JEPPIAAR
ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

III YEAR B.E. – VI SEM

ACADEMIC YEAR (2023 -24 EVEN SEM)

Name :

Register Number :

Subject Name :

Subject Code :

Batch :

JEPPIAAR
ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

This is a Bonafide Record Work of

____

Register No __________________________submitted for the Anna University Practical

Examination held on_____________ **CCS356 OBJECT ORIENTED SOFTWARE**

ENGINEERING LABORATORY during the academic year 2023-2024.

Signature of the Lab-In-Charge **Signature of the HOD**

Date: __________ **Internal:**

External:

Vision of Institution

To build Jeppiaar Engineering College as an Institution of Academic Excellence in Technical education and Management education and to become a World Class University.

Mission of Institution

M1	To excel in teaching and learning, research and innovation by promoting the principles of scientific analysis and creative thinking
M2	To participate in the production, development and dissemination of knowledge and interact with national and international communities
M3	To equip students with values, ethics and life skills needed to enrich their lives and enable them to meaningfully contribute to the progress of society
M4	To prepare students for higher studies and lifelong learning, enrich them with the practical and entrepreneurial skills necessary to excel as future professionals and contribute to Nation's economy

Program Outcomes (POs)

PO1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis:** Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
PO4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Educational Objectives (PEOs)

PEO1	Apply their technical competence in computer science to solve real world problems, with technical and people leadership.
PEO2	Conduct cutting edge research and develop solutions on problems of social relevance.
PEO3	Work in a business environment, exhibiting team skills, work ethics, adaptability and lifelong learning.

Program Specific Outcomes (PSOs)

Students will be able to

PSO1	Exhibit design and programming skills to build and automate business solutions using cutting edge technologies.
PSO2	Strong theoretical foundation leading to excellence and excitement towards research, t provide elegant solutions to complex problems.
PSO3	Ability to work effectively with various engineering fields as a team to design, build an develop system applications

COURSE OUTCOMES:

CO1	Compare Various Software Development Lifecycle Models.
CO2	Evaluate Project Management approaches as well as cost and schedule estimation strategies.
CO3	Perform Formal Analysis on specifications.

ATTENDANCE MANAGEMENT USING FACE RECOGNITION

AIM:

The aim of this project is to develop an automated attendance management system using face recognition technology to improve efficiency and accuracy in attendance tracking.

PROBLEM ANALYSIS AND PROBLEM PLAN:

The traditional technique of marking the participation uses forms such as the signature of the participants and the checking of the respective identification cards, these two methods in themselves contain many problems starting from the concern of the progress of the learning process, but also the distraction of students during exam sessions. The main problem with the traditional attendance system is that it is subject to manipulation and is prone to human error when entering data.

To avoid these problems, a system based on face recognition has been proposed, which efficiently replaces the traditional system, performing processes in an automated form, which saves time, increases efficiency, and provides flexibility and security throughout the process. of taking notes for the presence within the educational institution.

This system aims to automate the traditional system for receiving presence, in which case the organization will be able to maintain its data at any time and digitally. With the digitalization of the system, we enable better visualization of the data, in which case we could create final reports about student participation.

The basic principles of determining the scope of the system are:

1.Enable enrollment of new students.

2.Provide divisions for different subjects.

3.Save data and display reports.

Guidelines defining the scope:

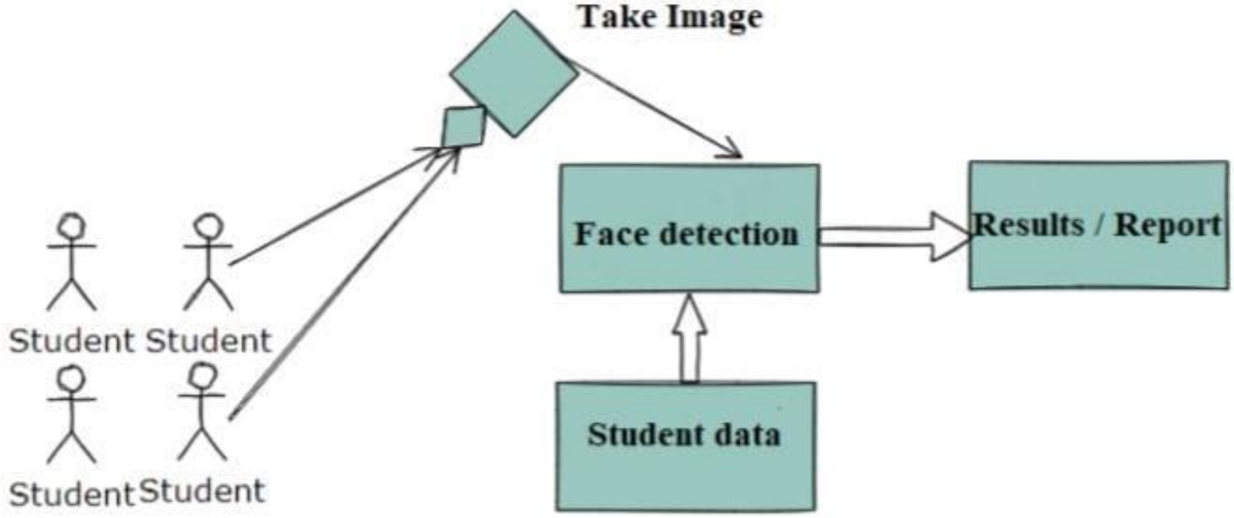
1.The ability of the system to receive data from the registration process.

2.Ability to transfer final reports to a database or .xslx file.

3.Ability to build specific reports depending on the respective subjects.

The basic requirements of the present marking system, whether traditional or built-in, are the same, but the system through face recognition performs the same processes, more efficiently, in a short time, and automatically.

SYSTEM OPERATION SCHEME



SYSTEM DESIGN:

In the system design phase, we determined to use UML diagrams, which provide us with a clear overview of the system even before we start the coding phase, as well as facilitate the coding process by defining functions, roles, classes, and interactions between users.

Using UML techniques, we built:

1. Use Case diagram

2.Activity diagram

3. Status diagram

4. Class diagram

5.Component diagram

6. Interaction diagram

1.USE CASE DIAGRAM:

A UML Use Case Diagram is a visual representation that depicts the interactions between actors (users or external systems) and a system under consideration. It illustrates the various use cases or functionalities that the system provides and how different actors interact with these use cases.

The benefit of use case diagrams is mostly based on communication between the request team and the user group.

A use case specification document should cover the following areas:

1.Actors - participating in and interacting in this use case

2.Preconditions - must be met for the use case to work

3.Unconditional - defines the various states in which the system is expected to be after it is executed.

The Use Case diagram lists the basic events that will occur when the system is executed. It includes all the primary actions that the system must perform.

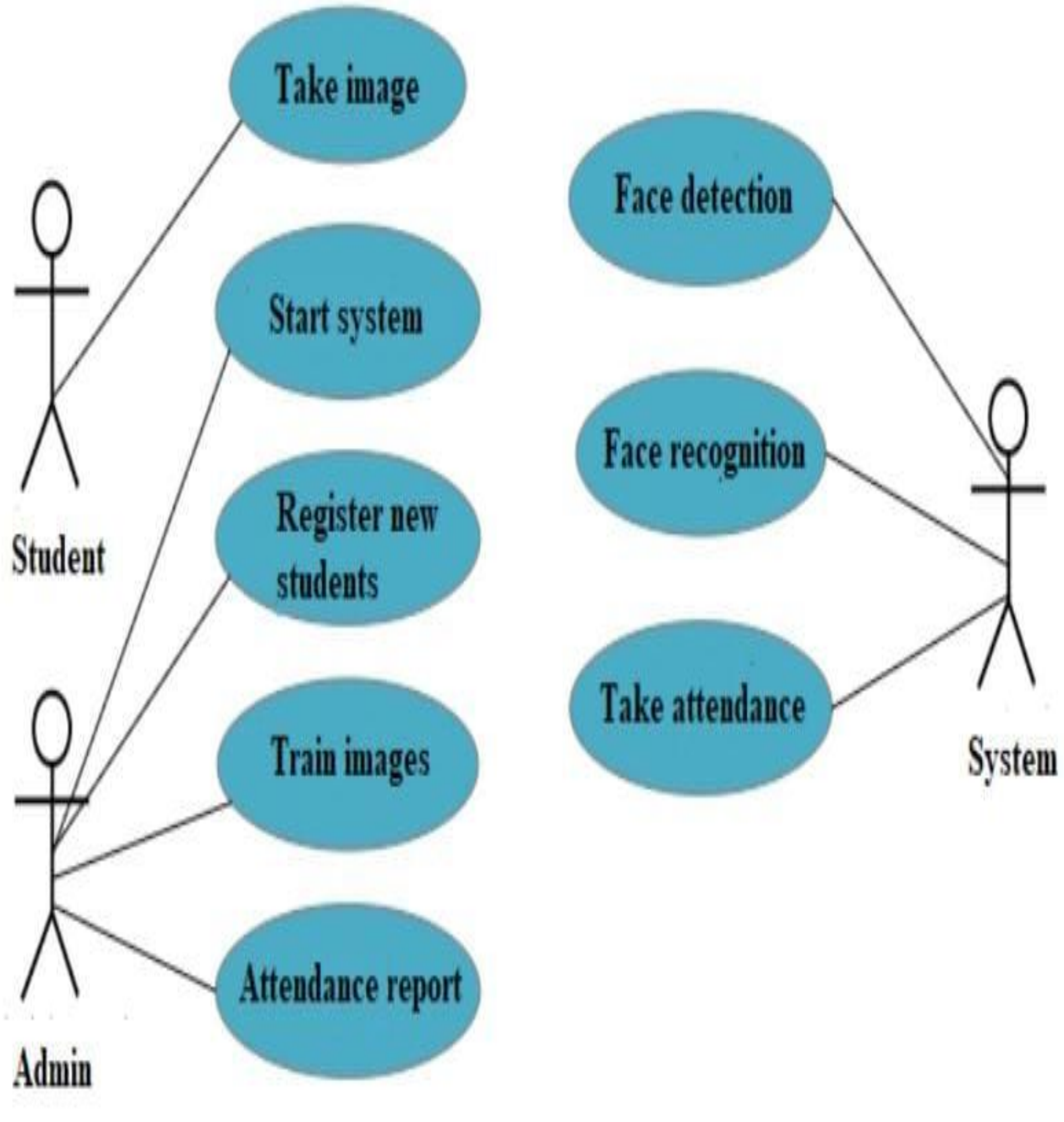
In our system are defined three actors,

1.student.

2. Administrator

3. System

each of them is presented as related to activities or functions which are planned to be executed in the system and which are the responsibility of relevant actors during the operation of the system.

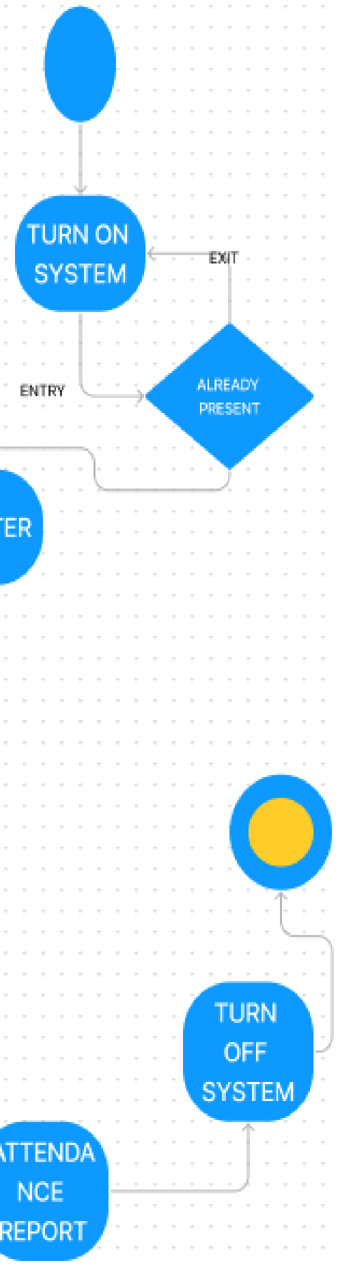

Take image
Face detection
Start system
Student
Face recognition
Register new students
Take attendance
System
Train images
Attendance report
Admin

2.ACTIVITY DIAGRAM:

The activity diagram is a presentation in the form of diagrams keeping a hierarchy of activities. These diagrams are important to understand the way the system works as well as the flow of activities previously stated in the use case diagram. Activities are states of action that automatically switch to another state after the action is completed, but always starting with the "circles" that symbolize the beginning and the end as well as the "arrows" which display the transition from one activity to the next. For our system, we have presented three activity diagrams to create an overview of how the system functions flow.

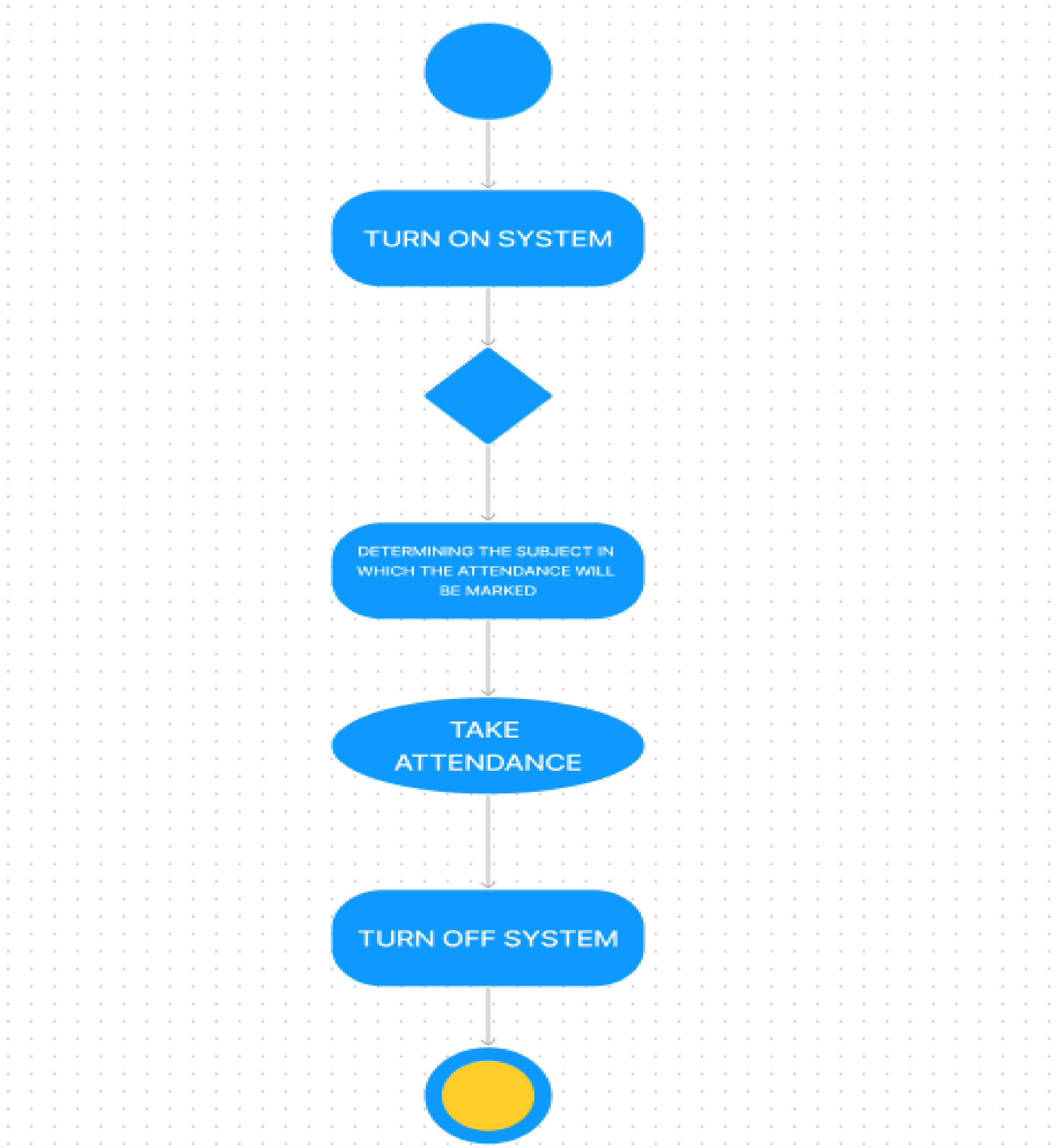
ACTIVITY DIAGRAM 1:

It shows the flow of events in the process when we are dealing with new students, who must first do the face registration in which case the system stores them, then training, and then the presence with these students can be noted. And finally, we have the generation of the report which is the result of the preliminary step respectively the activity of marking the presence.


TURN ON SYSTEM
EXIT
ENTRY
ALREADY PRESENT
REGISTER
TAKING IMAGES
TRAINING PROCESS
TAKE ATTENDA NCE
ATTENDA NCE REPORT
TURN OFF SYSTEM

ACTIVITY DIAGRAM 2:

It shows a simple diagram of the activity in which the operation of the whole system is presented in only 4 basic activities that the system performs.


TURN ON SYSTEM
DETERMINING THE SUBJECT IN WHICH THE ATTENDANCE WILL BE MARKED
TAKE ATTENDANCE
TURN OFF SYSTEM

ACTIVITY DIAGRAM 3:

Facial Features Registration Process:

The activity diagram for the facial features registration section details the systematic process leading to successful face image recognition and the subsequent addition of students to the presence preservation report.

Face Image Recognition Process:

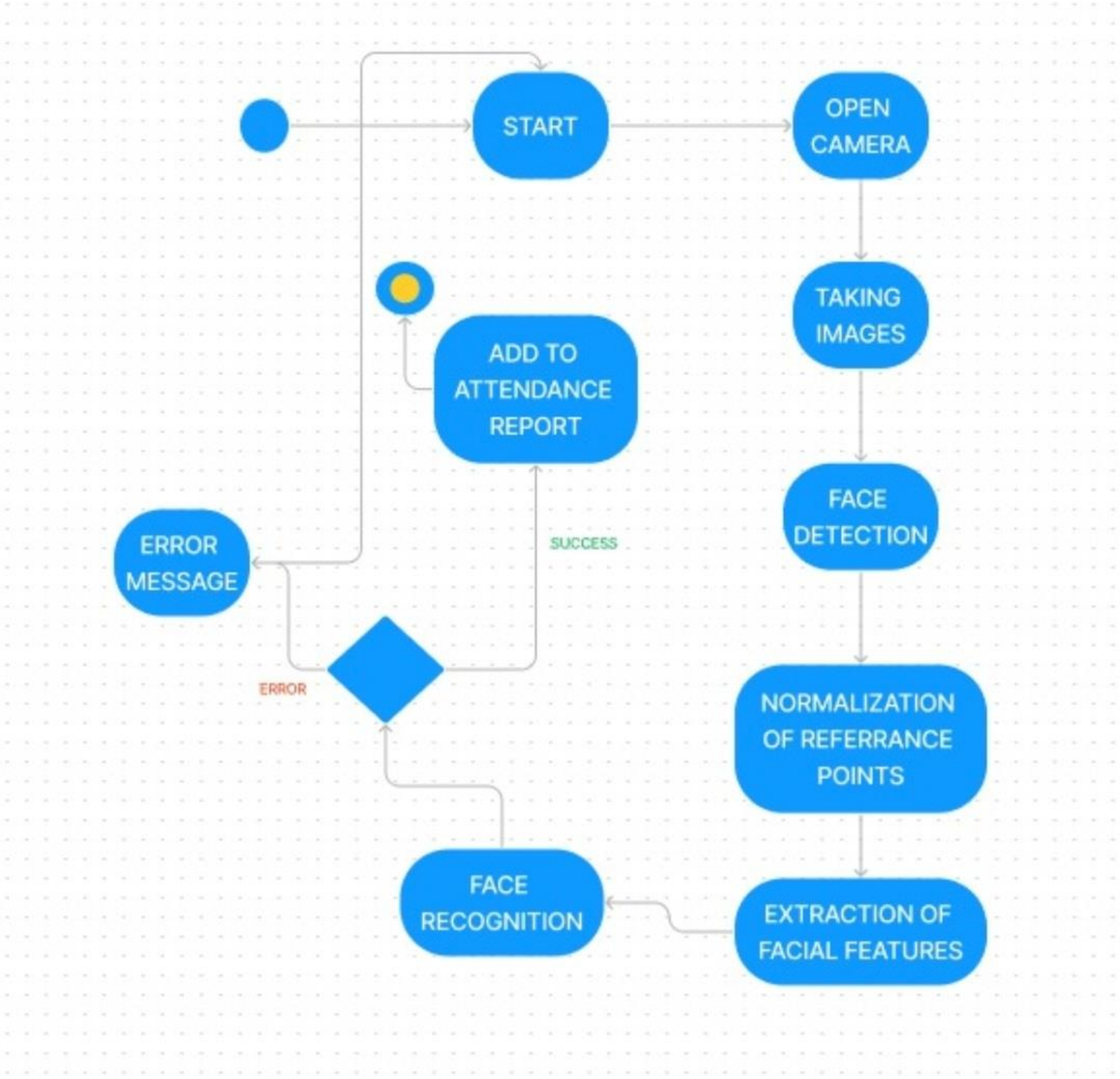
Initially, the system captures and validates unique facial features data, ensuring its accuracy before securely storing it. This validated data is crucial for the subsequent face image recognition process. During face image recognition, a new image is captured and compared against the stored facial features data.

Presence Marking Process (Success):

If a successful match is found, indicating a recognized student, the system proceeds to add them to the presence preservation report, updating their presence status accordingly.

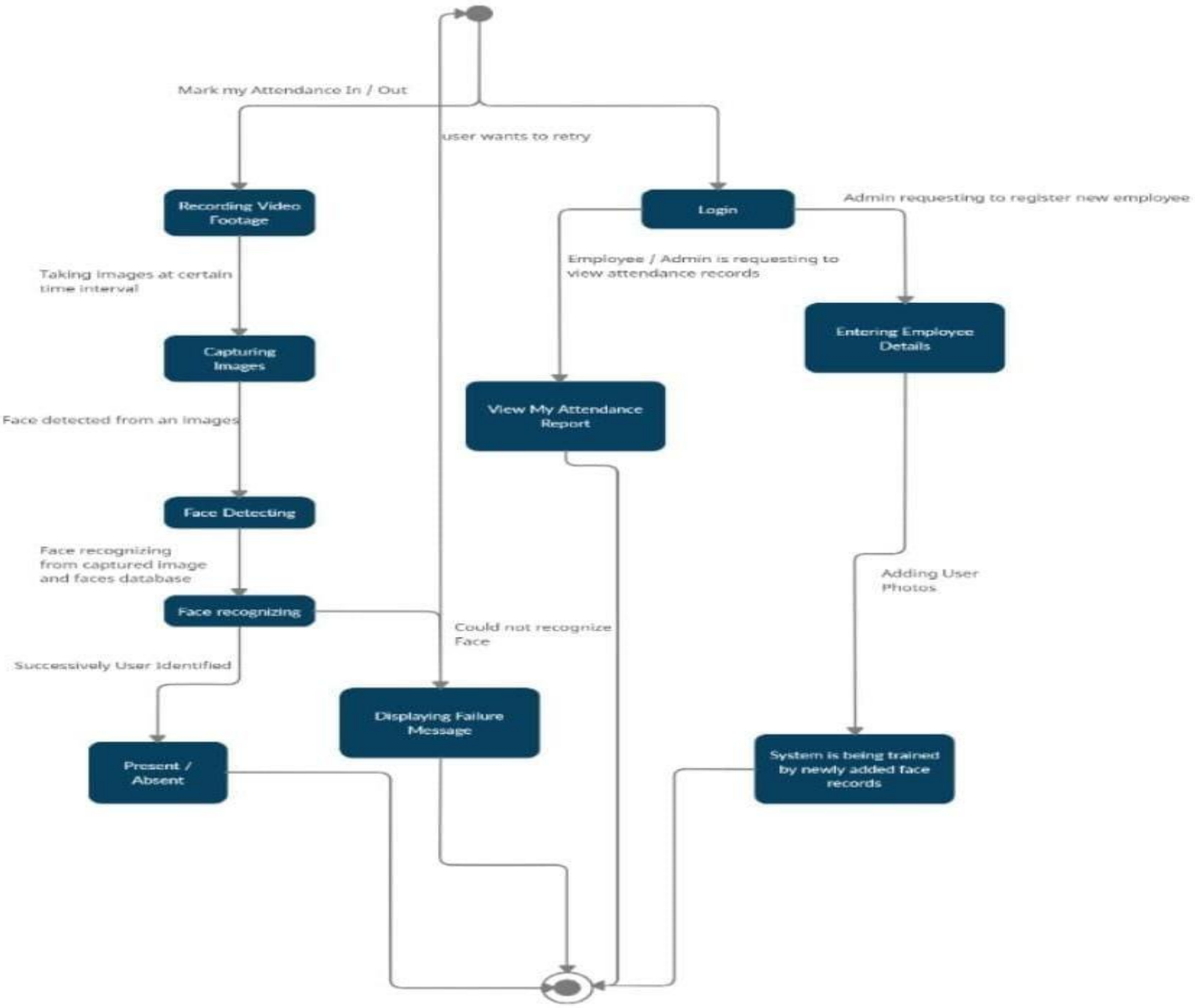
Error Handling (Failure):

However, if the face image recognition process fails to find a match or encounters an error, an error message is promptly displayed, elucidating the reason for the failure. This message serves to guide corrective actions, such as re-capturing the image or verifying the registered facial features data, to ensure the accuracy and reliability of the presence marking error.


START
OPEN CAMERA
TAKING IMAGES
FACE DETECTION
NORMALIZATION OF REFERRANCE POINTS
EXTRACTION OF FACIAL FEATURES
FACE RECOGNITION
ERROR
SUCCESS
ERROR MESSAGE
ADD TO ATTENDANCE REPORT

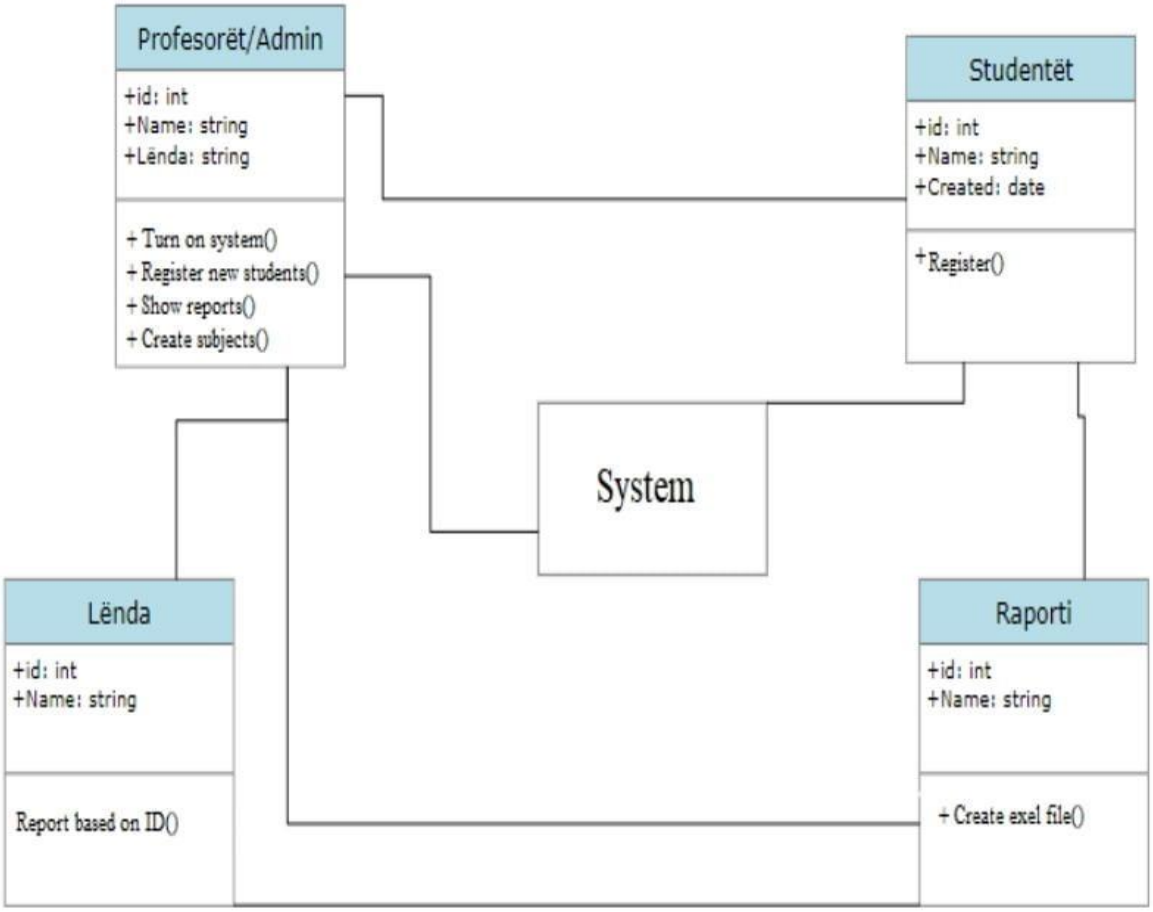
3.STATE DIAGRAM:

The state diagram helps us to better understand the simple functions as well as the complex ones that the system offers. By using the state diagram, we identify the dynamic behavior of the system as a whole or even of the subsystems. With the help of the state diagram, we determine the different states of the participants during the operation with the system.


Mark my Attendance In / Out
user wants to retry
Recording Video Footage
Login
Admin requesting to register new employee
Taking images at certain time interval
Employee / Admin is requesting to view attendance records
Capturing Images
Entering Employee Details
Face detected from an images
View My Attendance Report
Face Detecting
Face recognizing from captured image and faces database
Adding User Photos
Face recognizing
Could not recognize Face
Successively User Identified
Displaying Failure Message
Present / Absent
System is being trained by newly added face records

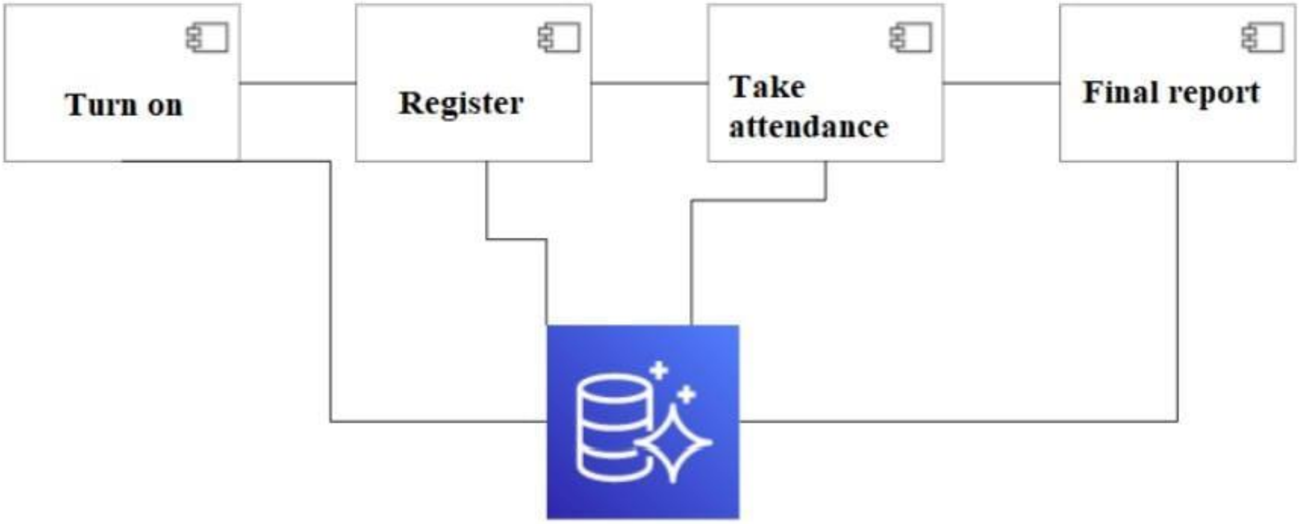
4.CLASS DIAGRAM:

The class diagram describes the structure of the system by presenting classes, attributes, operations, and relationships between objects. The class diagram is the main block of object-oriented modeling and is mainly used for general conceptual modeling of application structure and for detailed model translation modeling in coding. The class diagram can also be used for data modeling.



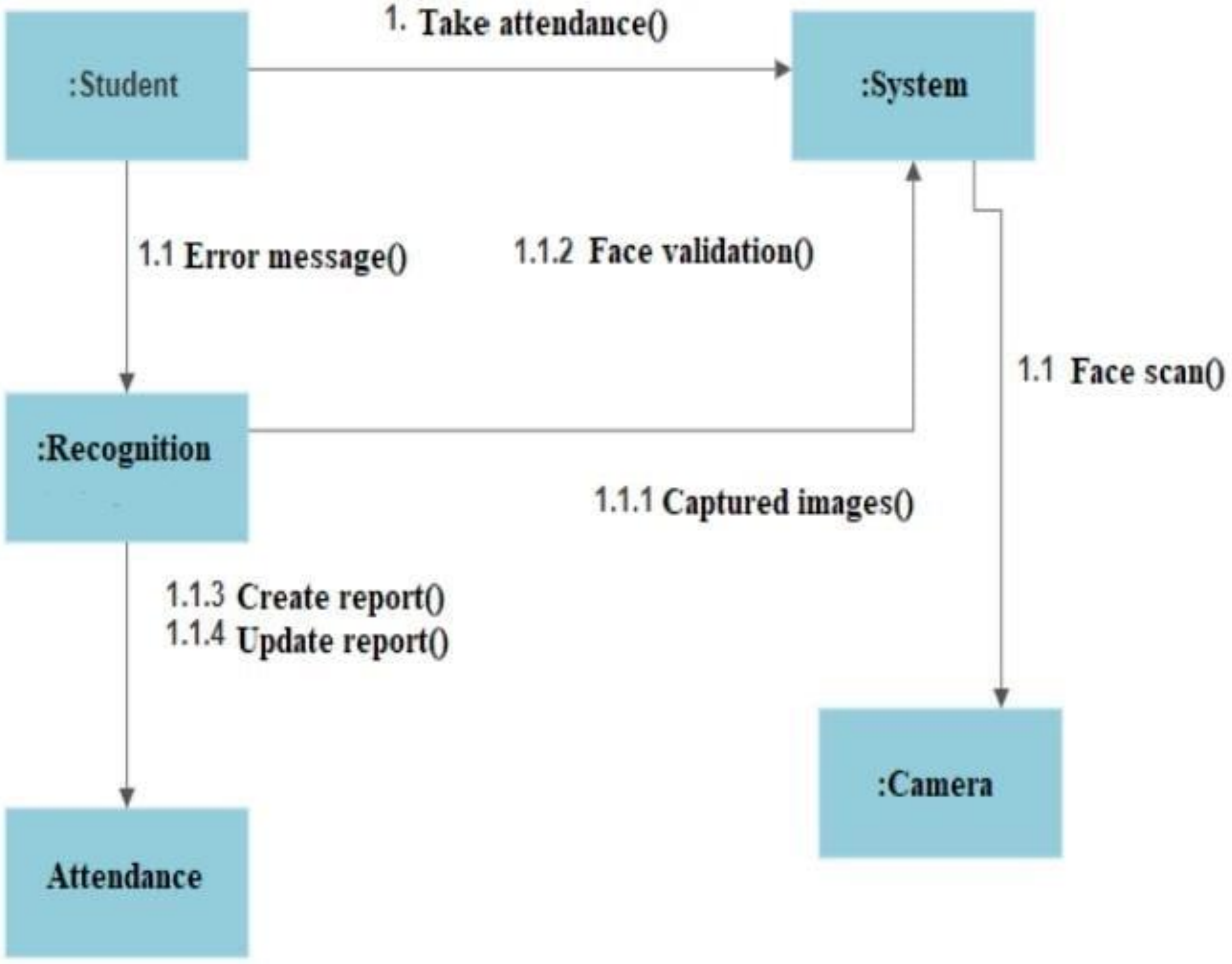
5.COMPONENT DIAGRAM:

The primary difference between a component diagram and other diagrams is that component diagrams represent the implementation perspective of a system. Therefore, the components in a component diagram reflect by grouping different designs of system elements, such as system classes. Firstly, the component must be replaceable, and secondly, the component must provide interfaces to enable other elements to interact and provide the services provided by the component.



6.INTERACTION DIAGRAM:

The interaction diagram, otherwise known as the communication diagram, enables the illustration of connections and interactions between software objects. Otherwise, the interaction diagram represents the use case behavior describing how the set of objects interacts to complete the task. The types of interaction diagrams are sequential diagrams and cooperative diagrams. The interaction diagram, in which we tried to show the communication between the components of the system together with the action sequences which start from the first sequence respectively from the "presence note".



SYSTEM REQUIREMENTS:

1.HARDWARE REQUIREMENTS:

1.1 COMPUTER

- The system requires computers or servers to host the software components.
- Specifications of the computers/servers depend on factors like the number of users, concurrent requests, and database size.
- Recommended hardware includes multi-core processors, sufficient RAM (4GB or more), and ample storage space (at least 100GB).

1.2 CAMERAS FOR CAPTURING FACIAL IMAGES:

- High-quality cameras are essential for accurate face detection and recognition.
- Cameras with a resolution of at least 720p or higher are recommended to capture clear facial images.
- The type of camera (webcam, IP camera, etc.) depends on the deployment environment (desktop, classroom, entrance gate, etc.).

2.SOFTWARE REQUIREMENTS:

2.1 OPERATING SYSTEM:

- The system can run on either Linux or Windows server operating systems.
- Linux distributions like Ubuntu, CentOS, or Debian are preferred for their stability, security, and compatibility with open-source software.
- Windows Server editions provide a user-friendly environment and support for Microsoft technologies.

2.2 PROGRAMMING LANGUAGES:

- Python is used for backend development due to its simplicity, readability, and extensive libraries for data processing and machine learning.

- HTML, CSS, and JavaScript are used for front-end development to create the user interface (UI) and user experience (UX).

- Python frameworks like Django or Flask can be utilized for web development, providing features like URL routing, form handling, and database integration.

2.3 LIBRARIES

- OpenCV (Open-Source Computer Vision Library) is a popular choice for face recognition algorithms, providing tools for image processing, feature detection, and pattern recognition.

- DLib (DLib C++ Library) offers facial landmark detection, face alignment, and machine learning capabilities for face recognition tasks.

TESTING:

During the development of the system from the first stages, unit testing is constantly practiced, which means testing small pieces of code, but which is important to detect any failures or errors that may occur in the system. After successfully passing the unit tests, the system finally went through the Integration testing, after the process of attaching the components, in which case the functionality of all components of the system was tested, and throughout this testing process, errors were identified, and then it has been intervened and improved and so the system is now fully functional by meeting those requirements which I have identified since the analysis stage.

SOURCE CODE:

```
import cv2
import numpy as np
import face_recognition as face_rec
import os
from datetime import datetime


def resize(img, size):
    if img is None:
        return None
    width = int(img.shape[1] * size)
    height = int(img.shape[0] * size)
    dimensions = (width, height)
    return cv2.resize(img, dimensions, interpolation=cv2.INTER_AREA)


path = 'C:/Users/USER/PycharmProjects/final/images'
studentImg =[]
studentName =[]
myList = os.listdir(path)
for cl in myList:
    curimg = cv2.imread(f'{path}/{cl}')
    if curimg is none:
```

```
 Print(f’Error loading image: {path}/{cl}”)
 continue  # Skip this image and proceed to the next one
    studentImg.append(curimg)
    studentName.append(os.path.splitext(cl)[0])

def findEncoding(images):
    imgEncodings = []
    for img in images:
        img_resized = resize(img, 0.50)
        if img_resized is None:
            print("Error resizing image")
            continue  # Skip this image and proceed to the next one
        img_rgb = cv2.cvtColor(img_resized, cv2.COLOR_BGR2RGB)
        encodeimg = face_rec.face_encodings(img_rgb)[0]
        imgEncodings.append(encodeimg)
    return imgEncodings

def MarkAttendance(name):
    with open('attendancesmart.csv', 'a+') as f:
        myDataList = f.readlines()
        nameList = [line.split(',')[0].strip() for line in myDataList]
    if name not in nameList:
            now = datetime.now()
            timestr = now.strftime('%H:%M')
            f.write(f'\n{name}, {timestr}')

EncodeList = findEncoding(studentImg)


vid=cv2.VideoCapture(0)
while True:
    success, frame = vid.read()
    if not success:
        print("Failed to capture video frame")
        break  # Break out of the loop if video capture fails
    Smaller_frames = cv2.resize(frame, (0, 0), None, fx=0.25, fy=0.25)
    facesInFrame=face_rec.face_locations(Smaller_frames)
    encodeFacesInFrame = face_rec.face_encodings(Smaller_frames, facesInFrame)

    for encodeFace, faceLoc in zip(encodeFacesInFrame, facesInFrame):
        matches = face_rec.compare_faces(EncodeList, encodeFace)
        facedis = face_rec.face_distance(EncodeList, encodeFace)
        print(facedis)
        matchIndex = np.argmin(facedis)
```

```
      if matches[matchIndex]:
  name = studentName[matchIndex].upper()
         y1, x2, y2, x1 = faceLoc
y1, x2, y2, x1 = y1*4, x2*4, y2*4, x1*4
cv2.rectangle(frame,(x1,y1),(x2,y2),(0,255,0),3)
         cv2.rectangle(frame,(x1,y2-25),(x2,y2),(0,255,0), cv2.FILLED)
         cv2.putText(frame, name, (x1+6, y2-6), cv2.FONT_HERSHEY_COMPLEX,
1, (255,255,255), 2)
         MarkAttendance(name)
         cv2.imshow('video',frame)
         cv2.waitKey(1)
```

CONCLUSION:

The Attendance Management System using Face Recognition revolutionizes attendance tracking with advanced technology, seamless integration, and robust security measures . It enhances accuracy and results in user friendly software product that minimizes manual intervention.