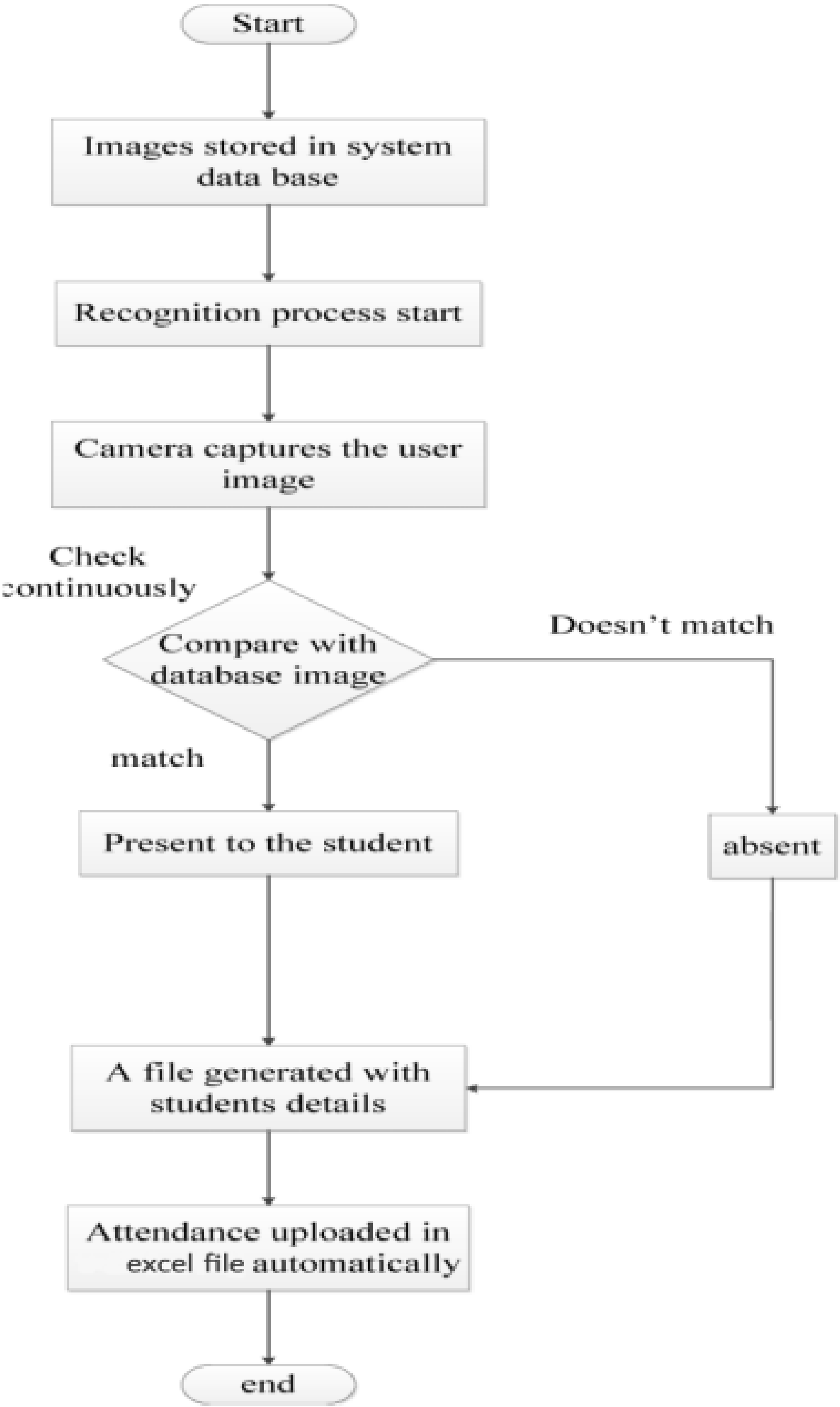
ACTIVITY DIAGRAM



SEQUENCE DIAGRAM

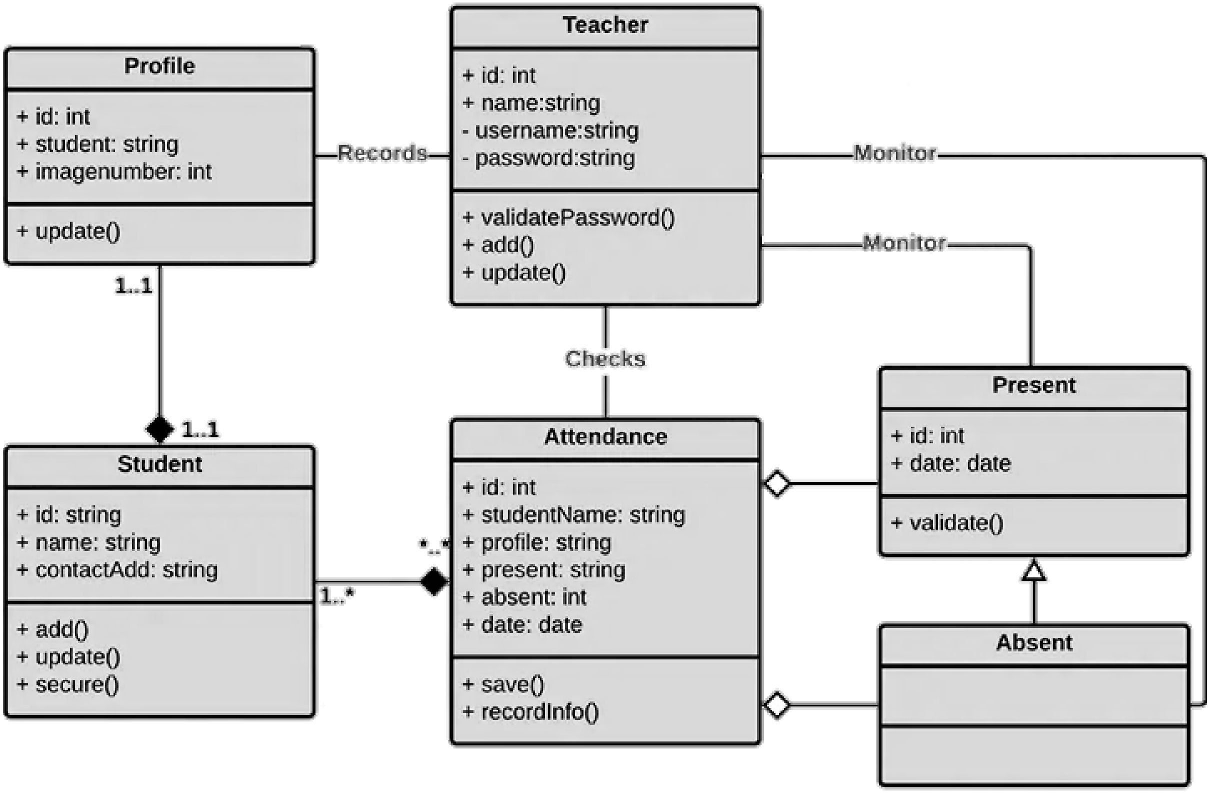


FLOW-CHART DIAGRAM



Start
Images stored in system data base
Recognition process start
Camera captures the user image
Check continuously
Compare with database image
Doesn't match
match
Present to the student
absent
A file generated with students details
Attendance uploaded in excel file automatically
end

CLASS DIAGRAM



CONCLUSION

This system is developed for maintaining the attendance record. The main motive behind developing this system is to eliminate all the drawbacks, which were associated with manual attendance system. The drawbacks ranging from wastage of time and paper, till the proxy issues arising in a class, are eliminated. Hence, desired results with user-friendly interface is expected in the future, from the system. The efficiency of the system could also be increased by integrating various steps and techniques in the future developing stages of the system.

FUTURE SCOPE

The research work has implemented a face recognition system by using PCA which is eigenvector based multivariate analyses. Often, its operation can be thought of as revealing the internal structure of the data in a way which best explains the variance in the data. By implementing PCA the proposed Face Recognition System supplies the user with a lower dimensional picture, a "shadow" of this object when viewed from its most informative viewpoint.

The algorithm has been tested with multiple students in the scene and also captured faces at different angles in the scene. The algorithm delivers quite good results but there is a room to improve the algorithm performance in case of large number of students and also in case of faces captured in a dark environment, so proposed system can be extended in the future to cover this aspect. The efficiency of the algorithm also can be increased further so there is also a room for future work in this area. This system can be enhanced further in terms of achieving more efficiency by ease of analysis of patterns in the data.