



Face Recognition System for Examinations

**A dissertation submitted for the Degree of Master of
Information Technology**

**D.R.N Wickrama
University of Colombo School of Computing
2019**



Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: D.R.N Wickrama

Registration Number: 2016/MIT/087

Index Number: 16550876

Signature:

Date:

This is to certify that this thesis is based on the work of

Ms. D.R.N Wickrama

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Prof. N.D Kodikara

Signature:

Date:

Abstract

Examination or the assessment is the ultimate outcome of the education. Each and every curriculum must be concluded with any kind of assessment. Personal identification in the examination systems of Sri Lanka has been performed by getting the candidate's identity card and check the relevant details of the candidate. The main issue arises in this scenario is the unique identification of each candidate. This is the major issue arises during the examination period faced by the examiners and their officers in charge of the examinations department. It is a hectic method to use the above-mentioned manual method in recognizing the candidates which is not applicable nowadays.

In this dissertation it is suggested the unique identification of the candidate by a system and giving out the details of the candidate without performing manually. The preview image of the candidate has been output through the system with the candidate index number, name and any other relevant details. And also, the exam start time and end time will be noted at the time of recognition. The future system would recognize each student and produce the details such as name in full, Index number required for the exam. Even though they have produced their Identity cards or any other materials to prove their identity there is a possibility to pose a fake candidate on behalf of the real candidate. It is essential to recognize their identity without any failure using a bio metric pattern.

This project was implemented as the pilot project for the examination systems in candidate face recognizing process, chosen St. Peter's College Colombo 4 as the client.

Face recognition system initially detects the candidate's face. The detected face later trains and stores inside the file folder. Training process is the most crucial event in this system. If the training process is not successful face detection will fail to perform. During detection time the system allows the candidates to insert the student name and his/her index number of the examination. Finally, the candidate will be able to scan the face prior to the exam. If the recognition is successful, student name and index number will display during the recognition process and a confirmation quote 'Your Face Recognized' also displays. Otherwise it displays 'Your Face not Recognized'.

OpenCV/Python suite is being used for developing this face recognition system. The object oriented and structural patterns have been used. CV2, PIL, NumPy, Pickle, OS libraries of image processing used in the face detection, face training and face recognition processes. SQLite is the database management system, stored all data of the candidates. The face recognition system developed from scratch as the stand-alone system in the windows environment.

Around fifty to sixty users participated in the evaluation process. The goals of the project achieved successfully. According to the User evaluation feedback, implemented face recognition system for examinations fulfills the requirements of the examination management system in Sri Lanka.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Professor N.D Kodikara, my supervisor for the invaluable guidance, encouragement and precious support given throughout the course. More importantly the valuable advice given to me to drive the project more successful.

Furthermore, I would like to express my sincere gratitude to staff of University of Colombo School of Computing for offering this masters degree program for seeking knowledge in the IT industry, from the beginning of the program.

Also take this opportunity to express a deep sense of gratitude to my client St Peter's College Colombo 4 for the guidance, assistance given during the all-time period starting from data gathering process. Especially St Peter's College Principal Rev. Fr. Trevor Martin for giving this opportunity to carry out this pilot project, Also Primary IT section Mater in charge Mr. K.W Iranga Shashipriya and IT teacher Ms. Shashikala Kodisinghe for giving fullest encourage in conducting data gathering process from students.

Further express my sincere gratitude to Mr. Susantha Weerakoon by providing advice, guidance and valuable support to make this effort success.

Last, but not least, express my heartfelt gratitude to my loving parents, and all family members for giving fullest freedom to carry out my project work throughout the year.

Contents

Title page	i
Declaration	ii
Abstract	iii
Acknowledgements	iv
Contents.....	v
List of Figures	vii
List of Tables.....	viii
List of Abbreviations	ix
1.Introduction	1
1.1 Problem Statement	1
1.2 Motivation.....	1
1.3 Goals and Objectives	1
1.4 Methodology	2
1.5 Scope.....	2
1.6 Chapter Breakdown	3
2. Background	4
2.1 Background Introduction	4
2.2 An Analysis	4
2.3 A review of Similar Systems	6
2.4 Implementation tools and technologies	8
2.5 A comparison of alternative design strategies	8
3. Methodology.....	9
3.1 Database Table Structure	9
3.2 File Organization	10
3.3 Backup Strategy Design.....	10
3.4 Design of Audit Trails	10
3.5 Security Scheme Design	11
3.6 Structure Chart.....	12
3.7 Diagrams	13
3.8 Test Plan	16
3.9 Tools	16
3.10 Hardware/Software	16
3.11 Code Structure	17

4. User Evaluation and testing	19
4.1 Introduction	19
4.2 Testing Stages	19
4.3 Test Plan and Test Cases	20
4.4 Acceptance Testing	25
4.5 Problems Encountered	25
4.6 Lessons Learnt	25
5. Conclusion	26
5.1 Introduction	26
5.2 Summary of the result	26
5.3 Future Improvements	27
References	28
APPENDIX A-DESIGN DOCUMENTATION	29
APPENDIX B– USER DOCUMENTATION	33
APPENDIX C– CODE LISTINGS	37
APPENDIX D-CLIENT CERTIFICATION LETTER	49
APPENDIX E-USER EVALUATION FORM	50
APPENDIX F-USER GUIDE	51
GLOSSARY	52

List of Figures

Figure 2.1 Process Modelling -DFD.....	4
Figure 2.2 Data Modelling-ERD	5
Figure 2.3 Structured English.....	5
Figure 3.1 Table Structure	9
Figure 3.2 SQL statement for Table Creation.....	10
Figure 3.3 Data View.....	11
Figure 3.4 Structure Chart	12
Figure 3.5 Class Diagram	13
Figure 3.6 State Diagram	14
Figure 3.7 Sequence Diagram.....	15
Figure 4.1 Welcome Screen.....	21
Figure 4.2 Main Login Screen	21
Figure 4.3 Dataset Creation	22
Figure 4.4 Face Training.....	23
Figure 4.5 Correct Face Recognition.....	24
Figure 4.6 Not Recognized Face.....	24
Figure A.1 Usecase Diagram Dataset Creation	29
Figure A.2 Usecase Diagram Face Training.....	30
Figure A.3 Usecase Diagram Face Recognition.....	31
Figure A.4 Activity Diagram	32
Figure B.1 User Registration Screen	33
Figure B.2 Successful User Registration	33
Figure B.3 User not found Error	34
Figure B.4 Password Error	34
Figure B.5 Successful User Login	34
Figure B.6 System Main Interface	35
Figure B.7 Student Face Recognition Main Screen.....	36
Figure B.8 Scanned Student Details	36

List of Tables

Table 2.1 Justification of Design Strategies	8
Table 4.1 Login Module	20
Table 4.2 Face Dataset Creation Module.....	21
Table 4.3 Face Training Module.....	22
Table 4.4 Face Recognition Module.....	23
Table 5.1 TP,TN,FP,FN Theory	26
Table A.1 Usecase Description for Dataset Creation	29
Table A.2 Usecase Description for Face Training.....	30
Table A.3 Usecase Description for Face Recognition.....	31

List of Abbreviations

ABIS-Automatic Biometric Identification System

API -Application Programming Interface

DFD – Data Flow Diagram

ERD – Entity Relationship Diagram

LBPH-Local Binary Patterns Histogram

OpenCV-Open Source Computer Vision

OS -Operating Systems

PIL -Python Image Library

PIP-Pip Installs Packages/Pip Installs Python

R&D -Research and Development

SDK -System Development Kit

SQL-Structured query Language

VIP-Very Important Person

Chapter 1

Introduction

1.1 Problem Statement

People are in an innovative world nowadays, almost all the humans are encouraged to do the exams throughout the year. Examinations matter a lot in their life. The way of conducting the examinations may change time to time. For example, online exams are most probably getting the attention of the candidates currently. Even though considerable amendments occur in the examination context the traditional style is still being conducted by the examinations department of Sri Lanka. This is the dissertation produced for the face recognition system for examinations.

The main purpose of this is to identify the students at the examination without any delay. In the current system students should produce their identity card to prove each candidate who sits for the exam. The proposed system would recognize each student and produce the details such as name in full, Index number required for the exam. Even though they have produced their Identity cards or any other materials to prove their identity there is a possibility to do forgeries as another outside person try to pose as real candidate.

1.2 Motivation

Personal identification in the examinations is the major issue arises during the examination period faced by the examiners and their officers in charge at the examinations. Due to this burning problem in recognizing the real user it is hectic method to use manual method to recognize the candidate which is not applicable in the current situation. Considerable amount of examination forgeries has been reporting from years in the current examination system.

Face recognition system for examinations will be an ideal solution to overcome the exam forgeries in examination context. For example, prepare forged documents, posing as an unknown candidate on behalf of the real candidate will be overcome in current examination system. Prior recognition of the candidate is the most crucial event in the examination system. With the aid of this system examiners, invigilators will be able to recognize each candidate precisely.

1.3 Goals and objectives

Goal

The main goal of this project is to successfully recognize the candidates at the examinations and prove their identity without any fault. The following factors illustrates the objectives of the project.

Objectives

- 1) Overcome the examination forgeries such as siting unqualified candidates in the examination.
- 2) Generate the candidates' details such as name, index no electronically at the examination venue.
- 3) Minimize the examination errors in entering student personal details.
- 4) Improve the efficiency and effectiveness in recognizing candidates.
- 5) Identify each candidate dynamically at the exam Centre.

1.4 Methodology

The system methodology based on this project is qualitative analysis, any academic institute can be used this system next year for recognizing students at the exam centers. Pre scan the student face before the exam and train the student face and allow to recognize at the exam day within the specified time range are acceptable functionalities of the current project. On the exam day the candidate face is being scanned and checked whether it will be accepted and produced the relevant student details. If the student face is being recognized the pre scanning and face training process is successful.

1.5 Scope

This project is carried out as a pilot project. Face recognition system identifies each school student and will be able to list each student detail individually. The database is developed to match the captured student details. The captured image should give the clear identification of the student. This project is initiated by targeting any national examinations conducted by the Department of Examinations in Sri Lanka. Each candidate's face is being scanned and recognized and produce relevant details of the candidate such as name in full, index no etc.,

The scope of the project listed below.

1) Register the students with necessary details of them suits for the examination matter.

Assumptions:

- Each and every student has a student registration number to identify them individually.
- Each student must enter his full name.
- Only school students will be used to do the relevant processing.

2) Record the student details such as name, Date of birth, exam time details etc.,

3) Preview the student details along with the captured image.

1.6 Chapter breakdown

The dissertation document contains the overall details of the project is being carried out. The remaining chapters of this dissertation consists as follows.

I. Background

This includes the background of the implementing the project. An analysis, a review of similar systems, comparison of design strategies can be included in this background chapter.

II. Methodology

The design of your solution described in this chapter. The reader will be able to understand the structure of the system. For example, database structure organization, backup strategy plan, diagrams that want to display.

III. User Evaluation and Testing

In this chapter, the reader will understand about the critical discussion evaluating the results. It concerns whether the project objectives were satisfied and if not reasons for that.

IV. Conclusion

This chapter includes all summarize details of the project. Also, during this time, it identifies the deficiencies of the final product. This should also highlight the project improvements could be made in future.

Chapter 2

Background

2.1 Background Introduction

Precise and accurate Personal identification in the examinations is the major issue arises during the examination period faced by the examiners and their officers in charge at the examinations. It may be a G.C.E Ordinary level examination, Advanced level examination or any other vocational or competitive examination. When consider about the background of this project, personal identification is a manual method have been following from years. In this manual system a student is being identified by using an identity card. The image pasted in the card is being checked with the real student present at the examination. This is becoming an obsolete method in examination system where as candidates are smart enough to carry out misbehavior activities nowadays. Examination misconducts would be easily revealed by using this automated face recognition system.

2.2 An Analysis

2.2.1 Functional requirements

1. Identify the candidate and produce his/her details such as name, index number to the examiner.
2. Store required candidates' details electronically.
3. Check the time in and out details at the examination center.

2.2.2 Non-Functional requirements

1. Produce the accurate student details.
2. Efficiently and effectively identify each student.
3. Perform faster student face recognition.

2.2.3 Existing system diagrammatic representation

2.2.3.1 Process Modelling- DFD-Gane & Sarson

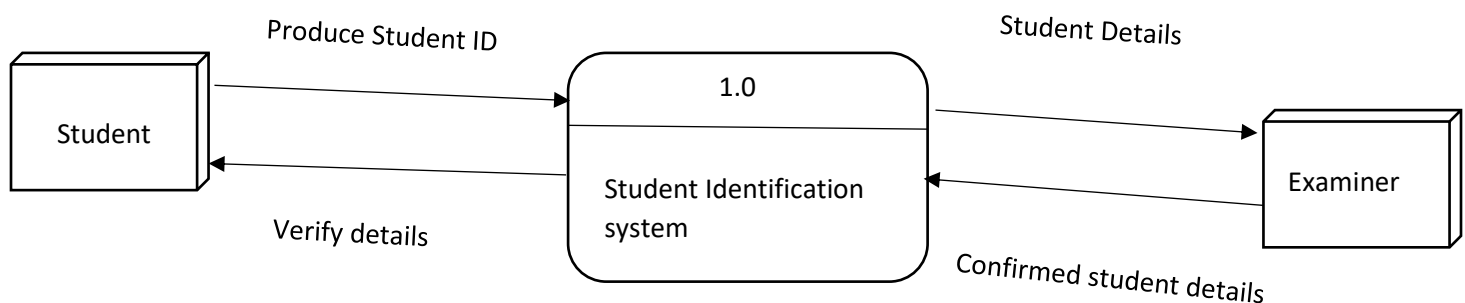


Figure 2.1: Process Modelling- DFD

Student will produce his/her Identity card at the examination where the examiner verifies the details against the student file. Once the student produces the Identity card examiner checks the relevant student details and confirms and verifies them. If the student details are matched with the stored details the student will be able to enter into the examination premises. Student ID and confirmed student details data flows direct to the system while verify details and student details direct out from the system.

2.2.3.2 Data Modelling- ERD

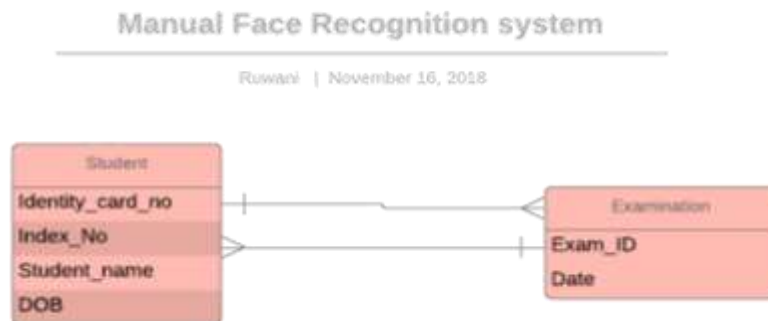


Figure 2.2: Data Modelling- ERD

Student entity uniquely identifies by the Identity card no other than that Index Number, Student name and Date of birth would be used in the student entity.

Examination entity uniquely identifies by the examination Id. Date of the examination is the attribute of this entity.

In this ERD model one student participates for many examinations. For a given examination many students are participating. Therefore, Student Examination entities have many to many relationship.

2.2.3.3 Structured English

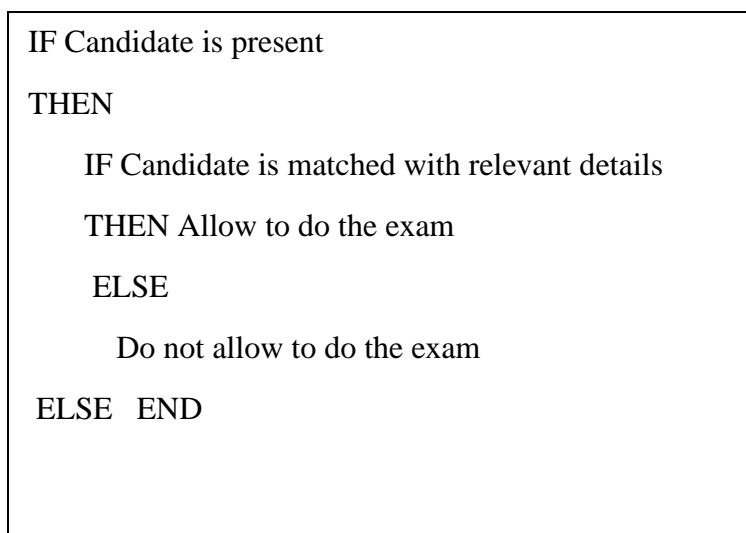


Figure 2.3: Structured English

Simple texts have been used to describe the examination scenario. IF, THEN, ELSE, END keywords used to represent the candidate verification process at the examination premises. If student has been verified then allow to do the exam otherwise not allow to do the exam.

2.3 A review of similar systems

2.3.1- DERMALOGs “Face Matching”[1]

DERMALOGs “Face Matching” is the fastest software for biometric Face Matching on the market. The speed is at 100 million matches per second on a single blade. Accompanied by a low failure rate the confirmed working speed is ahead of the competition.

DERMALOG face recognition technology combines facial feature biometrics. The technology is nonintrusive and widely accepted. Especially in combination with identification cards, it is able to confirm that the person submitting information matches the provided identification document. In criminal investigations, DERMALOG face recognition is recognized for user-friendliness and high verification rates. It runs on different operating systems.

Operating reliable even under varying light conditions including all types of persons – age, gender, size and skin color have no negative impact on the performance of DERMALOG’s Face Solution. Face Biometrics raise security levels at monitoring systems, access control, mobile solutions and border control: Face matching solutions by DERMALOG are part of border control in several countries.

Main Advantages of DERMALOG Face recognition

- ✚ Easy to integrate into ABIS
- ✚ Combine with high speed with high degree of accuracy
- ✚ Live detection of the feature points on the face.
- ✚ Automatic face cropping
- ✚ Possibility to estimate facial expression
- ✚ Outstanding matching performance
- ✚ Liveness detection to prevent spoof attacks
- ✚ Live passport image compliance checks of facial features
- ✚ Robust and fast tracking of multiple faces of videos
- ✚ Supports all common image file formats

2.3.2- Facefirst Software[2]

Protect access to sensitive area

Establish a virtual, biometric-enforced perimeter around military facilities, government buildings, corporate R&D areas and other locations.

Seamless Convenience

Face recognition is as fast as it is accurate, instantly validating identity in order to protect access to sensitive areas.

Identity Protection

Guarantee that only authorized individuals gain access to secure locations, while requiring far less personal information than other forms of identification.

Enhanced Security

Provide an additional layer of security beyond keys and cards.

Crowd management software for exclusive events

Quickly and securely verify attendees at any event or location without expanded queues.

Prioritize guests and VIPs

Instantly identify special visitors and give VIPs the attention they expect and deserve.

Optimize crowd control

Face recognition streamlines entry to conferences and events, reducing queue times and making it easier to provide customer service.

Identify a face in a crowd

FaceFirst's visual identification system is crowd-proof, making it easy to identify and admit many people efficiently and at the same time.

Face recognition Integration

With the FaceFirst platform, imagination is your only constraint.

Innovate with Facefirst

The FaceFirst SDK and robust APIs allow you to easily integrate face recognition with a variety of third-party access control systems, creating complete identification and authenticating solutions.

Replace password

Replace traditional passwords with secure and convenient biometrics templates, or use face recognition as a secondary form of authentication.

Integrate with hardware

Integrate the FaceFirst platform with third-party systems, devices, public safety programs and much more.

FaceFirst is a global patented facial recognition software platform designed to be scalable, fast and accurate while maintaining the highest levels of security, privacy and convenience. Using artificial intelligence and machine learning, the FaceFirst system can be used as a one-to-one access control solution, or can be used to establish a physical perimeter to keep out unwanted or potentially dangerous people. The FaceFirst API allows for easy integration with a variety of access systems.

2.4 Implementation tools and technologies.

The Proposed system will be used by administrative staff who are not familiar with programming. Therefore, system should be understandable and user-friendly working environment. Object oriented working environment used for developments. Latest technologies and tools are used for development. Development framework is Microsoft Visual Studio OpenCV [3]. The language used in here is OpenCV/ Python.

The database is MYSQL database management system. Development pattern has the Object-Oriented technologies.

2.5 A comparison of alternative design strategies.

This system is a face recognition system therefore suitable development is developing from scratch. The hardware architecture is stand-alone personal computer. In this implementation the appropriate operating system is Windows Operating System.

Since this project is facial recognition system it will be a standalone computer operating system. Windows operating system can be ideal for this project. Client server or web-based model is not suitable for this kind of project. Client server model is costlier than standalone computer model.

Justification of Design Strategies

	Developing from Scratch	Using Open Source component
Development strategy	✓	-
	Stand-alone model	Client Server model
Hardware Environment	✓	-
	Windows	Linux
System Software/OS	✓	-

Table 2.1: Justification of Design Strategies

Face recognition system does not need to get connected to the network at this scenario. The software was developed from the beginning without using any kind of pre-existing software component. Windows operating system is much easier to handle the applications than any other operating systems. User friendliness, much more hands-on experience encouraged to select the windows operating system.

Chapter 3

Methodology

3.1 Database Table Structure

SQLite is a software library that provides a relational database management system. The lite in SQLite means light weight in terms of setup, database administration, and required resource.

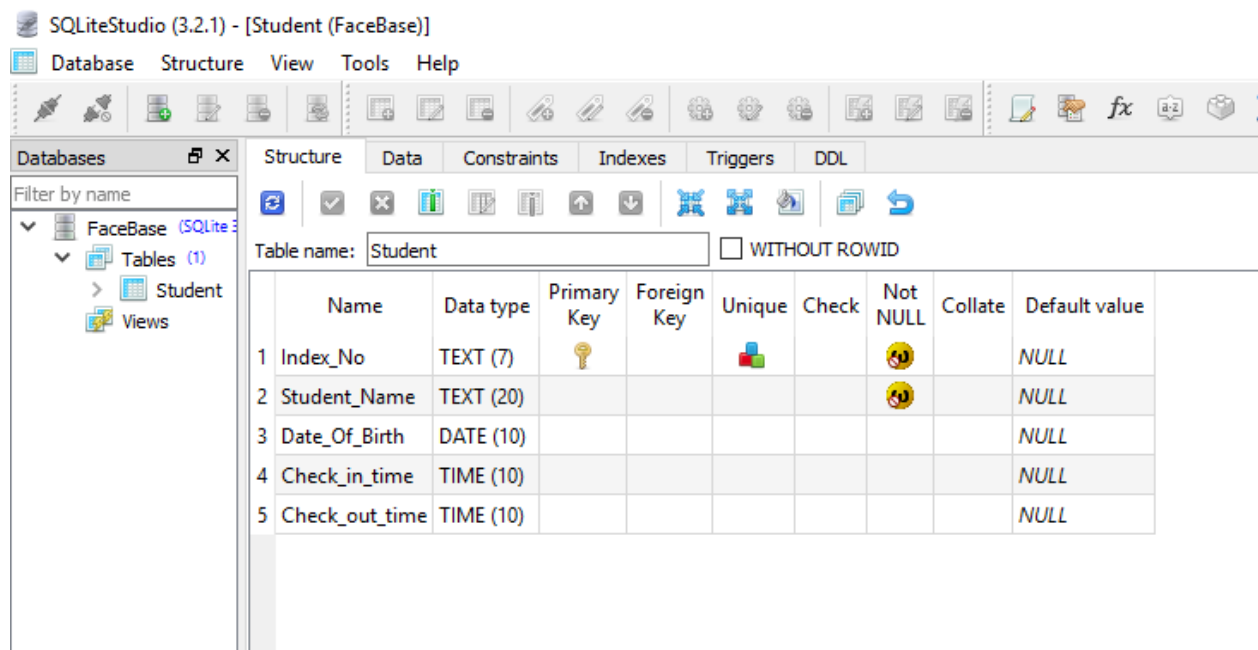


Figure 3.1: Database Table Structure

Version-SQLite Studio 3.2.1

SQLite has the following noticeable features:

- ✓ self-contained
- ✓ serverless
- ✓ zero-configuration
- ✓ transactional

SQLite does not require a server to run. SQLite database is integrated with the application that accesses the database. The applications interact with the SQLite database read and write operations are carried out directly from the database files stored on disk.

In SQLite everything is in a table. Here table name is FaceBase.

```
CREATE TABLE Student (  
    Index_No      TEXT (7)  UNIQUE  
                        PRIMARY KEY  
                        NOT NULL,  
    Student_Name  TEXT (20) NOT NULL,  
    Date_Of_Birth DATE (10),  
    Check_in_time TIME (10),  
    Check_out_time TIME (10)  
);
```

Figure 3.2: SQL statement for Table Creation

3.2 File Organization

Most image processing and manipulation techniques can be carried out effectively using two libraries: PIL and OpenCV

PIL, is one of the core libraries for image manipulation in Python. In Python, image processing using OpenCV is implemented using the CV2 and NumPy modules.

3.3 Backup Strategy Design

Backup solution with cloud technology

Cloud based backup is a great solution, the best really. The concept is that you make a backup to a remote destination. There are tons of services out there providing exactly that service, but they are not equally good and equally cheap.

A cloud backup that takes TeraBytes of data (unlimited data) at a low cost, without killing neither local storage nor internet connection.

Backup solution using physical storage

If you have enough capacity on your local disks, you can back up to them or to external USB drives. These backups are fast and convenient and you don't need a network. The downside of local backups is that if the system is destroyed by fire or flood, your backups can be destroyed as well if they are stored in the same location. Also, in many cases, you need to manage these backups on a computer-by-computer basis, which makes it cumbersome for larger environments.

3.4 Design of Audit Trails

A set of student records can be added to the database table name 'student' in the 'FaceBase' database. Each student's Name and Index number, date of birth can be displayed mainly along with the recognized image, additional details also can be recorded in the database such as check in and check out time of the examination also.

	Index No	Student Name	Date Of Birth	Check in time	Check out time
1	1	Nehan Dissanayaka	5-9-2008	10.00	12.00
2	2	Yohan Gajasinghe	8-8-2008	10.00	12.00
3	3	Sithula Samaraweera	25-5-2008	10.00	12.00
4	4	Sadew Dias	26-7-2008	10.00	12.00
5	5	Aayodhya	8-9-2008	10.00	12.00
6	6	Avisha Nihen	9-7-2008	10.00	12.00
7	7	Hiruka Boteju	14-8-2008	10.00	12.00
8	8	Devon Medies	1-7-2008	10.00	12.00
9	9	Shenuk Kapuge	25-8-2008	10.00	12.00
10	10	Aanjalo Janith	9-9-2008	10.00	12.00
11	11	Monash Senevirathna	14-6-2008	12.00	2.00
12	12	Aaden Juriansz	4-4-2008	12.00	2.00
13	13	Jonathan Herath	15-5-2008	12.00	2.00
14	14	Gauditha Aloysius	6-9-2008	12.00	2.00
15	15	Suveen Dassanayake	1-1-2009	12.00	2.00
16	16	Inuk Kularatna	14-1-2009	12.00	2.00
17	17	Pragarshan	28-1-2009	12.00	2.00
18	18	Aaron Jayathilake	8-6-2008	12.00	2.00
19	19	Steve Fernando	7-6-2008	12.00	2.00
20	20	Dinan Jayasinghe	11-1-2009	8.00	10.00
21	21	Thisath Jayasundara	1-11-2008	8.00	10.00
22	22	Savin	23-11-2008	8.00	10.00
23	23	Rasidu Siriwardena	12-12-2008	8.00	10.00
24	24	Chenura	7-9-2008	8.00	10.00
25	25	Naveen Shalinda	17-3-2008	8.00	10.00

Figure 3.3: Data view

3.5 Security scheme design

Key Aspects of Software Security

- Integrity
- Confidentiality
- Availability

Integrity within a system is the concept of ensuring authorized users can only manipulate information through authorized methods and procedures. To overcome this, the administrator is there with system access rights.

Confidentiality within a system is the concept of preventing unauthorized access to specific information or tools. In a perfect world the knowledge of the existence of confidential information/tools would be unknown to all those who do not have access. When this concept is applied within the context of an application only the authorized information/tools will be available.

Availability within a system is the concept of authorized users being able to access the system.

3.6 Structure Chart

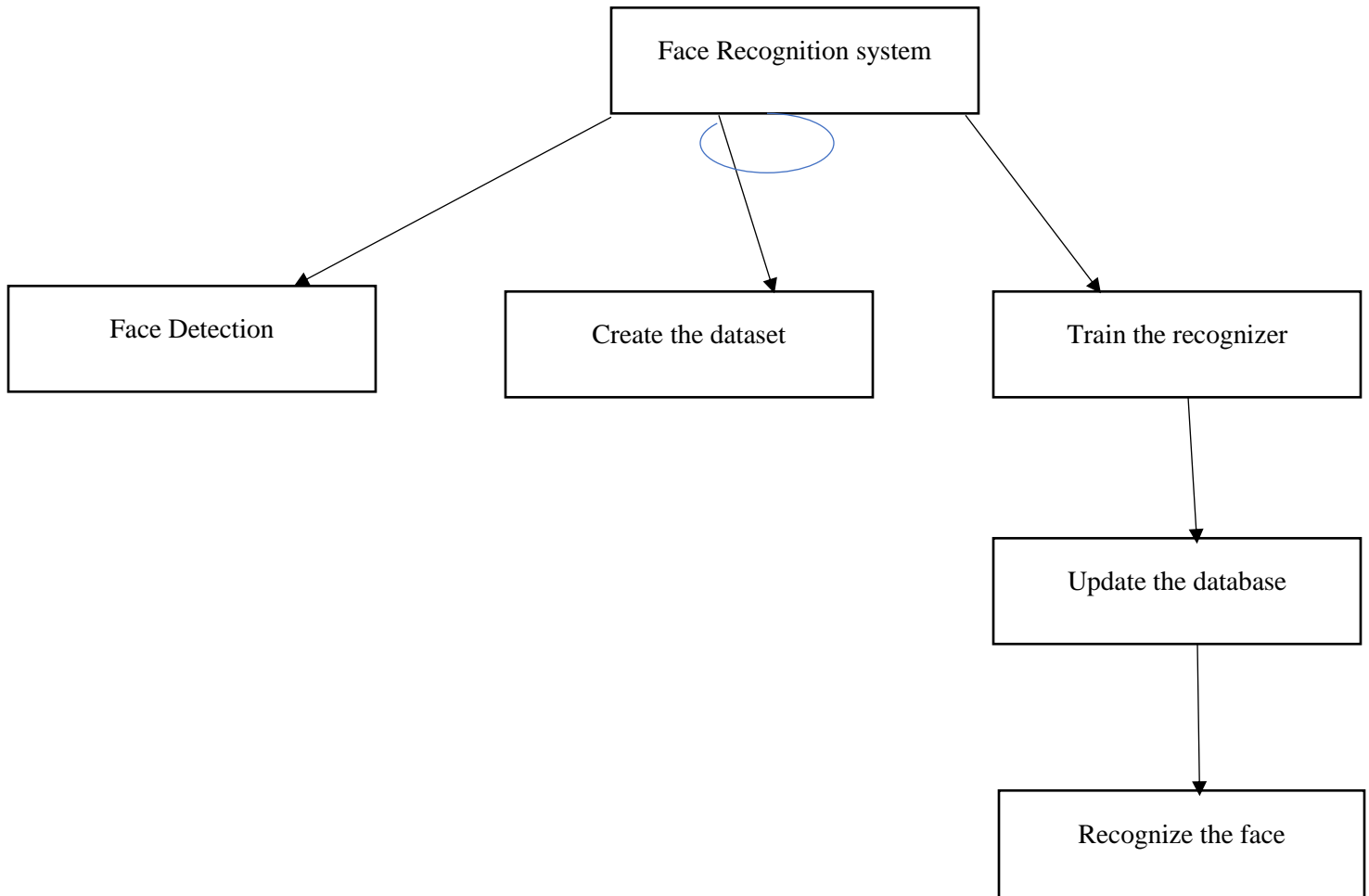


Figure 3.4: Structure Chart

3.7 Diagrams

3.7.1 Class diagram

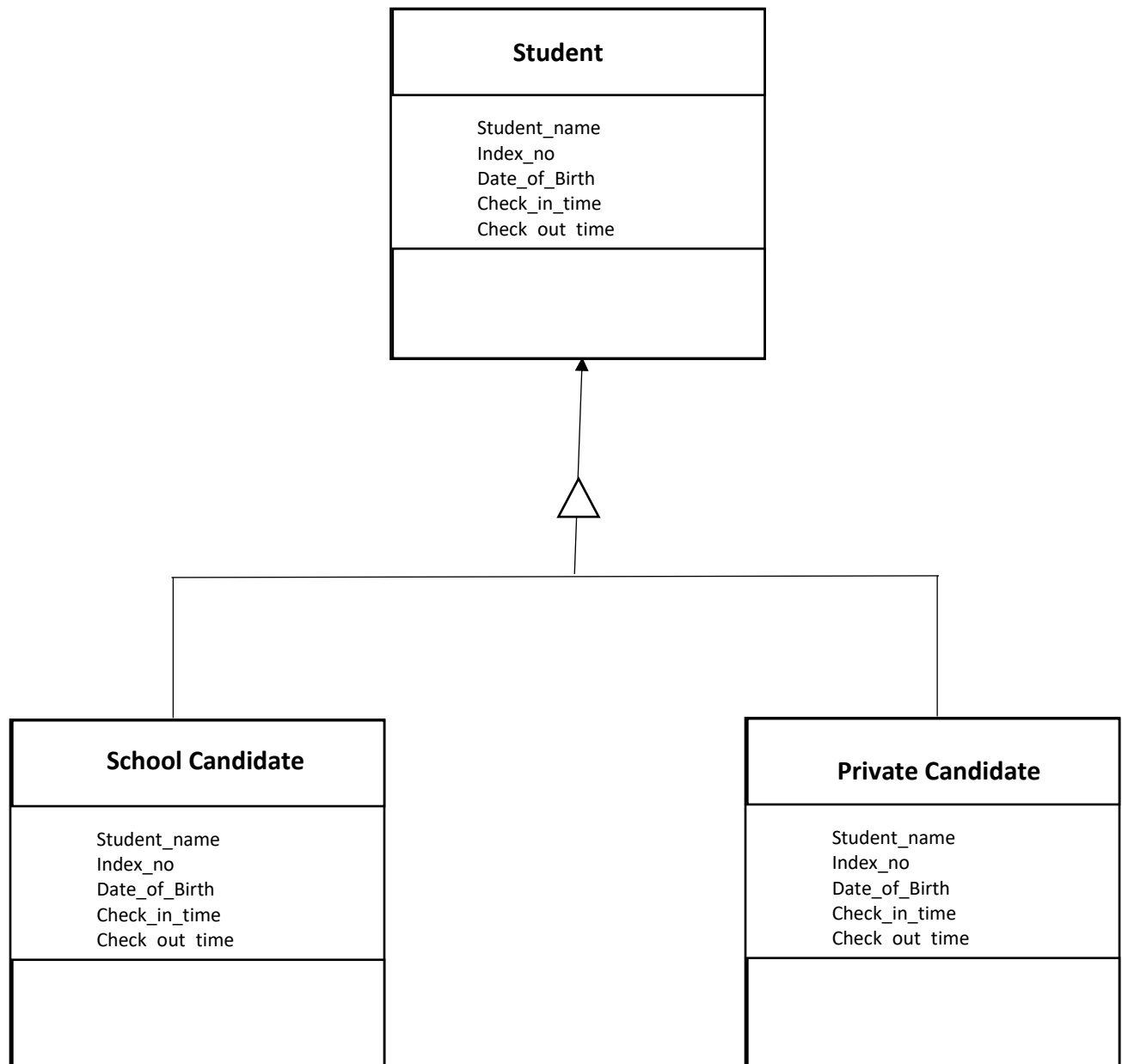


Figure 3.5: Class Diagram

State Diagram

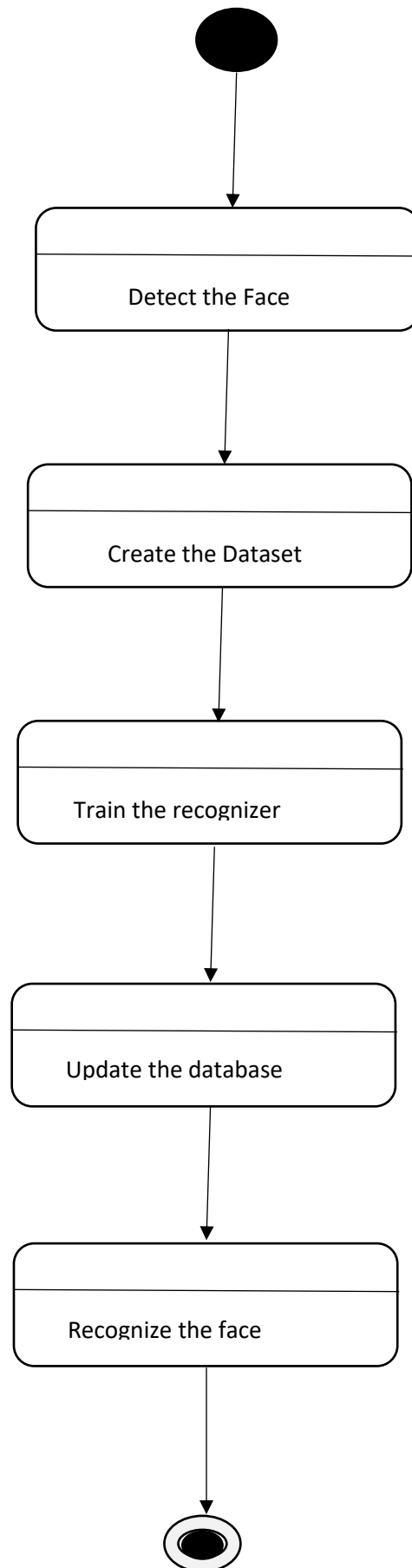


Figure 3.6: State Diagram

3.7.3 Sequence Diagram

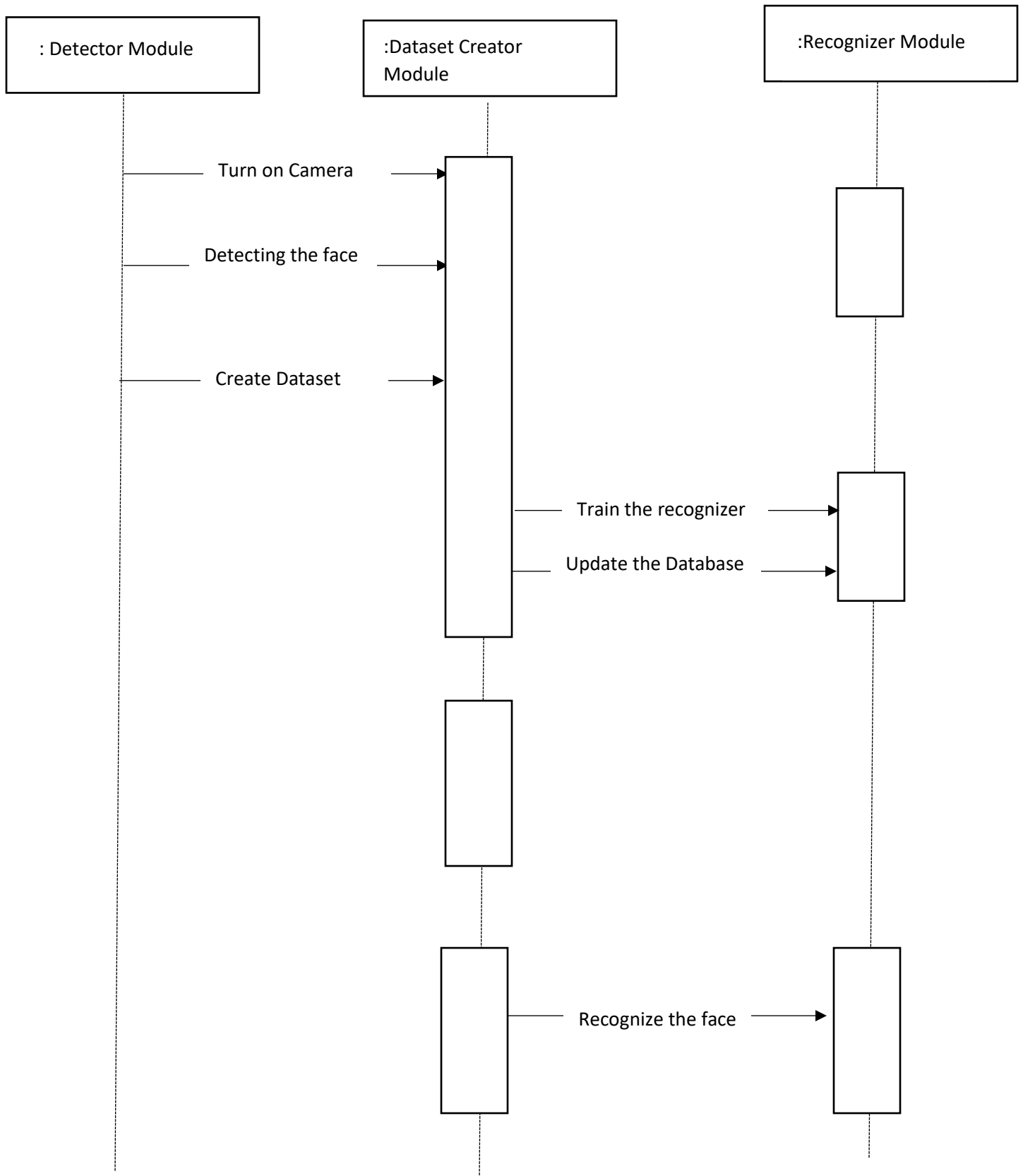


Figure 3.7: Sequence Diagram

3.8 Test Plan

Initially individual modules can be tested such as face recognition code can be tested with suitable test data in the following instances,

- ✓ Detecting the face of the student
- ✓ Creating the dataset of the student
- ✓ Training the recognizer
- ✓ Recognizing the face

3.9 Tools

OpenCV tools

CV module

NumPy module

PIP module

PIL module

3.10 Hardware/Software

Hardware

Keyboard

Mouse

Camera

RAM 4 GB

Processor 2.2 GHz

Hard disk

Backup devices

Software

Windows 10 -64-bit OS, x64 based processor

OpenCV version 3.4.4

Python version 2.7.15

SQLite Studio 3.2.1

3.11 Code Structure

3.11.1 Interface creation

The login interface created with the user registration and login phases.

In the user registration process the authorize user can save the username and the password in an internal file. Once it is successful the authorized user can log into the system.

Username and password are being compared with the given username and the password and generate necessary error/acknowledged messages.

Only the authorize user allows to operate the system such as scan the face, train the face and finally recognition of the candidate's face.[4]

3.11.2-Dataset creation

In this code structure the image scanning process is being automated.

For this process there are three libraries should be imported.

1. CV2 module
2. Sqlite3
3. NumPy

Initially the camera should be opened. Opening camera is being automated by this dataset creation code. The relevant database('FaceBase') is being connected at the beginning.

When the program starts up it allows to enter the index number and the full name of the candidate. If the record is already in the database the relevant details have been updated or inserted record is a new record it has been inserted into the database. The connected database has been disconnected at the end of the code.

A function called *insertorUpdate* is defined to handle the database operations of the program.

Face.xml file taken from OpenCV folder is being stored in the folder called 'Classifiers'.

3.11.3-Face training code

LBP Algorithm

LBP is a visual/texture descriptor, our faces are also composed of micro visual patterns. So, LBP features are extracted to form a feature vector that classifies a face from a non-face. It is a simple solution for the face recognition problem, which can be recognizing both the front face and side face. LBP is a simple and efficient text description operator which labels the pixels of an image by thresholds the neighborhood of each pixel and which produce the result as a binary number. Then the LBP combined with histogram. We can represent the face images with a simple data vector.

Strengths and weakness of LBPH

Even though current face recognition systems have reached a certain level of perfection but still there are many real application conditions which limits their good performance. The major advantage of this algorithm is that it produces better recognition rates in controlled environments and it is not profound to illumination.

How LBPH Algorithm works [5]

For each block, LBP looks at 9 pixels (3×3 window) at a time, and with a particular interest in the pixel located in the center of the window. It compares the central pixel value with every neighbor's pixel value under the 3×3 window. For each neighbor pixel that is greater than or equal to the center pixel, it sets its value to 1, and for the others, it sets to 0. After that, it reads the updated pixel values in a clockwise order and forms a binary number. Next, it converts the binary number into a decimal number, and that decimal number is the new value of the center pixel. Apply this for every pixel in a block. Then, it converts each block values into a histogram, finally one histogram for each block in an image. Finally, it concatenates these block histograms to form a one feature vector for one image.

3.11.4-Face Training [6]

To perform the Face training process following libraries should be imported.

1. CV2
2. NumPy
3. OS
4. PIL/Image

The captured images will be saved in an array list with the specific index number of the candidate. The captured color image will be converted into grayscale and stored in the specified folder.

Once the training face has been performed, a yml file called trainingData.yml is produced.

3.11.5-Face Recognition

Face recognition process is being performed with use of following imported libraries.

1. CV2
2. OS
3. NumPy
4. PIL/Image
5. Sqlite3
6. Pickle

Saved grayscale image is being compared with the captured image by camera, If the comparison is successful then the captured image will be produced along with text. For example, candidate full name and index number etc., The comparison is done with the aid of the database's stored details.

The relevant font color can be generated with the use of RGB function.

Chapter 4

User Evaluation and Testing

4.1 Introduction

Evaluation focuses on how well users can learn and use a product to achieve their goals. It also refers to how satisfied users are with that process. Face recognition system for examinations is not very difficult to handle with the use of simple techniques to conduct the face detection, face training and recognition processes. User evaluation form is annexed in the APPENDIX E.

Software Testing [7]

Software testing is an investigation conducting to provide shareholders with information about the quality of the software product or service under test. It can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.

4.2 Testing stages

The testing process should therefore proceed in stages where testing is carried out incrementally in combination with system implementation. The most widely used process consists of **four stages**:

- 1. Unit Testing:** Individual components are tested to ensure that they operate correctly. Each component is tested independently without other system components. e.g., Login module testing, Dataset creation module testing, Training module testing.
- 2. Integration Testing:** This phase involves testing collections of modules which have been integrated into sub-systems. Sub-systems may be independently designed. The most common problems which arise in large software systems are sub-system interface mismatches. The sub-system test process should therefore focus on the detection of interface errors by thoroughly working out the interface. e.g., Integrate Login module and Dataset creation module.
- 3. System Testing:** Sub systems are integrated to make up the entire system. The testing process is concerned with finding errors that result from unanticipated interactions between sub-systems and system components. It is also concerned with validating that the system meets its functional and non-functional requirements. The final face recognition system is being tested at the end of this testing strategy.
- 4. Acceptance Testing:** This is the final stage in the testing process before the system is accepted for operational use. The system is tested with data supplied by the system client. Acceptance testing may disclose errors and omissions in the system requirements definition because the real data exercises the system in different ways from the test data. It may also reveal requirements problems where the system's facilities do not really meet the user's needs or the system's performance is not acceptable.

4.3 Test Plan and Test Cases

Testing method is basically combined with *Verification* and *Validation*.

Test plan is a document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process.

A Test Case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application.

Test cases for the system

Below tables illustrates the tests cases generated for the face recognition system.

i. Login Module

Test caseID		1	
Tested Component		Login Module	
Module Name		Login	
Test Case		Login Screen	
Expected Outcome		Login success and open module	
Test Case Description			
No	Test Case	Actual Output	Status
1	Login Success	Enter correct user name and correct password.	Pass
2	Login Not Success	Enter either incorrect password or incorrect username.	Failure

Table 4.1: Login Module

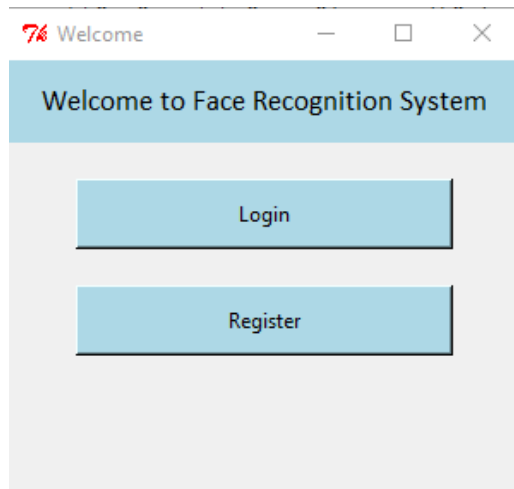


Figure 4.1: Welcome Screen

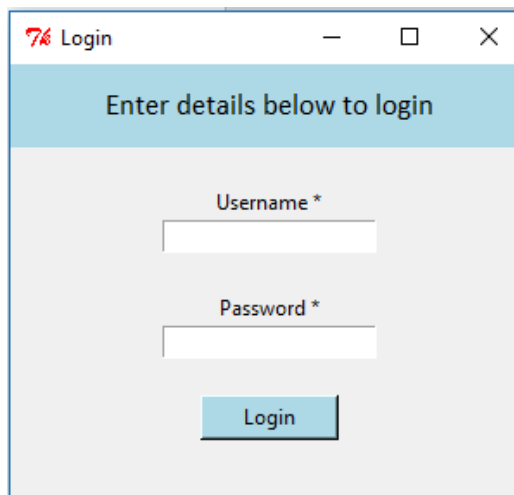


Figure 4.2: Main Login Screen

ii. Face Dataset Creation Module

Test caseID			2		
Tested Component			Face Dataset creation Module		
Module Name			Face Dataset creation		
Test Case			Face Dataset creation Screen		
Expected Outcome			Creating a face dataset		
Test Case Description					
No	Test case		Actual output	status	
1	User enters ID		Display Detected face	Ok	
2	User enters Name		Display Detected face	Ok	

Table 4.2: Face Dataset Creation Module

```

Enter your Index Number51
Enter your Name"G.A.Pavanya Ganguli"
.....

```



Figure 4.3: Dataset creation

ii. Face Training Module

Test caseID			3
Tested Component			Training Module
Module Name			Face Training
Test Case			Face Training Screen
Expected Outcome			Trained face output images
Test Case Description			
No	Test case	Actual output	status
1	User clicks on the 'Check the Face Scan' button.	If it is successful user gets the trained data.	Ok

Face Training Module: Table 4.3

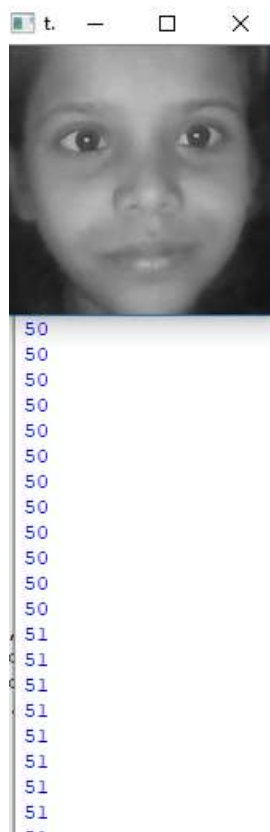


Figure 4.4: Face Training

iii. Face Recognizing Module

Test caseID		4	
Tested Component		Face Recognition Module	
Module Name		Face Recognition	
Test Case		Face Recognition Screen	
Expected Outcome		Recognized face with details	
Test Case Description			
No	Test case	Actual output	status
1	User clicks the ‘Scan at Exam’ button	If the scanning is successful display ‘Your Face Recognized’, if not ‘Your Face Not Recognized’	ok

Table 4.4: Face Recognizing Module

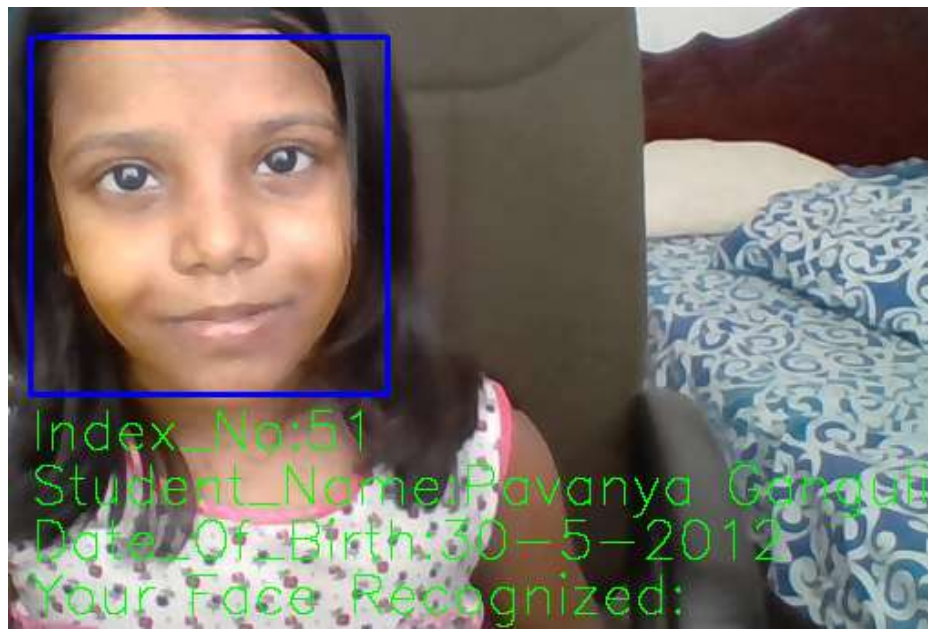


Figure 4.5: Correct Face Recognition



Figure 4.6: Face- Not Recognized

4.4 Acceptance Testing

The user acceptance testing should be conducted in examination working environment of the school with real transaction data sets. Authorized users were selected from the section, and asked to work with student face recognition system according to their privileges. Activities monitored while users working with the system.

To consider as a successful project the final outcome needs to be accepted by the end users. Therefore, system will handover to users for test run with questionnaire to capture the relevant authorized user feedbacks.

4.5 Problems Encountered

If the users are not detected previously (before the exam) the system should generate a message as “Your Face Not Recognized”. When a strange face is being detected it randomly selected another dataset and displayed from the database. This means overlapping faces in the database.

User training period was little bit difficult as this is a face recognition system. Users are not much familiar with this type of working environment.

During face detection procedure, students are not in still mood when they try to change their face mood. Therefore, the task was little bit difficult in capturing the face.

4.6 Lessons Learnt

Python language special features such as face recognizing techniques learnt during this project as follows. Face overlapping is being reduced by adding a special code other than the face recognizing code. Relevant libraries to be imported in creating dataset, face training and face detection process are recognized. Face training should be implemented by generating a collection of sample datasets.

The above-mentioned technical skills have been learnt in implementing the face recognizing system. Furthermore, this project gave me a chance to collaboratively work in an implementation environment specially with schoolchildren in my school. Proper time management and planning of the project was very crucial events in this post graduate degree program.

Chapter 5

Conclusion

5.1 Introduction

St. Peter's College Colombo 4 was established in 1927. IT infrastructure has been developed in 2017 and currently using smart boards in teaching subjects. Rapid growing IT infrastructure is most significant benchmark in education.

Face recognition system for examinations is most benefited for this type of academic institute.

5.2 Summary of the result

The final system was being tested by getting sample student data. These data captured as front face image of the student with the relevant student details such as student name, student index number, date of birth of the student and exam time if required.

At the time of detecting the face the program allows authorized user to insert each student index number and name of the student. The database is being updated against each student's details.

Next step is to train all detected faces of the students. Afterwards at the time of recognition time the student can scan his face where he can see his index number, name and date of birth with the recognition. Another special note called "You Face Recognized" is displayed with the recognized image.

More than fifty students had been participated to scan through their faces and check the successfulness of the project.

True Positive, True Negative, False Positive and False Negative theory

True Positive <i>Interpretation: You predicted positive and it's true.</i>	False Positive <i>Interpretation: You predicted positive and it's false.</i>
False Negative <i>Interpretation: You predicted negative and it's false.</i>	True Negative <i>Interpretation: You predicted negative and it's true.</i>

Table 5.1: TP, TN, FP, FN Theory

Following formulae are illustrated for the calculation of True Positive Rate, True Negative Rate

False Negative Rate and False Positive Rate. The reasons for obtaining False Negative 15% and False Positive 20% are difficulty in recognizing the face in some instances specially when student movements effect the capturing the face and environment brightness changes.

$$\text{True Positive Rate} = \frac{\text{TP}}{\text{TP} + \text{FN}} = 80\%$$

$$\text{True Negative Rate} = \frac{\text{TN}}{\text{TN} + \text{FP}} = 85\%$$

$$\text{False Negative Rate} = \frac{\text{FN}}{\text{FN} + \text{TP}} = 15\%$$

$$\text{False Positive Rate} = \frac{\text{FP}}{\text{FP} + \text{TN}} = 20\%$$

5.3 Future Improvements

Face recognition of a candidate is one of a crucial stage in exam management system. This project is suitable at the time of student registration to issue the admission letters for qualified students.

Implement the exam management system and identify the modules of the system to integrate the face recognition system to serve the education industry better.

One module can be identified as issuing the admission letters to the candidate. This module can have sub processes identify as count the qualified students, storing qualified student details in the database later issuing admission letters to the relevant students.

All three modules in this face recognition system name as face detection, face training and face recognition can be integrated into this examination management system.

References

[1] ANON

Your Bibliography: [3]2019. [Online]. Available: <http://www.DERMALOGs.com>. [Accessed: 28- May- 2019].

[2] FACE RECOGNITION SOFTWARE: BEST-IN-CLASS ENTERPRISE FACIAL RECOGNITION SECURITY PLATFORM

Your Bibliography: [1]"Face Recognition Software: Best-in-Class Enterprise Facial Recognition Security Platform", *FaceFirst Face Recognition Software*, 2019. [Online]. Available: <http://www.facefirst.com>. [Accessed: 28- May- 2019].

[3] PATNAIK, L., GUPTA, V., MALLICK, S., BAPAT, K. AND SHRIMALI, V.

Learn OpenCV (C++ / Python)

In-text: (Patnaik et al., 2019)

Your Bibliography: Patnaik, L., Gupta, V., Mallick, S., Bapat, K. and Shrimali, V. (2019). *Learn OpenCV (C++ / Python)*. [online] Learnopencv.com. Available at: <https://www.learnopencv.com>

[4] OPENCV FACE RECOGNITION | HOW TO WRITE A FACE RECOGNITION PROGRAM + SQLITE DATABASE IN PYTHON PART-4

In-text: (YouTube, 2019)

Your Bibliography: YouTube. (2019). *OpenCV Face Recognition | How to write a face recognition program + Sqlite Database in python Part-4*. [online] Available at: <https://www.youtube.com/watch?v=6gWS2CdtZrs>

[5] NI-CHI-TECH/FACE-RECOGNITION

In-text: (GitHub, 2019)

Your Bibliography: GitHub. (2019). *ni-chi-Tech/Face-Recognition*. [online] Available at: <https://github.com/ni-chi-Tech/Face-Recognition>

[6] FACE RECOGNITION AND TRAINING IN OPENCV

In-text: (YouTube, 2019)

Your Bibliography: YouTube. (2019). *Face recognition and training in opencv*. [online] Available at: <https://youtu.be/WfrHM24aFIF>

[7] SOFTWARE TESTING - DEFINITION, TYPES, METHODS, APPROACHES

In-text: (Software Testing Material, 2019)

Your Bibliography: Software Testing Material. (2019). *Software Testing - Definition, Types, Methods, Approaches*. [online] Available at: <https://www.softwaretestingmaterial.com/software-testing/>

APPENDIX A-DESIGN DOCUMENTATION

Use Case diagrams and descriptions

Use case description for Dataset Creation Module

Use case	Dataset creation
Actors	Authorized user, Candidate
Overview	
Authorized user scans the candidate's face with necessary details. Candidate should give his/her full name with relevant details.	
Pre conditions	
The camera should be turned on. The candidate's details may or may not be in the database.	
Flow of events	
1. Authorized user let the candidate to give personal details.	
2. Candidate gives the full name and other relevant details.	
3. Authorized user scans the face.	
Post conditions	
After the scanning process the necessary modifications must be updated in the database. The camera should be turned off.	

Table A.1: Use case description for Dataset Creation Module

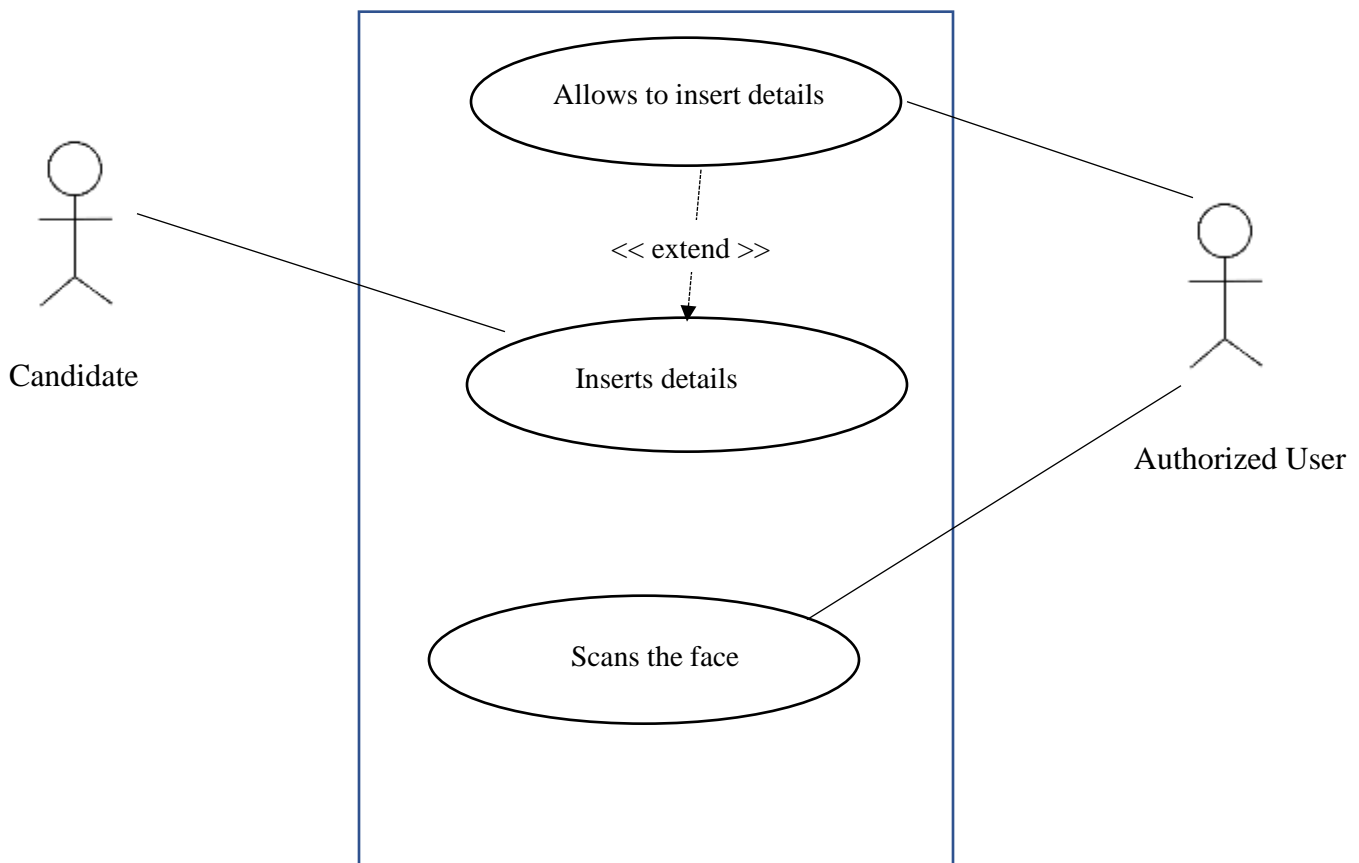


Figure A.1: Use Case Diagram Dataset Creation

Use case description for Face Training Module

Use case	Face Training
Actors	Authorized user
Overview	
Authorized user trains the faces of the candidates.	
Pre conditions	
Necessary candidate faces should be scanned before face training process.	
Flow of events	
1. Checks the scanned faces are in the specified folder.	
2. Trains all the faces stored in the folder.	
Post conditions	
The required yml file will be generated in the specified folder	

Table A.2: Use case description for Face Training Module

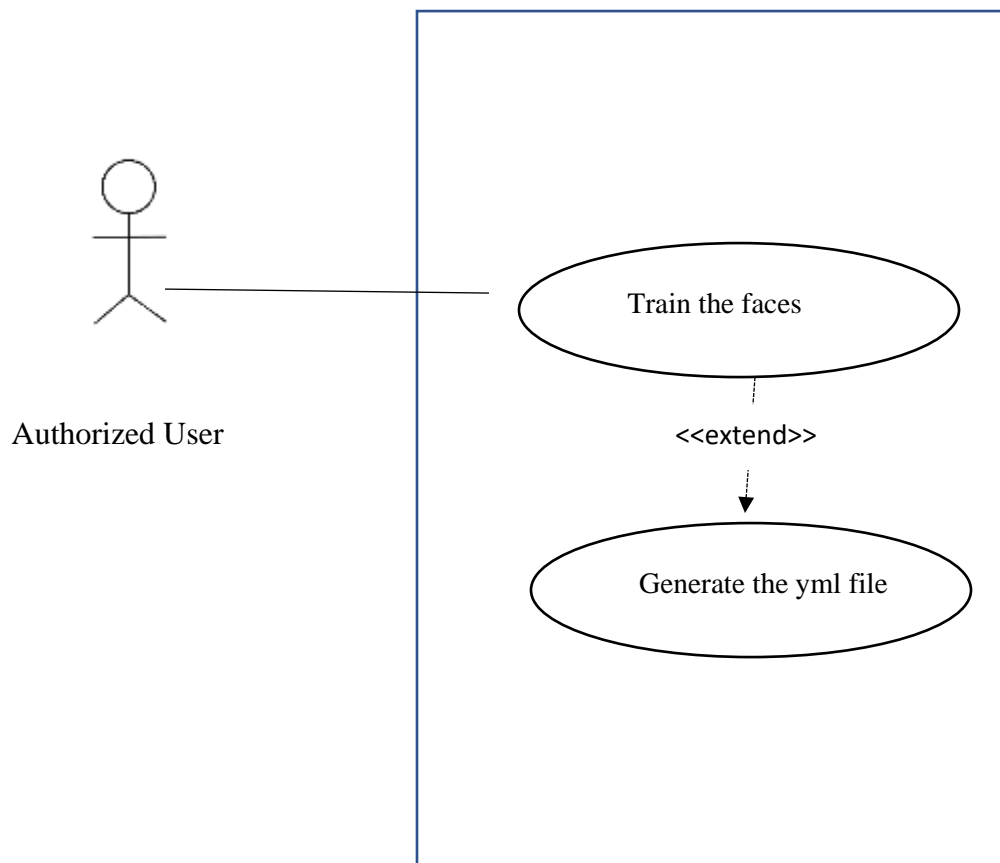


Figure A.2: Use Case Diagram Face Training

Use case description for Face Recognition Module

Use case	Face Recognition
Actors	Authorized user, Candidate
Overview	
Authorized user recognizes the relevant faces of the candidate.	
Pre conditions	
The camera should be turned on. The faces should be trained before the recognition.	
Flow of events	
1. Allow the candidate to scan his/her face at the exam center	
2. If the face is recognized the candidate's details will be displayed if not A note will be displayed 'Your Face Not Recognized'.	
Post conditions	
The camera should be turned off.	

Table A.3: Use case description for Face Recognition Module

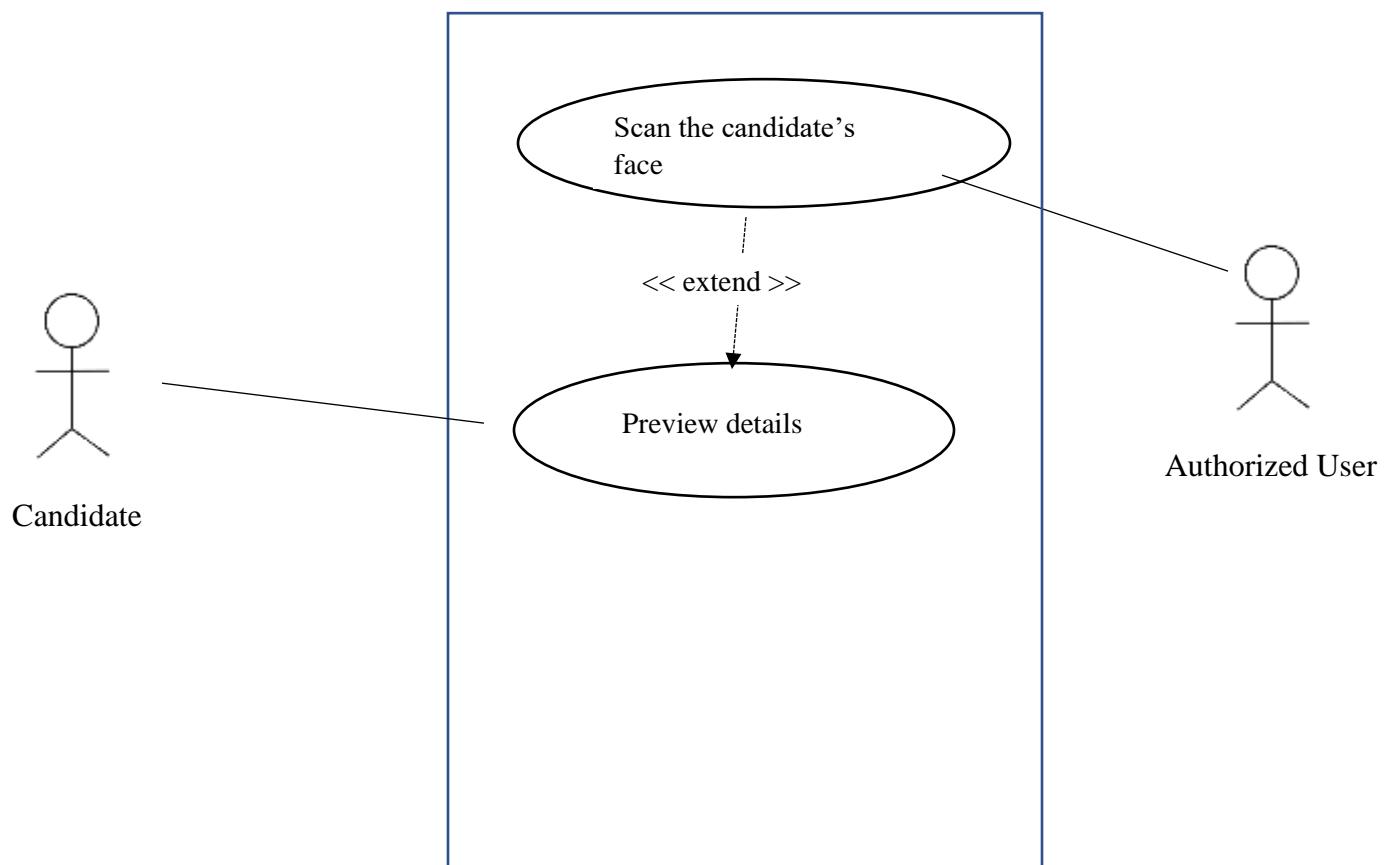


Figure A.3: Use Case Diagram Face Recognition

Activity diagram for face recognition system

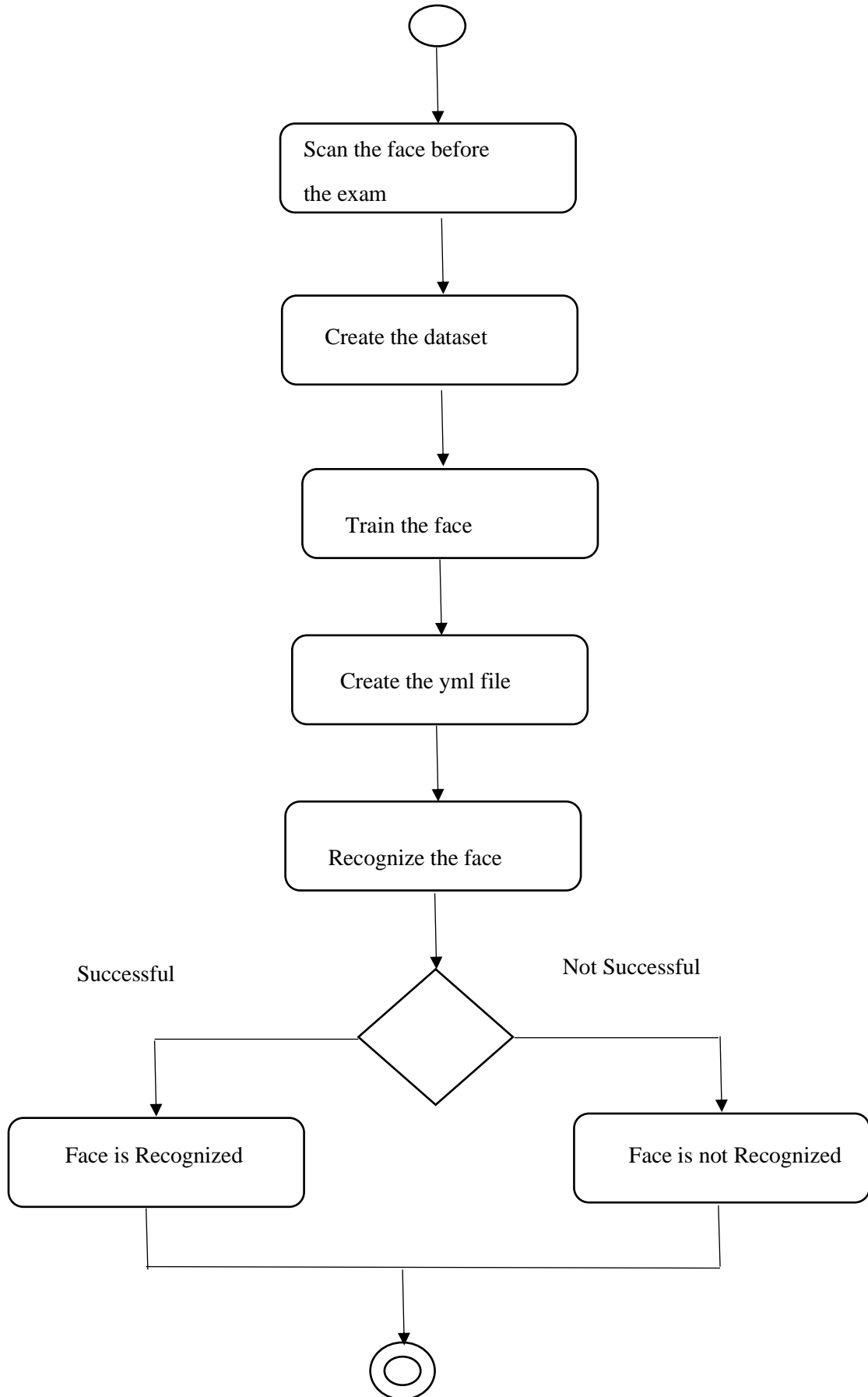
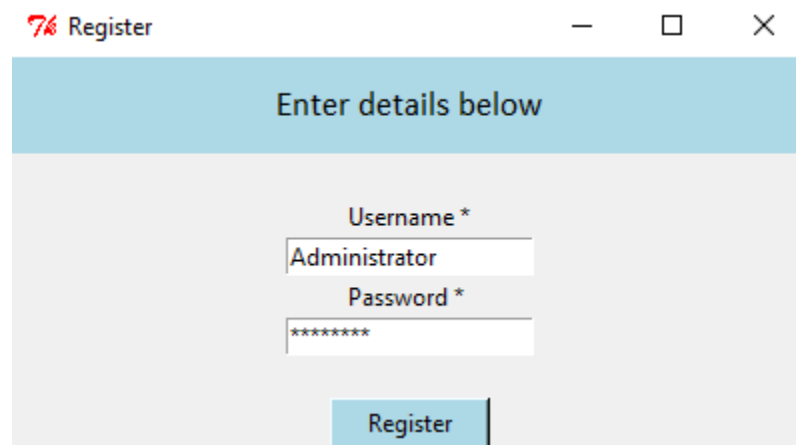


Figure A.4: Activity Diagram

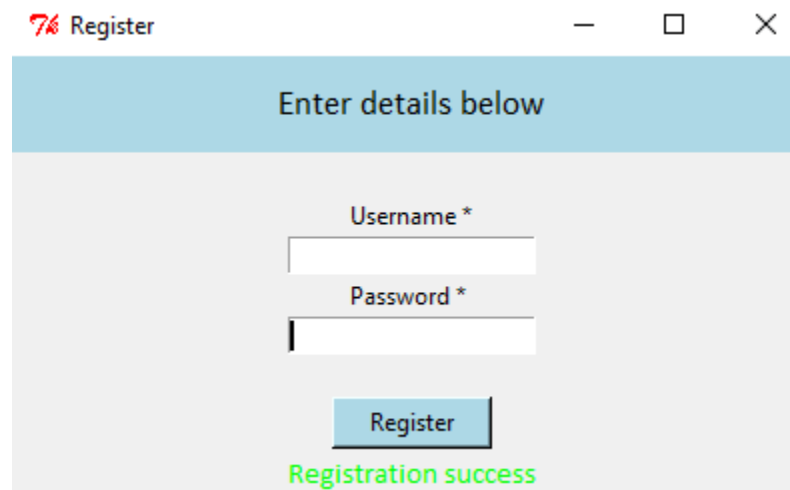
APPENDIX B-USER DOCUMENTATION

Interfaces-Login and system



A screenshot of a web application window titled "76 Register". The window has a light blue header bar with the text "Enter details below". Below the header, there are two input fields: "Username *" and "Password *". The "Username *" field contains the text "Administrator". The "Password *" field contains a series of asterisks "*****". Below the input fields is a blue button labeled "Register".

Figure B.1: User Registration Screen



A screenshot of the same web application window titled "76 Register". The window has a light blue header bar with the text "Enter details below". Below the header, there are two input fields: "Username *" and "Password *". The "Username *" field is empty. The "Password *" field is empty. Below the input fields is a blue button labeled "Register". At the bottom of the window, the text "Registration success" is displayed in green.

Figure B.2: Successful User Registration

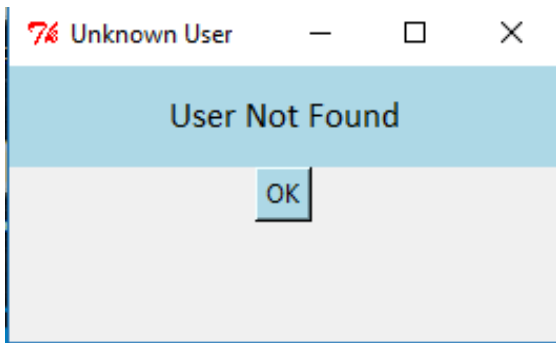


Figure B.3: User not found Error

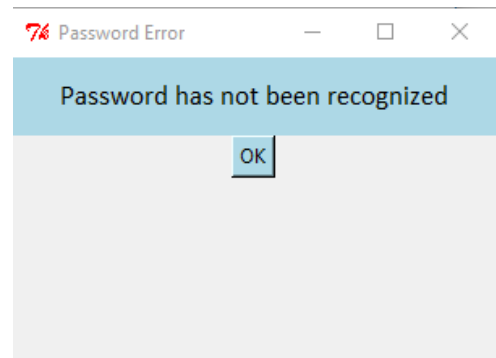


Figure B.4: Password Error

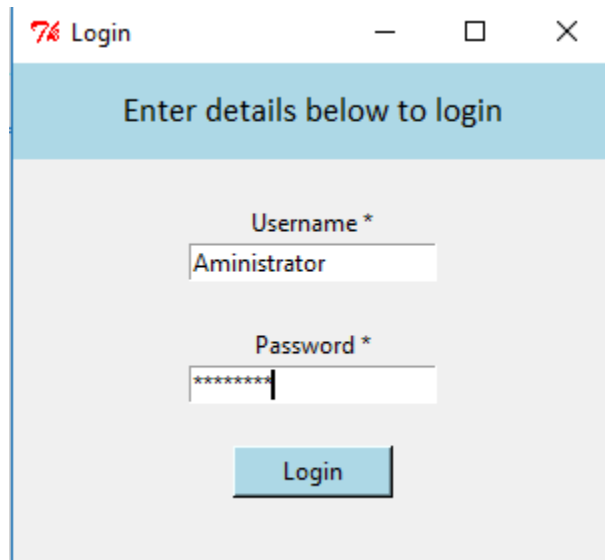


Figure B.5: Successful User Login

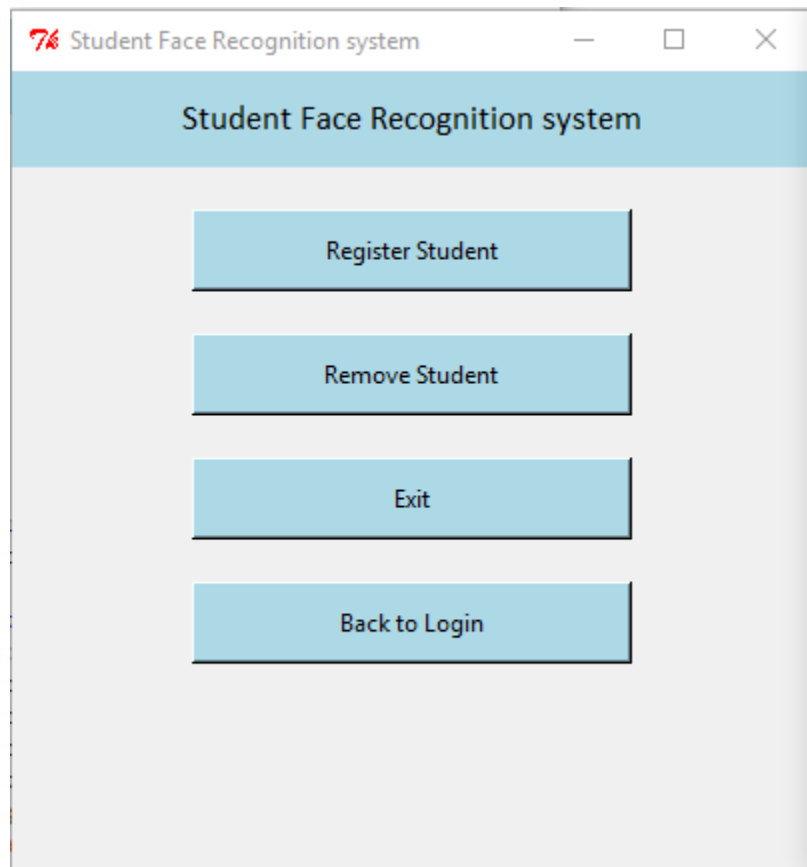


Figure B.6: System Main Interface

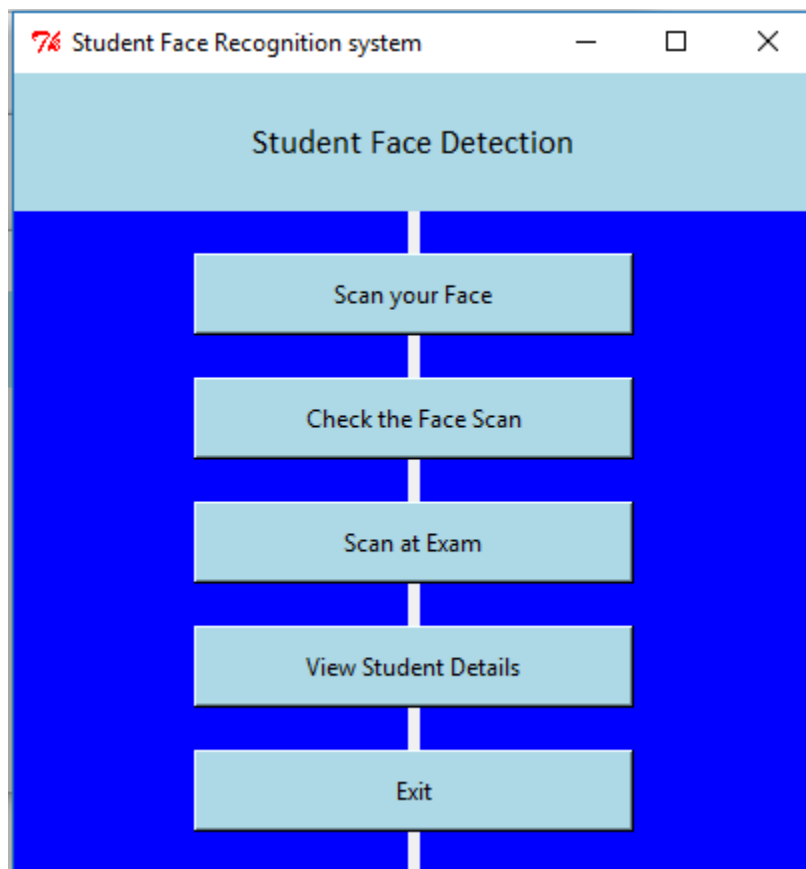


Figure B.7: Student Face Recognition Main Screen



Figure B.8: Scanned Student Details

APPENDIX C-CODE LISTINGS

User login code

```
from Tkinter import *
import os
import sys
import cv2
from PIL import ImageTk,Image
import sqlite3

def delete2():
    screen3.destroy()

def delete3():
    screen4.destroy()

def delete4():
    screen5.destroy()

def detect_face():
    global screen6
    screen6 = Toplevel(screen)
    screen6.title("Student Face Recognition system")
    screen6.configure(background='blue')
    screen6.geometry("400x400")
    Label(screen6, text = "Student Face Detection", width="400", height="3",bg="Light
blue",font=("calibri",13)).pack()
    Label(screen6,text="").pack()
    Button(screen6, text = "Scan your Face", height="2", width="30",bg="Light
blue",command=dataset).pack()
    Label(screen6,text="").pack()
```

```

Button(screen6, text = "Check the Face Scan", height="2", width="30",bg="Light
blue",command=trainner).pack()

Label(screen6,text="").pack()

Button(screen6, text = "Scan at Exam", height="2", width="30",command=run_detection,bg="Light
blue").pack()

Label(screen6,text="").pack()

Button(screen6, text = "View Student Details", height="2", width="30",command=view,bg="Light
blue").pack()

Label(screen6,text="").pack()

Button(screen6, text = "Exit", height="2", width="30",command = exit,bg="Light blue").pack()

Label(screen6,text="").pack()

Button(screen6, text = "Open Database", height="2", width="30",command = open_db,bg="Light
blue").pack()

Label(screen6,text="").pack()

def open_db():

    os.system("explorer D:\\SqliteStudio\\SqliteStudio.exe")

def run_detection():

    os.system('facedetect.py')

def dataset():

    os.system('DatasetCreator.py')

def trainner():

    os.system('trainner.py')

```

```

def view():
    os.system('list.py')

def remove():
    os.system("explorer D:\\\\SQLite")

def login_success():
    global screen3
    screen3 = Toplevel(screen)
    screen3.title("Student Face Recognition system")
    screen3.geometry("400x400")

    Label(screen3, text = "Student Face Recognition system",bg="Light blue", width="300", height="2",
font=("calibri",13)).pack()

    Label(screen3,text="").pack()

    Button(screen3, text = "Register Student", height="2", width="30",bg="Light
blue",command=detect_face).pack()

    Label(screen3,text="").pack()

    Button(screen3, text = "Remove Student", height="2", width="30",bg="Light
blue",command=remove).pack()

    Label(screen3,text="").pack()

    Button(screen3, text = "Exit", height="2", width="30",bg="Light blue", command=exit).pack()

    Label(screen3,text="").pack()

    Button(screen3, text = "Back to Login", height="2", width="30",bg="Light
blue",command=login).pack()

    Label(screen3,text="").pack()

def password_not_recognized():
    global screen4
    screen4 = Toplevel(screen)
    screen4.title("Password Error")
    screen4.geometry("300x300")

```

```

Label(screen4, text = "Password has not been recognized",bg="Light blue", width="300", height="2",
font=("calibri",13)).pack()

Button(screen4, text = "OK", command=delete3,bg="Light blue").pack()

def user_not_found():
    global screen5
    screen5 = Toplevel(screen)
    screen5.title("Unknown User")
    screen5.geometry("150x100")
    Label(screen5, text = "User Not Found",bg="Light blue", width="300", height="2",
font=("calibri",13)).pack()
    Button(screen5, text = "OK", command=delete4,bg="Light blue").pack()

def register_user():
    print("Working")

    username_info = username.get()
    password_info = password.get()

    file=open(username_info, "w")
    file.write(username_info+"\n")
    file.write(password_info)
    file.close()

    username_entry.delete(0, END)
    password_entry.delete(0, END)

Label(screen1, text = "Registration success", fg = "green", font=("calibri",11)).pack()

```

```

def login_verify():
    username1 = username_verify.get()

    password1 = password_verify.get()
    username_entry1.delete(0, END)
    password_entry1.delete(0, END)

    list_of_files = os.listdir(os.getcwd())
    if username1 in list_of_files:
        file1 = open(username1, "r")
        verify = file1.read().splitlines()
        if password1 in verify:
            login_success()
        else:
            password_not_recognized()
    else:
        user_not_found()

```

```

def register():
    global screen1
    screen1=Toplevel(screen)
    screen1.title("Register")
    screen1.maxsize(400,500)

    global username
    global password
    global username_entry
    global password_entry
    username = StringVar()
    password = StringVar()

```

```

Label(screen1, text = "Enter details below",bg="Light blue", width="300", height="2",
font=("calibri",13)).pack()

Label(screen1, text = "").pack()

Label(screen1, text = "Username *").pack()


global username_entry
global password_entry
username_entry=Entry(screen1, textvariable = username)


username_entry.pack()
Label(screen1, text = "Password * ").pack()
password_entry=Entry(screen1, textvariable = password,show="*")
password_entry.pack()
Label(screen1, text = "").pack()

Button(screen1, text="Register", width =10, height =1, command = register_user,bg="Light
blue").pack()

def login():
    global screen2
    screen2 = Toplevel(screen)
    screen2.title("Login")
    screen2.geometry("300x250")

    Label(screen2, text = "Enter details below to login",bg="Light blue", width="300", height="2",
font=("calibri",13)).pack()
    Label(screen2, text = "").pack()


    global username_verify
    global password_verify


    username_verify = StringVar()
    password_verify = StringVar()

```

```

global username_entry1
global password_entry1

Label(screen2, text = "Username *").pack()
username_entry1 = Entry(screen2, textvariable = username_verify)
username_entry1.pack()
Label(screen2, text="").pack()
Label(screen2, text = "Password *").pack()

password_entry1 = Entry(screen2, textvariable = password_verify, show="*")

password_entry1.pack()
Label(screen2, text = "").pack()
Button(screen2, text = "Login", width = 10, height = 1, command = login_verify, bg="Light blue").pack()

def main_screen():
    global screen
    screen = Tk()
    screen.geometry("300x250")
    screen.title("Welcome")

    Label(text="Welcome to Face Recognition System", bg="Light blue", width="300", height="2", font=("calibri",13)).pack()

    Label(text="").pack()
    Button(text="Login", height="2", width="30", command = login, bg="Light blue").pack()
    Label(text="").pack()
    Button(text="Register", height="2", width="30", command = register, bg="Light blue").pack()
screen.mainloop()
main_screen()

```

Dataset creation code

```
import cv2
import sqlite3
import numpy as np

faceDetect=cv2.CascadeClassifier('Classifiers/face.xml');
cam=cv2.VideoCapture(0);

def insertorUpdate(id,Name):
    conn=sqlite3.connect("FaceBase.db")
    cmd="SELECT * FROM Student WHERE Index_No="+str(id)
    cursor=conn.execute(cmd)
    isRecordExist=0
    for row in cursor:
        isRecordExist=1
    if(isRecordExist==1):

cmd="UPDATE Student SET Student_Name="+str(Name)+" WHERE Index_No="+str(id)
    else:
        cmd="INSERT INTO Student(Index_No,Student_Name) Values("+str(id)+"," +str(Name)+")"

    conn.execute(cmd)
    conn.commit()
    conn.close()

id=raw_input('Enter your Index Number')
Name=raw_input('Enter your Name')
insertorUpdate(id,Name)
sampleNum=0;
while(True):
```

```
ret,img =cam.read();

gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
faces=faceDetect.detectMultiScale(gray,1.3,5);

for(x,y,w,h) in faces:
    sampleNum=sampleNum+1;
    cv2.imwrite("Data/User."+str(id)+ "." +str(sampleNum)+ ".jpg",gray[y:y+h,x:x+w])
    cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)
    cv2.waitKey(100);
cv2.imshow("images",img);
cv2.waitKey(1);
if sampleNum>20:
    break;
cam.release()
cv2.destroyAllWindows()
```

Face Training code

```
import os
import cv2
import numpy as np
from PIL import Image

recognizer=cv2.createLBPHFaceRecognizer();
path='Data'

def getImagesWithID(path):
    imagePaths=[os.path.join(path,f) for f in os.listdir(path)]
    faces=[]
    IDs=[]

    for imagePath in imagePaths:
        faceImg=Image.open(imagePath).convert('L');
        faceNp=np.array(faceImg,'uint8')
        ID=int(os.path.split(imagePath)[-1].split('.')[1])
        faces.append(faceNp)
        print ID
        IDs.append(ID)
        cv2.imshow("training",faceNp)
        cv2.waitKey(10)
    return IDs, faces

Ids,faces=getImagesWithID(path)
recognizer.train(faces,np.array(Ids))
recognizer.save('Recognizer/trainningData.yml')
cv2.destroyAllWindows()
```

Face Recognition code

```
import cv2,os
import numpy as np
from PIL import Image
import pickle
import sqlite3

recognizer =cv2.createLBPHFaceRecognizer()
recognizer.load("Recognizer\\trainingData.yml")
CascadePath ="Classifiers/face.xml"
faceCascade =cv2.CascadeClassifier(CascadePath);
path ='Data'

def getProfile(id):
    conn=sqlite3.connect("FaceBase.db")
    cmd="SELECT * FROM Student WHERE Index_No="+str(id)
    cursor=conn.execute(cmd)
    profile=None
    for row in cursor:
        profile=row
    conn.close()
    return profile
cam = cv2.VideoCapture(0)
font = cv2.cv.InitFont(cv2.cv.CV_FONT_HERSHEY_SIMPLEX, 1, 1, 0, 1, 1)
profile={}
while True:
    ret, img =cam.read()
    if ret is True:
        gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    else:
        break
```

```
faces=faceCascade.detectMultiScale(gray,scaleFactor=1.2,minNeighbors=5,minSize=(100,100),flags=cv2.CASCADE_SCALE_IMAGE)
```

```
for(x,y,w,h) in faces:
```

```
    id, conf = recognizer.predict(gray[y:y+h,x:x+w])
```

```
    cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
```

```
    if(conf<60):
```

```
        profile=getProfile(id)
```

```
    else:
```

```
        id=0
```

```
        profile=getProfile(id)
```

```
        cv2.cv.PutText(cv2.cv.fromarray(img),"Your Face Not  
Recognized:",(x,y+h+120),font,(255,0,0))
```

```
    profile=getProfile(id)
```

```
    if(profile!=None):
```

```
        cv2.cv.PutText(cv2.cv.fromarray(img),"Index_No:"+str(profile[0]),(x,y+h+30),font, (0,255,0))
```

```
        cv2.cv.PutText(cv2.cv.fromarray(img),"Student_Name:"+str(profile[1]),(x,y+h+60),font,  
(0,255,0))
```

```
        cv2.cv.PutText(cv2.cv.fromarray(img),"Date_Of_Birth:"+str(profile[2]),(x,y+h+90),font,  
(0,255,0))
```

```
        cv2.cv.PutText(cv2.cv.fromarray(img),"Your Face Recognized:",(x,y+h+120),font,(0,255,0))
```

```
cv2.imshow("Face",img)
```

```
cv2.waitKey(10)
```

APPENDIX D-CLIENT CERTIFICATION LETTER

From,
Head of IT,
St Peter's College,
Colombo 4.

To,
Ms. D.R.N Wickrama,
127/3, Anderson Road,
Dehiwela.

Dear Ms. Ruwani,

With great pleasure and delight, we would like to inform you that your software has been accepted by our school. After thorough inspection and trials, we were satisfied by the results your software provided us.

We would like to install the software within a month and hence the details regarding this are expected to be heard from your side at the earliest.

Yours faithfully,

.....
Krishan Sanjeewa,
Head of IT,
St. Peter's College,
Colombo 4.

APPENDIX E-USER EVALUATION FORM

Face Recognition System - St. Peter's College Colombo 4					
Date:		Name:		Designation:	
No	Question	Strongly Agree	Agree	Strongly disagree	Disagree
1	Interfaces are simple and Understandable.				
2	Easily navigate through the system.				
3	System displays information more familiar.				
4	Functions are more reliable.				
5	Easy to modify student details.				
Comments:					

Note.

Please evaluate new software by referring each statement given on the form by putting a tick (✓).

APPENDIX F-USER GUIDE

The user needs to follow the below mentioned instructions to operate the Face Recognition system.

- ✓ Register the user in the system.
- ✓ Login procedure.
- ✓ Create the image dataset of the candidate.
- ✓ Train the candidate face.
- ✓ Scan at the exam venue.

How to create users in face recognition system

- 1) Click on 'Register' on the Welcome screen.
- 2) Suggest a user name and a password then register the user.
- 3) Next time use this username and password to login to the system.
- 4) If step three is successful, a label will display 'Registration Success'.

How to login to the system

- 1) Click on 'Login' on the Welcome screen.
- 2) Enter the username and password which you created during the user registration.
- 3) Click on 'Login' button.
- 4) If step three is successful automatically open the main interface.

Create the image dataset of the candidate

- 1) Click on 'Register Student' on the main interface.
- 2) Automatically open the Student Face Detection main screen.
- 3) Then click on 'Scan your face' button.
- 4) User allows to scan the candidate's face.

Train the candidate face

- 1) Click on 'Check the Face Scan' button on the Face Detection main screen.
- 2) User will be able to train the faces of all candidates.

Scan at the exam

- 1) Click on 'Scan at Exam' button on the Face Detection main screen.
- 2) Candidate's face recognition progresses in this stage.

Remove student details

- 1) Click on 'Remove Student' on the main interface.
- 2) It will automatically open the required directory.
- 3) Open the 'Data' folder which stores the scanned faces of the students.
- 4) Select the required student image and delete them all.

GLOSSARY

Client-server model- Client–server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.

Cloud Backup-It is a strategy for backing up data that involves sending a copy of the data over a proprietary or public network to an off-site server.

CV2 - Python Interface for OpenCV.

Database- A database is an organized collection of data, generally stored and accessed electronically from a computer system.

NumPy - Library for the Python programming language, adding support for large, multi-dimensional arrays and matrices.

Object-Oriented Technology- It is a software design model in which objects contain both data and the instructions that work on the data.

OpenCV- Library of programming functions mainly aimed at real-time computer vision.

Pickle - It is used for serializing and de-serializing a Python object structure.

Pillow - It is a friendly Python Imaging Library.

PIP - Package-management system used to install and manage software packages written in Python.

SQLite – It is a relational database management system contained in a C programming library.

Stand-alone model-A desktop or laptop computer that is used on its own without requiring a connection to a local area network (LAN) or wide area network (WAN).