**STUDENT ENTRANCE MASKED**

**FACE RECOGNITION AND**

**VERIFICATION SYSTEM**

A Research/ Capstone Project

Presented to the Faculty of the

College of Computer Studies and Information Technology

Southern Leyte State University

In Partial Fulfillment of the Requirements

for the degree Bachelor of Science in Information Technology

By

Myrelle R. Alcesto

Alphonse Victor N. Antegra

Neil Brian S. Antipolo

Jesheryll P. Calivoso

Ms. Jannie Fleur Oraño

Adviser

December, 2021

Republic of the Philippines

**SOUTHERN LEYTE STATE UNIVERSITY**

Sogod, Southern Leyte

Website: www.slsuonline.edu.ph

Email: slsumaincampus@gmail.com

op@slsuonline.edu.ph

Telefax No. (053) 382-3294

***College of Computer Studies and Information Technology***

**APPROVAL SHEET**

The Capstone Project Study entitled *Student Entranced Masked Face Recognition and Verification*prepared and submitted by Alcesto, Antegra, Calivoso, and Antipolo has been examined and is recommended for approval and acceptance.

RECOMMENDED:

**JANNIE FLEUR V. ORAÑO**  **RHODERICK MALANGSA**

Adviser/ITSO Manager Research Facilitator

APPROVED by the Committee on Oral Examination with a grade of PASSED on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ALEX C. BACALLA**

Chairman

**JORTON TAGUD** **RENE RADAZA**

Member Member

ACCEPTED and APPROVED in partial fulfillment of the requirements for a Bachelor of Science in Information Technology.

**ALEX C. BACALLA**

Dean, CCSIT

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**DEDICATION**

First of all, we dedicate this study to our Almighty God for giving us strength and guidance in doing this project.

To our parents, relatives and loved ones, for being our primary source of motivation to continue struggling to finish this project.

To our instructors, who impart their knowledge to us to prepare us for our future.

To our adviser, Ms. Jannie Fleur Oraño, for her continuous support and guidance to us. For teaching us and for being our motivator to finish this study.

To each member of team BLANK, for having each other’s backs in times of difficulty and uncertainty. This challenge is not easy, but we made it possible to achieve and complete together because we are empowered and motivated by everyone around us.

And last but not the least, to our Alma Mater, Southern Leyte State University for molding us to become great professionals for a better future.

**ACKNOWLEDGEMENT**

We would like to thank all the people who continuously support and help us to the success of this project.

To the Almighty God, for his guidance and for giving us strength to overcome our weaknesses.

To our ever patient and humble adviser, Ms. Jannie Fleur Oraño who never stops sharing her knowledge to us and for motivating us despite a lot of challenges we had encounter during the development of this study.

To our Capstone Instructor who always reminds us to be strong and continuously following up the progress our study.

To our parents who never failed to understand us and for giving us emotional and financial support.

And lastly, to our friends and classmates who stayed with us and for reminding us that we are not alone.

**ABSTRACT**

A pandemic is an epidemic of an infectious disease that has spread across a large region, for instance multiple continents or worldwide, affecting a substantial number of people. In our time, where we currently face an ongoing pandemic, we are required to wear face masks to avoid spreading viruses or lessen the chances of being affected by one. With our current situation, wearing face masks affects some of the ways we do things like having to adapt and do extra things to make it work than how it is usually done.

Due to this current COVID-19 Pandemic, having physical contact to anyone and even machineries are very risky. You can get infected with just touching the objects that an infected person had a contact with. With this in mind, the researchers decided to have this study and create a verification system wherein it can detect your face with or without facemask covering your face. With our system, having to have students’ identity confirmed in the university would make it easier, safer, reliable and compliance to protocols in mitigating the transmission of infectious diseases.

**Keywords :**

Pandemic, verification, face recognition, deep learning, convolutional neural network

**TABLE OF CONTENTS**

**Content Page No.**

Title Page 0

Approval Sheet 1

Dedication 2

Acknowledgement 3

Abstract 4

Table of Contents 5

List of Figures 7

Lists of Table 8

**Chapter I – INTRODUCTION**

* 1. Project Context 9
  2. Purpose and Description of the Project 10
  3. Objectives of the Project 11
  4. Scope and Limitations of the Project 11

**Chapter II – REVIEW OF RELATED LITERATURE**

2.1 Related Literature/ Theoretical Background 13

2.2 Related Studies 13

**Chapter III- TECHNICAL BACKGROUND**

3.1 Technicality of the Project 15

3.2 Details of the Technology to be Used 16

3.3 How the Project will Work 17

**Chapter IV- METHODOLOGY**

4.1 Requirements Specification 19

4.1.1 Operational Feasibility 19

Fishbone Diagram 19

Functional Decomposition Diagram 20

4.1.2 Technical Feasibility 21

Compatibility Checking 21

Relevance of the Technologies 21

4.1.3 Schedule Feasibility 21

4.1.4 Economic Feasibility 22

Cost and Benefit Analysis 22

Cost Recovery Scheme 22

4.1.5 Requirements Modeling 23

Object Modeling 23

4.1.6 Risks Assessment Analysis 25

4.2 Design

4.2.1 Output and User Interface Design 25

4.2.2 Data Design 28

4.2.3 System Architecture 29

4.3 Development

4.3.1 Software Specification 30

4.3.2 Hardware Specification 30

4.3.3 Program Specification 31

4.3.4 Programming Environment 31

4.3.5 Deployment Diagram 32

4.3.6 Test Plan 32

4.4 Testing

4.4.1 Unit Testing 33

4.4.2 Integration Testing 36

4.4.3 Compatibility Testing 36

4.4.4 Performance Testing 36

4.4.5. Stress Testing 38

4.4.6 Load Testing 38

4.4.7 System Testing 38

**Conclusion and Recommendations** 39

**Implementation Plan**  40

**Bibliography**  43

**Appendices**  45

**Curriculum Vitae** 96

**Glossary** 100

**LISTS OF FIGURES**

Figure 1 System Architecture

Figure 2 Student Registration Process

Figure 3 Process of Face Recognition and Verification

Figure 4 Fishbone Diagram

Figure 5 Functional Decomposition Diagram

Figure 6 Gantt Chart

Figure 7 Requirements Modeling

Figure 8 Use Case Diagram

Figure 9 Sequence Diagram

Figure 10 Activity Diagram

Figure 11 Main Interface for face Detection and Recognition

Figure 12 Face Recognition running

Figure 13 Entity Relationship Diagram

Figure 14 Network Model

Figure 15 Deployment Diagram

Figure 16 Model Accuracy

Figure 17 Model Loss

Figure 18 Screenshot of a successful verification without facemask

Figure 19 Screenshot of a successful verification with facemask

**LIST OF TABLES**

Table 1 Cost and Benefits Analysis

Table 2 Cost Recovery Scheme

Table 3 Risks Assessment Analysis

Table 4 Data Dictionary

Table 5 Test Plan

Table 6 CSV File containing the predictions

Table 7 CSV File containing the predictions with mismatch

Table 8 Compatibility Test on the Devices

**Chapter I**

**INTRODUCTION**

* 1. **Project Context**

Originating within the city of Wuhan, Hubei Province, China, a novel coronavirus has resulted in an ongoing outbreak of viral pneumonia and has spread extensively all over the world. The World Health Organization (WHO) has already categorized coronavirus disease 2019 (COVID-19) as a pandemic affecting countries and territories and causing continuous fears and deaths to people. This disease is a respiratory infection with varying severities, ranging from the common cold to serious respiratory disease. This virus can spread from person to person with symptoms such as a cough, fever, aches and pains, nasal congestion, runny nose, sore throat or diarrhea and in more severe cases, pneumonia (Katzer, 2020).

Knowing that COVID-19 could also be easily spread through hand contact between people or with surfaces that people often in contact such as doorknobs, countertops, keyboards, and phones. Health agencies everywhere recommended that people refrain from touching their faces and clean their hands after touching public surfaces. The pandemic has generally affected some of the ways we do things like having to adapt and do extra things to make it work than how it is usually done.

Almost everyone wears a mask during the COVID-19 coronavirus epidemic. Face recognition techniques, the most important means of identification, have nearly failed, which has brought huge dilemmas to authentication applications that rely on face recognition, such as community entry and exit, face access control, face attendance, face gates at train stations, face authentication based mobile payment, face recognition based social security investigation, etc. In particular, in the public security check like railway stations, the gates based on traditional face recognition systems cannot effectively recognize the masked faces but removing masks for passing authentication will increase the risk of virus infection. Because the COVID-19 virus can be spread through contact, the unlocking systems based on passwords or fingerprints are unsafe. It is much safer through face recognition without touching, but the existing face recognition solutions are no longer reliable when wearing a mask (Wang & Ebrahim 2020; Khan 2021).

In an institution like Southern Leyte State University Main Campus, student’s identification is a necessary component and with this regard, a Smart Walkway Gate Machine has already been installed and being used. Students are required to bring their ID when entering the campus. The ID will be then swiped to the machine to open the gate. Upon swiping the ID, an image of the user will be shown on the screen nearby. However, verification of student is quite a challenge since students will be wearing facemasks and the guard won’t be able to verify if the bearer of the ID is the actual owner.

During this time of pandemic, the university has been strict towards its students and guest. And because of that, the campus adapted to the use of QR system that is currently being used in the municipality. However, the same problem might be encountered, the guard are not sure whether the bearer of the QR code is the owner.

To solve above mentioned difficulties, it is necessary to improve the existing face recognition approaches that heavily rely on all facial feature points, so that identity verification can still be performed reliably in the case of incompletely exposed faces.

With this in mind, the researchers decided to have this study and create a verification system where it can detect your face with or without a face mask covering your face. With our system, having to have students’ identity confirmed in the university would make it easier, safer, reliable and compliance to protocols in mitigating the transmission of infectious diseases.

* 1. **Purpose and Description of the Project**

COVID-19 is an infectious disease that can be acquired through close contact with an infected person. With this reason, minimal contact to other person is encouraged to avoid the spreading of virus. Since traditional verification needs close contact to verify the identity of a student who wants to enter the campus, it can be a potential risk for spreading the said virus. As a result, we made a system that can verify a student without any close contact using face recognition. This project can help minimize the spread of virus and can help maintain the health and security of the students.

The proponents of this project are tasked to gather images from several students from Southern Leyte State University – Main Campus which will be used to train the system so that it can correctly recognize a student’s face. The system aims to verify the identity of a student without taking their face masks or face shield off to lessen the chance of them being infected with COVID-19. The user/student will face the camera and let the machine scan the student’s face to confirm if the student is an SLSU student. The student does not need to physically interact with the machine, like clicking buttons which is a great way to avoid contracting the virus.

**Objectives of the Project**

**General Objective**

The main objective of the study is to design and develop a SLSU Gate Access Control System using Face Recognition (with or without facemask or face shield) to provide an effective student verification system in which no physical interaction would be made to lessen the chances of the users to be infected with the virus during this time of pandemic.

**Specific Objective**

1. To design and implement an efficient storage and retrieval of verified students.

2. To implement algorithm for training the classification model.

3. To design and develop a user-friendly interface for the face recognition student verification system

4. To evaluate the performance of the system in recognizing students.

**1.3 Scope and Limitations of the Project**

This project primarily focuses on the student identity verification processes of Southern Leyte State University – Main Campus with facial recognition features that recognizes registered users with or without face masks and is not capable of recognizing digital faces that are not included during the training of the classification model. New face images, however, can be incorporated into the system by retraining the classification model.

The data collection of the project will be conducted in Southern Leyte State University – Main Campus. The design and development of the application will be conducted at the College of Computer Studies and Information Technology from August 2021 to April 2022.

**Chapter II**

**REVIEW OF RELATED LITERATURE**

**2.1 Related Literature/ Theoretical background**

Face recognition technology has made many developments in the world, it also became a widely used biometric technology mainly because it possesses the merits of both accuracy and low intrusiveness [1]. Access control using face recognition has some advantages compared to other biometric systems as it does not require expensive or specialized equipment. The system can be built using a simple video camera and a personal computer [2]. It is also a non-contact type of biometric technology where the person can just face the camera and it will be recognized and detected by the system. This type of technology is easy to use and easy to detect trespassers. Access control based on Face recognition will be able to recognize faces who can access a certain location and will detect trespassers which can help secure the area.

There have been numerous approaches that has been used for face recognition over time, but recently deep learning became a popular approach as it solves complex problems and demonstrates promising improvements involving face detection.

**2.2 Related Studies**

The study of Januzaj et al. [3] proposed a real-time access control based on face recognition using HAAR and Principal Component Analysis (PCA) algorithm. HAAR-like features have been used for face detection and PCA has been used for face recognition. The study shows that factors like background, lighting and number of photos affects the accuracy rate of recognizing faces. There has been 62% accuracy rate of recognizing faces if all the factors are controlled.

Ibrahim et al. [4] also proposed an automated face recognition system for office door access control where it uses Eigenfaces based on Principal Component Analysis (PCA) and Artificial Neural Networks algorithm. The study includes analysis of 3 main factor of recognizing faces – illumination, head orientation and distance on the development of the system. Result shows that the system achieved good performance of recognizing faces having 80% accuracy rate when subject is between 40 to 60 cm camera distance, and head angle is within range of -20 to +20 degrees.

Another study of Shah et al. [5] proposed an intelligent security system based on face recognition using Voila-Jones algorithm for face detection, PCA algorithm for face feature extraction and Eyeglass Detection and Removal algorithm. The proposed system can recognize single or multiple faces from an image captured in different imaging conditions, different face orientation and even if with glasses on. The system has been tested and compared from other face recognition algorithm and other well-known face datasets. The study concluded that the proposed system achieved 97% face recognition accuracy rate and the algorithms used in the system has higher accuracy rate compared to other face recognition algorithms.

Additional study of Kiritbhai et al. [6] proposed an auto gate pass system using face recognition base on real-time video processing. The algorithm used in the study is deep learning algorithm. The process would be the student would face the camera so that the system will be able to detect and capture the student’s face. Once the system detects the face, it will compare the details which is stored in the database and recognize the face. After the system recognizes the face it will make the entry into e-register and would send a notification to the student. Result shows that the proposed system has high recognition accuracy rate and is useful to grant gate pass to students and is a great tool for the security of the university.

Having gone through all the mentioned literature and studies, deep learning has proved many successful attempts of making different applications using Face Recognition. These studies inspired the proponents to make a verification system that uses face recognition as means for following health protocols for the pandemic.

**Chapter III**

**TECHNICAL BACKGROUND**

**3.1 Technicality of the Project**

This study will be implemented as a Computer-Based software, where it can executable without the use of internet connection. This can be achieved through the help of the model that will be created to train the acquired datasets to recognize and detect faces. The researchers will use PyQt5 and Python Programming language for the design and implementation of the verification system.

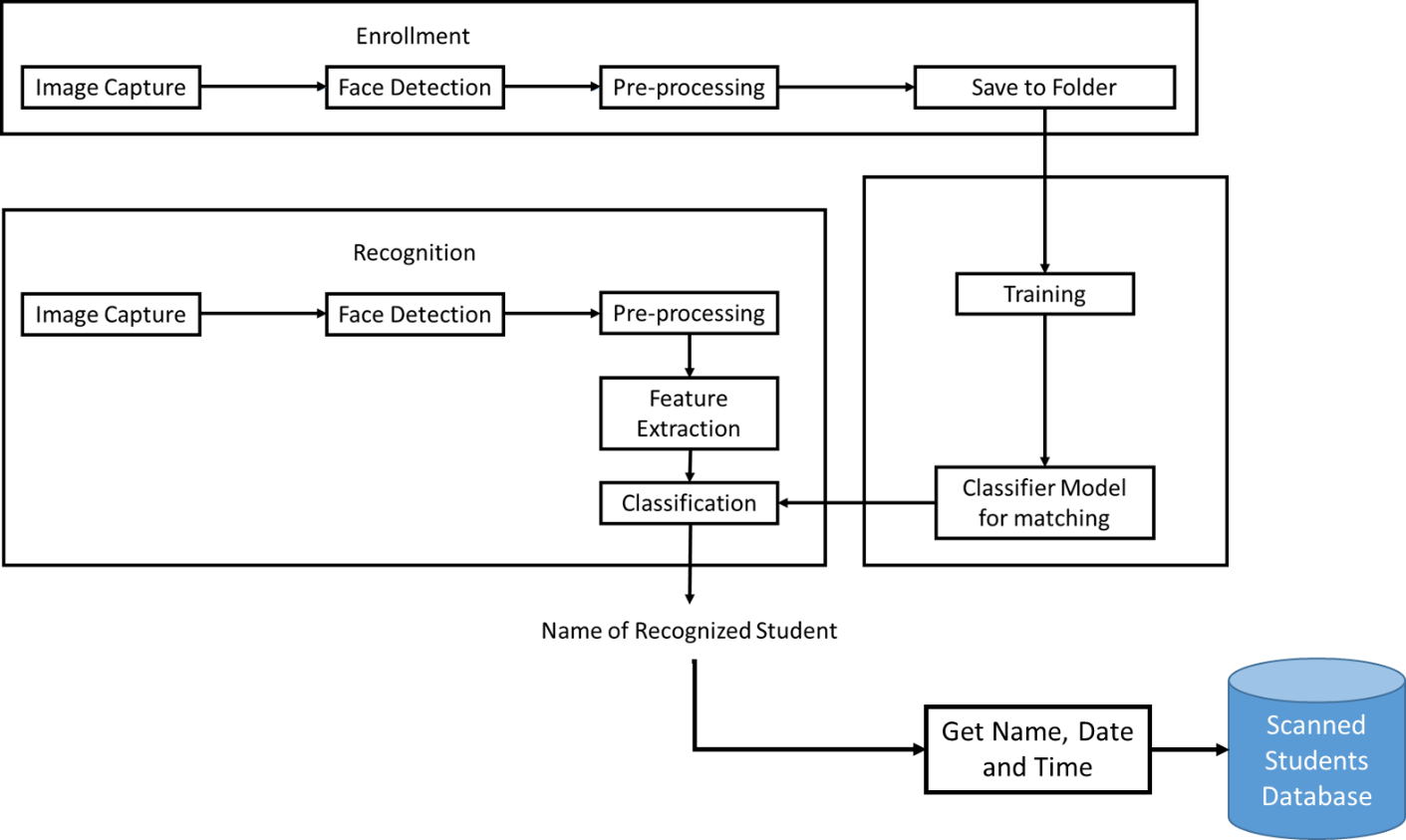


Figure 1. System Architecture

Figure 1 is the system architecture of the SLSU Gate Access Control System using Face Recognition. For the new users, they will first need to be enrolled to the system by having their photo either submitted by them or capture using the system webcam. The collected photos will be saved by the admin in a folder with their ID number as the folder name. Then they will be added to the list to be trained as the next batch for the model. The photos will be used for the system training, and as a result a classifier model will be create which will be used to recognize the student. The student will just need to face the webcam and system will detect the face. The image or the face detected will be processed and using the classifier model, the student will be then recognized by the system. After the student has been identified, it will be then saved to the database. A database will serve as the storage for the record of students that has been identified by the system.

Language: Python

Libraries: OpenCV, DLib, NumPy, PyQt

Platform: Computer-based

Database: MySql

**3.2 Details of the Technologies to be Used**

Here is the list of technologies to be used by the researchers to create the project;

* **Python 3.8**
* **XAMPP –** used for accessing MySql database.
* **Operating System** – This is a must for a computer to work with its intended function. We would use Windows 10 for our project.
* **MySql** – We would use SQLite as the storage for the registered and scanned students for our system.
* **Camera** – This is another critical component of the system because will detect and recognize faces through this input device. You may use the built-in camera of your laptop, but for this system, we would use an external camera connected to the laptop.
* **PyQt –** It is a Python binding of the cross-platform GUI toolkit Qt, implemented as a Python plug-in.
* **Qt Designer** – Used to design and build GUIs for our system.
* **Jupyter Notebook** – Used for coding the training and classification of the model.
* **Keras** – It is a Python library used in creating deep learning models.
* **Tensorflow** – A library for multiple machine learning tasks.
* **OpenCV –** OpenCV stands for Open-Source Computer Vision and is a library that focuses on real-time computer vision. This will be used for its functions for image processing, identify objects, and in our case, identifying faces.
* **NumPy** – NumPy or Numerical Python, is a library that consists of functions that processes array. Through this, mathematical and logical operations can be applied to arrays. This library will be used to convert images of the student’s faces to a form of array to store trained models.

Installations and usage of the libraries and packages will be required to support this project and make it viable for actual deployment.

**3.3 How the Project Will Work**

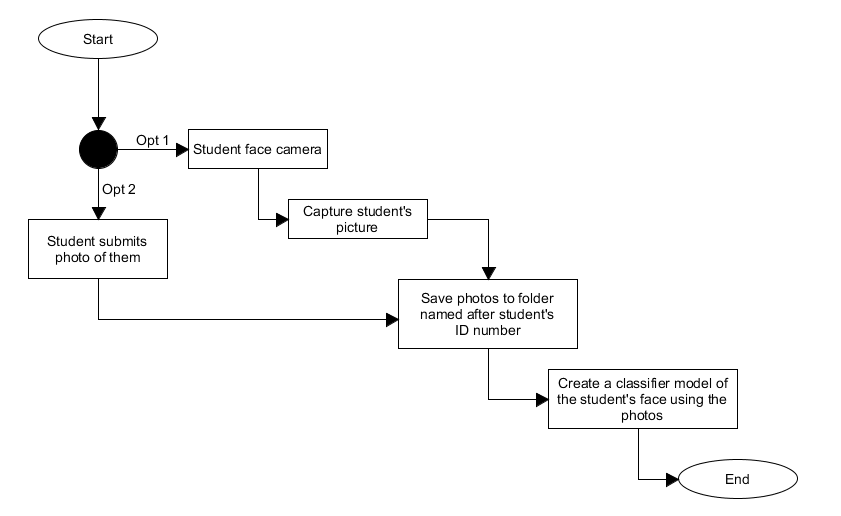
The project will be a computer-based application which the user will need to provide an image of themselves for face recognition. But before the process of recognition, the user should be registered first to the system. Below is the process of student registration for the face recognition verification.

Figure 2. Student Registration Process

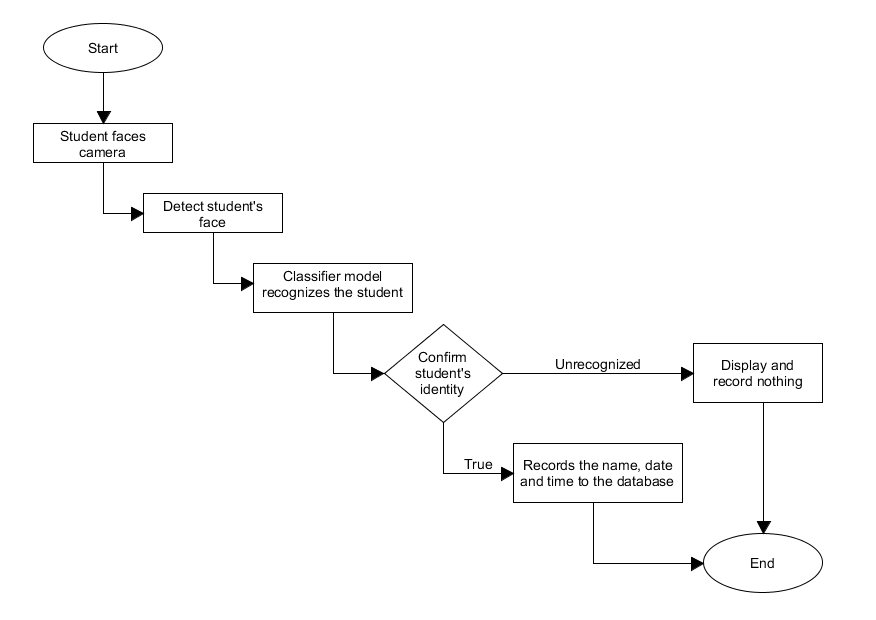
To start, a student is required to submit their photo or have their faces captured by the system. If they choose the latter, they will have to face the camera, then have a picture of them taken. An admin would be present whether to input or overlook the process on saving the student’s photo and name. The photo acquired would be then used to create a model that will identify them the time they use our system. The photo stored will then be used for the next batch of training and updating the model with the current trained samples.

Figure 3. Process of Face Recognition and Verification

Figure 3 shows process of recognizing the student using the system. To take the student’s identification via the system, the student must place himself/herself at the specified location and face at the camera to be detected by the system. The system then will try to detect a face through the camera and the classifier model will be recognizing the student. If the system finds a match, the system will add the student’s name to the record that he/she has accessed the system along with the time and date the student was scanned. If the system doesn’t recognize the student, it will not show any details and will record nothing.

**Chapter IV**

**METHODOLOGIES**

**4.1 Requirements Specification**

**4.1.1 Operational Feasibility**

***Fishbone Diagram***

Students can enter the campus using other student’s ID

Machine

People

Environment

Method

Security Guards relies on machine.

Only requires the student ID as verification.

Only one verification process for entering the campus.

Mandatory wearing of masks due to pandemic

Fails to recognize if the bearer of the ID is the actual owner.

Does not scan student face.

Relies on ID scan verification

Figure 4. Fishbone Diagram

Figure 4 shows the causes on how students can enter the campus even if they are not students of the university.

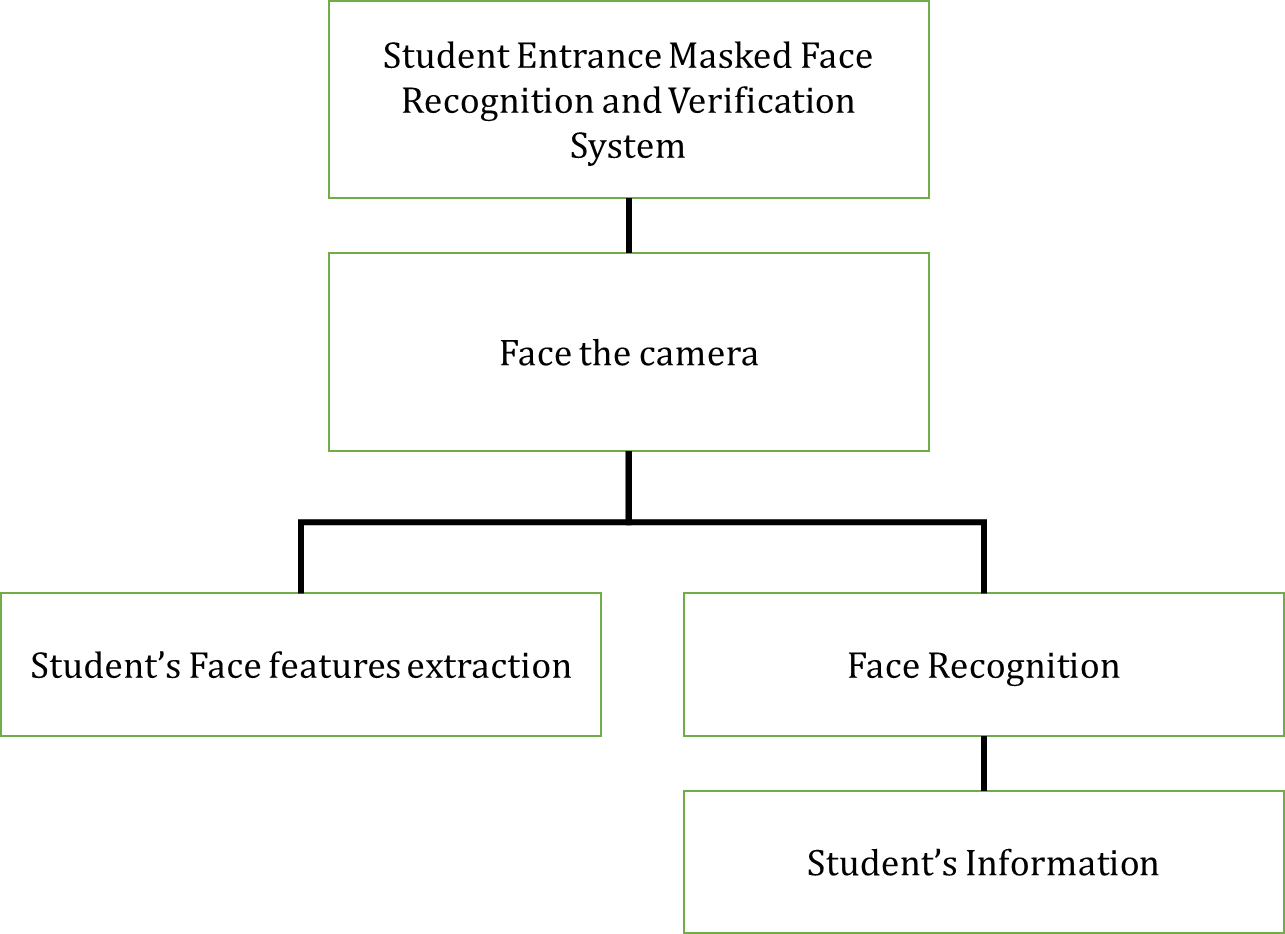
**** ***Functional Decomposition Diagram***

Figure 5. Functional Decomposition Diagram

The functionalities of the Student Entrance Masked Face Recognition and Verification System are show in Figure 5. The main objective of the project is to recognize and identify a student via face recognition and their corresponding name, ID number, course and year.

*Face the camera*

This phase will prompt the user to face the camera for the system to capture and process. Users will be asked to reposition when their face can’t be recognized.

*Student’s Face feature extraction*

Features are parts of a face in an image that help to predict or identify the image that has been captured. Image feature will be executed with the help of the CNN model classifier to recognize the face of the user.

*Face Recognition*

This functionality of the system executes after the image extraction execution is successful. Detected features are classified by the CNN model to recognize the student captured from the camera.

*Student’s Information*

This phase of the system executes after the recognition of the face detected. The information of the student/user is being queried from the database. Student’s information contains the name of student, ID number, course and year.

**4.1.2 Technical Feasibility**

***Compatibility Checking (Hardware/Software)***

The application can be installed in any computer and laptop devices with operating systems with Windows version 7, 8 and 10. The application software can be executable without internet connection as long as the prerequisite libraries and conditions are met.

***Relevance of Technologies***

A computer/laptop and camera are the needed devices for the application to function and execute properly, as this project is constructed with the purpose of detecting and recognizing faces used for security purposes for the school.

**4.1.3 Schedule Feasibility**

Figure 6. Gantt Chart

The figure 6 depicts how this study was carried out, as well as the period and tasks complete by the proponents.

**4.1.4 Economic Feasibility**

***Cost and Benefit Analysis***

|  |  |
| --- | --- |
| **Expenses** | **Amount** |
| Internet Expenses | 2,000 |
| Paper and Photocopy Expenses | 600 |
| Transportation | 480 |
| Miscellaneous | 350 |
| TOTAL | 3,430 |

Table 1. Cost and Benefits Analysis

Table 1 reflects the lists of the total expenses incurred during the creation of the project. The table reflects that the Internet Expenses has the highest expense due to the proponent’s need to search reference articles, proposals, software and other platforms that can be useful for the project’s completion.

***Cost Recovery Scheme***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Expenses** | Aug | Sept | Oct | Nov | Dec | Jan | Feb | Mar | Apr |
| Internet Expenses | 200 | 200 | 400 | 400 | 0 | 0 | 0 | 400 | 400 |
| Paper and Photocopy | 0 | 100 | 100 | 0 | 0 | 0 | 0 | 200 | 200 |
| Transportation | 0 | 120 | 120 | 0 | 0 | 0 | 0 | 120 | 120 |
| Miscellaneous | 0 | 100 | 100 | 0 | 0 | 0 | 0 | 50 | 100 |
| TOTAL | 200 | 520 | 720 | 400 | 0 | 0 | 0 | 770 | 820 |

Table 2. Cost Recovery Scheme

Table 2 reflects the division of expenses from Table 1 in order to pay the costs incurred during the creation of the project. The internet cost and paper and photocopy costs had gradually increased every month as the development of the project progressed.

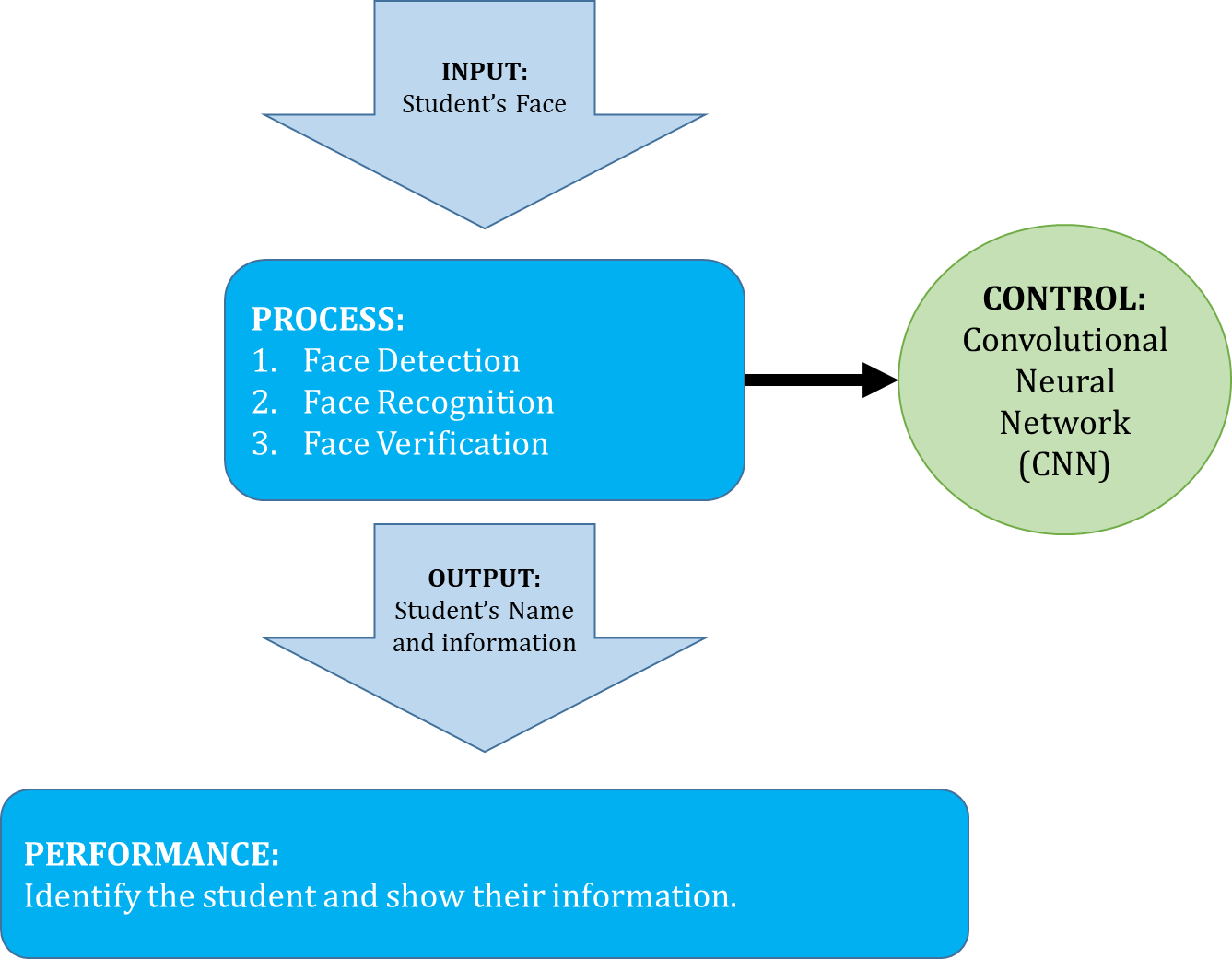
**4.1.5 Requirements Modeling**

Figure 7. Requirements Modeling

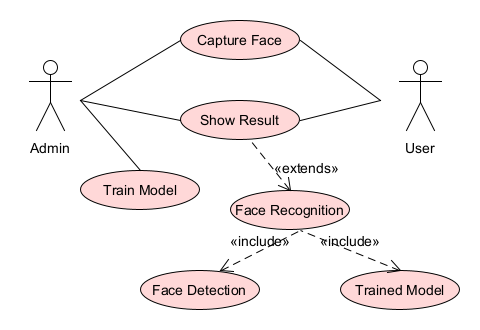
 ***Object Modeling***

Figure 8. Use Case Diagram

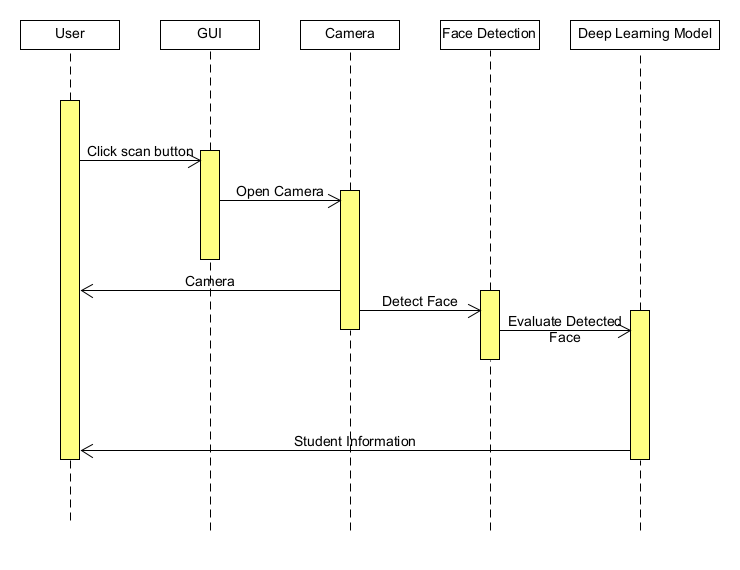
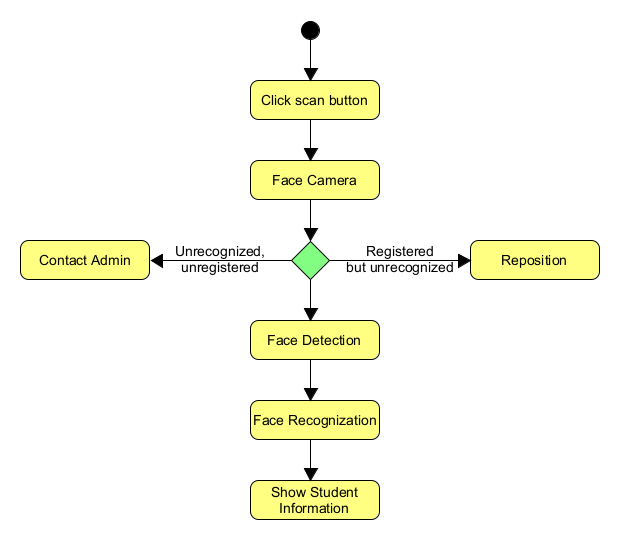
Figure 9. Sequence Diagram

Figure 10. Activity Diagram

**4.1.6 Risk Assessment Analysis**

|  |  |  |
| --- | --- | --- |
| **Risk** | **Possible Conflict** | **Mitigation Method** |
| User were not yet registered | Since the user’s face is not yet included in the training, it may result to failure in prediction. | Applying warning message to for the user that he/she is unrecognized. The user will be asked to contact admins and get registered. |
| Dim surroundings | The student’s face feature may not be visible and the system may not be able to detect or recognize the student. | Pull off some warning messages that the lighting is too dark, turn-on nearby lights if possible.  Set-up should be installed in a well-lit area. |
| User is too far from the camera | There might be problems in detecting the student’s face and will not be able to recognize the student. | Display a warning message that the student is too far, should be just within 2 meters. |

Table 3. Risks Assessment Analysis

**4.2 Design**

**4.2.1 Output and User Interface**

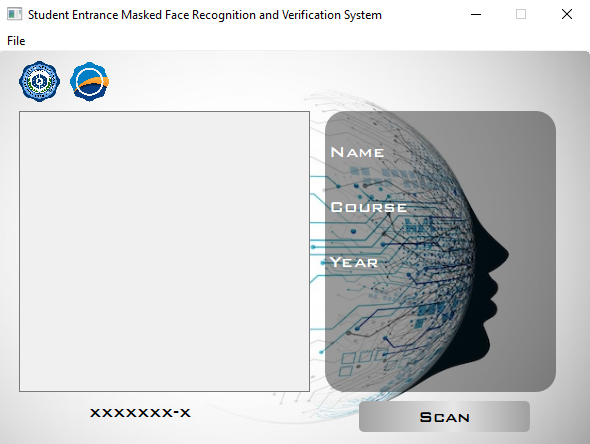
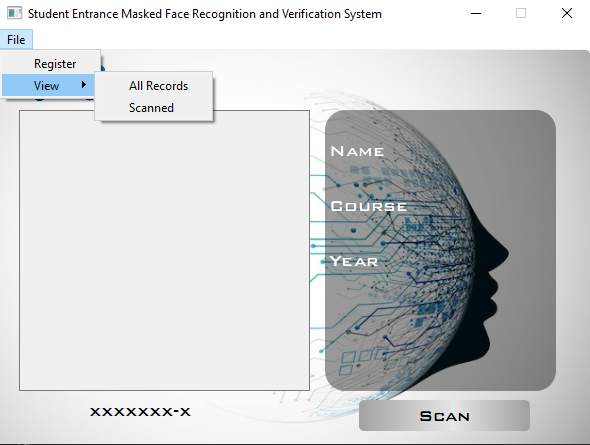
****

Figure 11.A. Main Interface for face Detection and Recognition



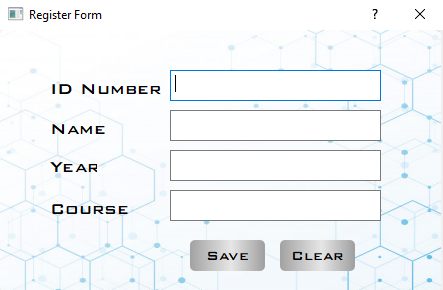
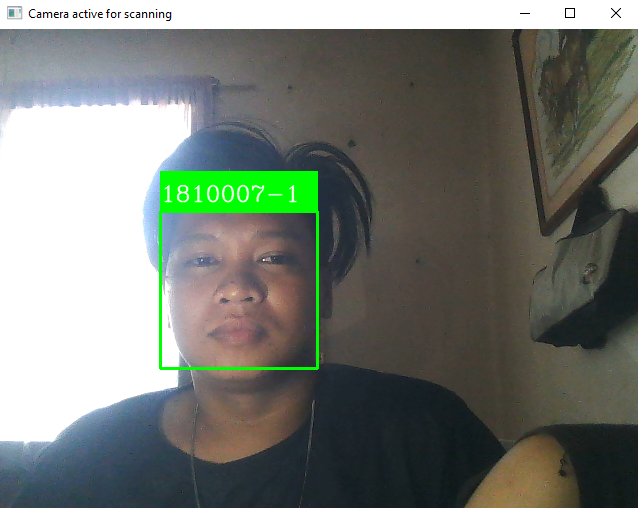
****Figure 11.B Main Interface for Face Detection and Recognition

Figure 11.C Main Interface for Face Detection and Recognition

Figure 11.A, Figure 11.B and 11.C illustrates the main interface of the application with its following function

* Scan button – used to enable the face detection and recognition functions.
* File button – contains another set of buttons as shown in Figure 11.B
  + Register – used to list students who are not yet registered
  + View (All Records) – used to view all student records
  + View (Scanned) – used to view the record of scanned students



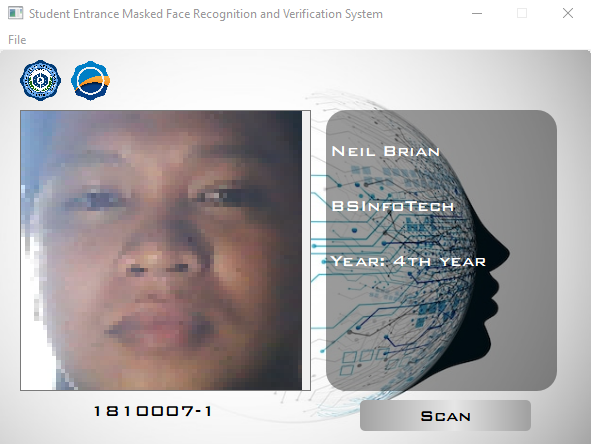
Figure 12.A Face Recognition running

Figure 12.B Face Recognition running

Figures 12.A and 12.B shows the running application capturing and recognizing one of the test samples.

**4.2.2 Data Design**

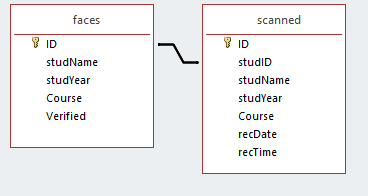
****

Figure 13. Entity Relationship Diagram

***Data Dictionary***

|  |  |
| --- | --- |
| **Keywords** | **Description** |
| faces | This is the table name containing all the registered data of the students. |
| scanned | It is a table name containing all the scanned faces of the recognized students |
| ID | It is a term used to described the primary key of each table |
| studName | It is a term used to describe the student’s name |
| studYear | It is a term used to describe the student’s year |
| Course | It is a field in the table that contains the course of the students |
| Verified | A table field that contains the trained status if the student’s face was included in the model training. |
| recDate | A table field containing the recorded dates of the when the student was scanned by the system. |
| recTime | A table field containing the recorded time of when the student was scanned by the system. |

Table 4. Data Dictionary

**4.2.3 System Architecture**

***Network Model***

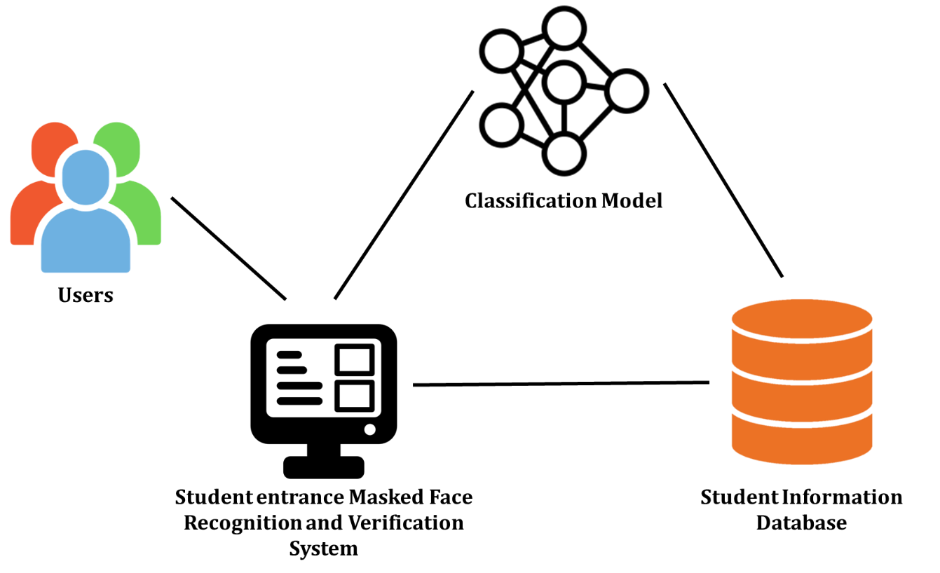
****Figure 14. Network Model

Figure 14 represents the relationship of how the user can access the application, the figure represents where the user interacts with as they use the application.

***Security***

This project does not require intensive security feature because the user data can be only accessed and modified by the system admins. The functionality of the application of this project just simply requires the users to face the camera and have their faces scanned for recognition and verification. The application could still have the participation of an overseer, especially when uploading new student data and updating the trained model.

**4.3 Development**

**4.3.1 Software Specification**

The software that has been used for the development of the application were as follows:

* **Jupyter Notebook**, this was used for coding and training the classification model.
* **Tensorflow**, it is a library used for multiple machine learning tasks.
* **VS Code**, it is a source code editor with features that include but not limited to debugging, syntax highlighting, intelligent code completion, snippets, code refractoring and embedded GIT.
* **Python**, it is a high-level general-purpose programming used for integrating the GUI and the trained classification model.
* **XAMPP**, an open-source cross-platform web server that we used for MySQL database.
* **Qt Designer**, this was used for the creation of the GUI using PyQt applications.
* **Keras,** it is a Python library used in creating deep learning models.

**4.3.2 Hardware Specification**

The hardware components that were used for creating this project were as follows:

* **Laptop**

ROG Strix-G

Intel Core i5 9th Gen

8 GB RAM

Windows 10

* **Camera**

1080p

**4.3.3 Program Specification**

The project is expected to detect and recognize a face from the registered users either wearing a face mask or not, and in this case, it will be used as an access control for the school gate.

This project will be able to record the student’s registered data when they try to use the system. This project will also use the trained model to recognize the student accessing the system and will display their saved data to GUI as proof that they are a student of the school.

**4.3.4 Programming Environment**

***Front-End***

The project application front-end code was created using Python programming language together with PyQt for the user interface. PyQt allows the codes from the Qt Designer to be attached to the python program in order to give a user interface that can be easily interacted by the user.

***Back-End***

The back-end environment used for the creation of the project is python with tensorflow and keras for training and acquiring the classification model used for the face recognition application.

MySql database was also used to store the list of the registered data model and to save the capture and recognized images of the students.

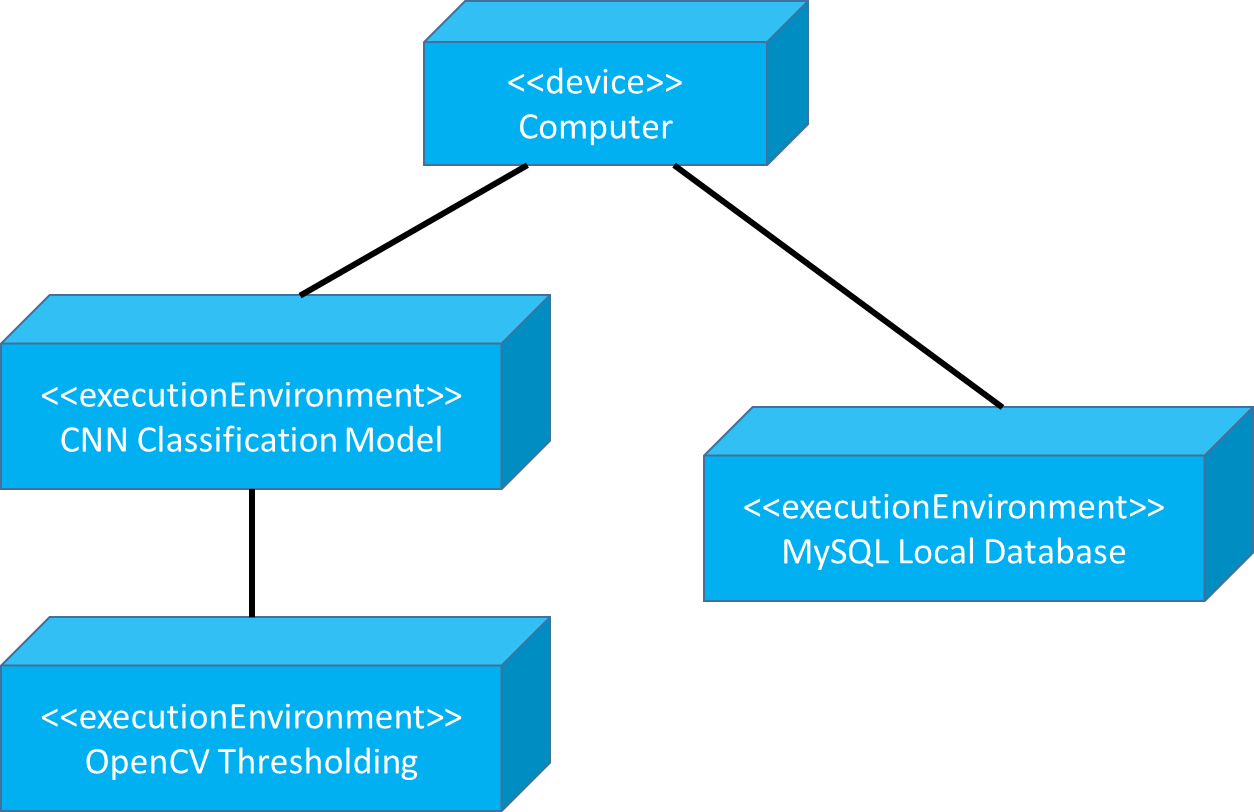
**4.3.5 Deployment Diagram**

Figure 15. Deployment Diagram

**4.3.6 Test Plan**

|  |  |  |
| --- | --- | --- |
| TYPE OF TESTING | ACTION PLAN | DURATION |
| Unit Testing | Test the system’s function such as the model performance and segmentation process | Within  November 2021 |
| Integration Testing | Test the integration of the Graphical user interface and the model | Within  November 2021 |
| Compatibility Testing | Test the application running capability on different platforms | Within  December 2021 |
| Performance Testing | Test the performance of the system in terms of sensitivity, reactivity and  stability under a particular workload. | Within  February 2022 |
| Stress Testing | Test the robustness of the software  beyond the limits of normal operation. | Within  March 2022 |
| Load Testing | Test the usage of the software program  by multiple users accessing the  program concurrently. | Within  March 2022 |
| System Testing | Test the complete and fully integrated  software product. | Within  April 2022 |

Table 5. Test Plan

**4.4 Testing**

**4.4.1 Unit Testing**

The model was created from a total of 2989 training samples from 57 different classes. The classes were gathered from all the students enrolled on the main campus, regardless of the year and course. The model was trained with 292 validation samples and trained with 500 epochs. The model created procured about 0.9948, or 99%, as the highest training accuracy achieved. Figure num> shows a graph of the classification model training plot for the last 20 epochs because the training time was not consistent and was mostly interrupted.

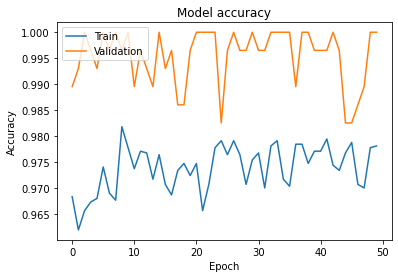
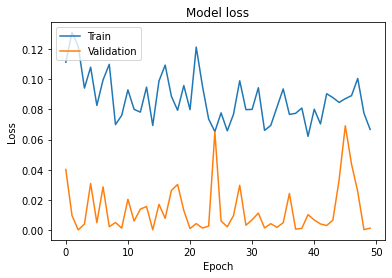
****The graph shows the training and validation accuracy was progressing close to 1.0.

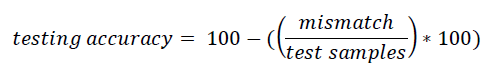
Figure 16. Model Accuracy

Figure 17. Model Loss

Upon getting the plot points and the highest accuracy value, the model has undergone a testing phase to test its accuracy and performance. The model was tested by loading the Keras model (see appendices). There are a total of 298 test samples used for testing the model in which these samples were not included in the training datasets. By using the testing code (see appendices), the model generated a comma-delimited file (.csv) containing the lists of predictions and the mismatched data samples.

Table 6. CSV File containing the predictions

Table 7 shows the list of predictions as well as the mismatched samples of the model training. The number of images in each class vary depending on how many images they sent to the researchers. The average image of each class is 5 representing each student. There was a total of 2 mismatched samples highlighted in red (see Table 7) and the rest are all accurate predictions of the model.

Table 7. CSV File containing predictions with mismatch

To calculate the testing accuracy, a formula was provided (see formula above). Using the formula, divide the number of mismatch samples (see highlighted samples in Table 7) over the number of test samples which is 298. The testing accuracy results to 99.33 percent compared to the highest value achieved during the previous testing which is 99.48 percent. Therefore, the model was good enough to be used to its graphical user interface.

**4.4.2 Integration Testing**

|  |  |  |
| --- | --- | --- |
| Device Type | Model | Specification |
| Laptop | ROG Strix-G | Windows 10, i5 9th Gen |
| Laptop | Acer | Windows 10, Intel Core 5 7th Gen |
| Laptop | Predator Helios | Windows 10, Intel Core i5 8th Gen |
| Computer |  | Windows 10, Ryzen 5 3600 |

The integration of the Graphical User Interface and the model is one of the most essential steps of making sure that the project is functioning for what it is intended to do. A small batch of testing samples were used to test the model if the model would still function as intended as it was integrated to its graphical user interface. The integration proved to be successful for it displayed the same result as the model provided before integration of the GUI.

**4.4.3 Compatibility Testing**

The application was tested in different devices with different specifications to check the capability to operate as it was intended. The application was installed to the following devices:

Table 8. Compatibility Test on the Devices

**4.4.4 Performance Testing**

After knowing that the compatibility of the application to different devices was successful, a performance test was conducted to measure the response time, reliability, scalability, stability, and any gaps in the application. During this phase, a student who has contributed images for the training was requested to test the system if it can capture and verify a user. The test was successful as it verified the user accurately.

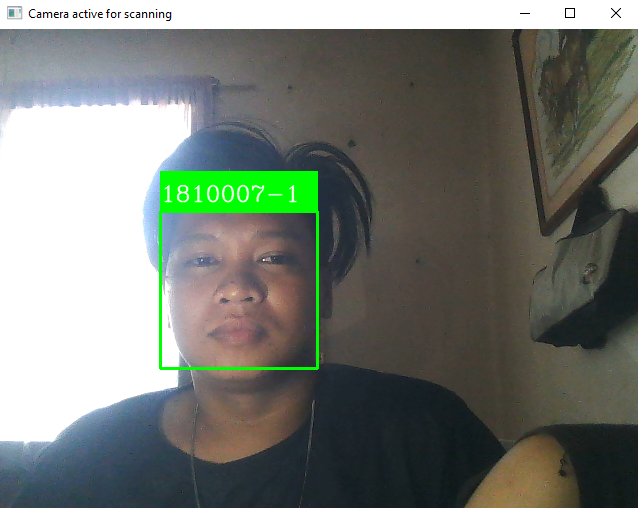
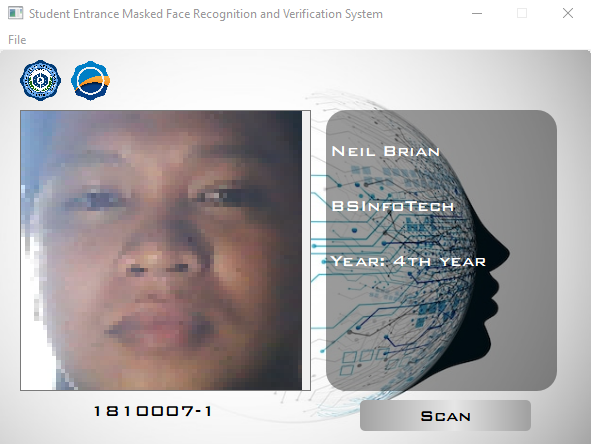
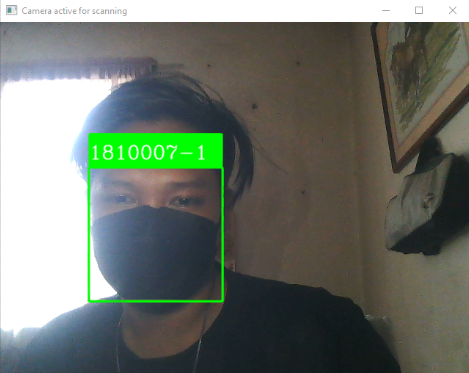
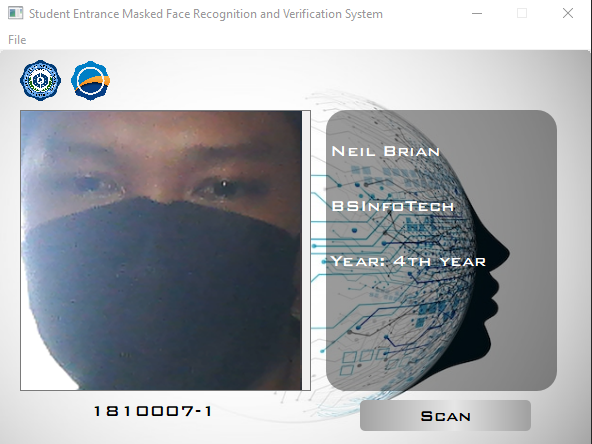
Figure 18. Screenshot of a successful verification w/out facemask

Figure 19. Screenshot of a successful verification with facemask

Figure 18 shows the sample output of face recognition when tested by a student contributor. The student was detected easily by the system and was able to recognize the student correctly.

**4.4.5 Stress Testing**

During this testing stage, the software was tested by running the application and testing its face detection and recognition with different environment settings. It was tested if it would still detect faces and recognize them while the environment was dim, too bright, a picture was blurry, the distance of a face from the camera was too great, displaying objects instead of faces, and consecutively showing and switching a trained face and an untrained face.

**4.4.6 Load Testing**

During this testing stage, it was simulated through having different and multiple faces directed at the camera to test whether it would capture each face and identify them accordingly and save them to the list of users who have been scanned by the system.

**4.4.7 System Testing**

Upon the creation of this project, there were several testing stages performed to test the capability and the overall function of the system. As all of the testing was successful, the overall system testing was also successful. The classification ability of the system was able to return the expected result.

**CONCLUSION AND RECOMMENDATIONS**

The researchers were able to successfully developed the Student Entrance Masked Face Recognition and Verification System. Python was the primary used in the analysis of the dataset in establishing optimized parameters for the Convolutional Neural Network model. It was then implemented in Keras and integrated in a computer based platform. The generated model obtained an overall accuracy rate of 99% which indicates that the model is reliable in predicting the students’ identity. Upon testing, the application was able to recognize the student’s identity whether they are wearing a face mask or not.

However, this study still needs further improvements to apply, such as auto adjustment of exposure when its senses that the image is either over-exposed or under-exposed. The face classifier will also need improvement for better detection. This enhancement will be very helpful for application in recognizing the student’s face accurately.

**IMPLEMENTATION PLAN**

**Project Implementation Checklist**

|  |  |  |
| --- | --- | --- |
| **Statement Type** | **Include** | **Exclude** |
| Executable | **/** |  |
| Non-executable | **/** |  |
| Declarations | **/** |  |
| Compiler Directions | **/** |  |
| Comments | **/** |  |
| **How Produced** | | |
| Programmed | **/** |  |
| Generated with source code generators |  | **/** |
| Converted with automated translators |  | **/** |
| Copied of reused without change |  | **/** |
| Modified | **/** |  |
| Removed | **/** |  |
| **Origin** | | |
| New Work: no prior existence | **/** |  |
| Prior Work: taken or adapted from |  | **/** |
| A previous version, build or release |  | **/** |
| Commercial off-the-shelf software |  | **/** |
| Government furnished software |  | **/** |
| Another product |  | **/** |
| Other commercial library |  | **/** |
| A reuse library (software designed for reuse) |  | **/** |
| Other software component or library |  | **/** |
| **Usage** | | |
| In or as part of primary product | **/** |  |
| External to or in support of primary product |  | **/** |
| **Delivery** | | |
| Delivered | **/** |  |
| Delivered as source |  | **/** |
| Delivered in compiled or executable form | **/** |  |
| Not Delivered |  | **/** |
| Under configuration control |  | **/** |
| Not under configuration control |  | **/** |
| **Functionality** | | |
| Operative | **/** |  |
| Inoperative |  | **/** |
| Functional (Intentional dead code, reactivated for special purposes) |  | **/** |
| Nonfunctional (unintentionally present) |  | **/** |
| **Replications** | | |
| Master source statements (originals) | **/** |  |
| Physical replications of master statements, stored in master code | **/** |  |
| Copies inserted, instantiated or expanded when compiling or linking |  | **/** |
| **Development Status** | | |
| Estimated or planned | **/** |  |
| Designed | **/** |  |
| Coded | **/** |  |
| Model Performance Evaluation Completed | **/** |  |
| Unit Test Completed | **/** |  |
| Integrated into components | **/** |  |
| Test readiness review completed | **/** |  |
| Software (CSCI computer software configuration item) tests completed | **/** |  |
| Systems test completed | **/** |  |
| **Evaluation** | | |
| User Evaluation Completed | **/** |  |

**Implementation Contingency**

Unforeseen incidents usually occur during project development, which can affect the implementation process of the project. This could create a risk to the project development resulting to failure if not being resolved immediately. As a result, the proponents had made alternative steps to continue the project development if problems may occur.

The proponents have been using Haar-Cascaded Face Detection classifier to detect faces that can be captured by the camera, however if the current classifier will not work, Multi-Task Cascaded Convolutional Network (MTCNN) will be used instead. Although both are good classifiers for face detection, Haar-Cascaded Face Detection are better in detecting edges and lines and can detect faces faster than the MTCNN. When it comes to the algorithm to be used for model training, K-Nearest Neighbors (KNN) algorithm was chosen as an alternative algorithm to be used as this algorithm also works well with OpenCV and can also acquire high recognition accuracy same way as CNN.

Although the alternative methods are already being prepared, but those methods were not used, as the current methods are successfully working well with only minimal problems.

**Infrastructure/Deployment**

Before the deployment of the project, the proponents will prepare the necessary files for the users to be used. The proponents will prepare executable files for installation for the devices that will be used and will make a demo and user’s manual for the users to know how the system will work. The proponents will select a few students that has been registered already by the system to test the functionality and reliability of the project. An evaluation form ISO 9126 (Appendix B) will be given to the users to rate the project’s functionality and usability.

Upon the deployment of the project, selected users will be encouraged to do stress testing to the system as well to see the limitations and the things to be improved on the system. They will be requested to answer the evaluation form honestly and will be asked also for recommendations for improvement of the system. The completed ISO 9126 answered by the users will be analyzed and will be used by the proponents as bases on how to improve the system.

**BIBLIOGRAPHY**

[1] Krishnan, M. G., & Balaji, S. B. (2015). Implementation of automated attendance system using face recognition. *International Journal of Scientific & Engineering Research*, *6*(3).

[2] Bryliuk, D., & Starovoitov, V. (2002). Access control by face recognition using neural networks. *Institute of Engineering Cybernetics, Laboratory of Image Processing and Recognition*, *4*.

[3] Januzaj, Y., Luma, A., Januzaj, Y., & Ramaj, V. (2015, June). Real time access control based on face recognition. In *International Conference on Network security & Computer Science (ICNSCS-15)* (pp. 7-12).

[4] Ibrahim, R., & Zin, Z. M. (2011, May). Study of automated face recognition system for office door access control application. In *2011 IEEE 3rd International Conference on Communication Software and Networks* (pp. 132-136). IEEE.

[5] Shah, M., Shukla, D., & Pandya, D. Smart Gate: Intelligent Security System Based on Face Recognition.

[6] Kiritbhai, P. M., & Jyotsna, C. (2021, June). An Automated University Gate Pass Monitoring System Using Deep Learning. In *2021 International Conference on Communication, Control and Information Sciences (ICCISc)* (Vol. 1, pp. 1-6). IEEE.

Honigsbaum, M. (2009). Historical keyword Pandemic. The Lancet, 373.

Morens, D. M., Folkers, G. K., & Fauci, A. S. (2009). What is a pandemic? J Infect Dis, 200(7), 1018-1021. doi: 10.1086/644537

Wang, Z. (2020, March 20). *Masked Face Recognition Dataset and Application*. ArXiv Vanity. <http://www.arxiv-vanity.com/papers/2003.09093/>

Ebrahim, B. (2020, May 15). *Masked Face Recognition Application - DataDrivenInvestor*. Medium. <https://medium.datadriveninvestor.com/masked-face-recognition-application-9405b397aaea>

Khan, A., Gedekar, A., Khawaja A., Barghat, M., Khan, N. (2021). *Masked Face Recognition Using Convolutions Neural Network*. *International Journal of Innovative Research in Computer and Communication Engineering* (Vol. 9, Issue 5). DOI: 10.15680/IJIRCCE.2021.0905129

**APPENDICES**

**Appendix A**

**RELEVANT SOURCE CODE**

**Model Training Source Code**

**Text

Description automatically generated**

**Graphical user interface, text

Description automatically generated**

**Text

Description automatically generated**

**Python Files Source Code**

from scan import Ui\_mainWindow

from studRegister import Ui\_registerForm

from viewData import Ui\_viewRecords

from viewScanned import Ui\_viewScanned

from labels import setLabels

import sys

import cv2

import numpy as np

import pymysql

from datetime import datetime

import re

from keras.preprocessing import image

from keras.models import load\_model

from PyQt5 import QtWidgets, QtCore, QtGui

from PyQt5.QtWidgets import QMainWindow, QApplication

#db connection

connection = pymysql.connect(host = "localhost", user = "root", password = "", database = "capstone")

cursor = connection.cursor()

facedetect = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

font = cv2.FONT\_HERSHEY\_COMPLEX

cap = cv2.VideoCapture(0)

savedModel = load\_model("D:/Capstone/modelFaces2.h5")

class MyMainWindow(QMainWindow):

def \_\_init\_\_(self, parent=None):

super(MyMainWindow, self).\_\_init\_\_(parent)

self.mainInterface = Ui\_mainWindow()

self.mainInterface.setupUi(self)

#button event scan

self.mainInterface.btnScan.clicked.connect(self.capture\_feed)

#register button

self.mainInterface.actionRegister.triggered.connect(lambda: self.register())

#view database record

self.mainInterface.actionViewRegistered.triggered.connect(lambda: self.viewRegistered())

#view scanned

self.mainInterface.actionViewScanned.triggered.connect(lambda: self.viewScanned())

#register student open form

def register(self):

self.window = QtWidgets.QDialog()

self.ui = Ui\_registerForm()

self.ui.setupUi(self.window)

self.window.show()

#show all registered data in database

def viewRegistered(self):

self.window = QtWidgets.QDialog()

self.ui = Ui\_viewRecords()

self.ui.setupUi(self.window)

self.window.show()

#show records of scanned users

def viewScanned(self):

self.window = QtWidgets.QDialog()

self.ui = Ui\_viewScanned()

self.ui.setupUi(self.window)

self.window.show()

#show webcam

#and identify

def capture\_feed(self):

def saveStudentDT(id, name, year, course, df, tf):

insertSQL = "INSERT INTO scanned(ID, NAME, YEAR, COURSE, DATE, TIME) VALUES (%s, %s, %s, %s, %s, %s);"

val = (id, name, year, course, df, tf)

cursor.execute(insertSQL, val)

connection.commit()

while True:

success, imgOriginal = cap.read()

gray = cv2.cvtColor(imgOriginal, cv2.COLOR\_BGR2GRAY)

faces = facedetect.detectMultiScale(gray, 1.3, 5)

for x, y, w, h in faces:

crop\_img = imgOriginal[y: y + h, x: x + w]

self.img = cv2.resize(crop\_img, (64, 64))

#self.img = image.load\_img(crop\_img, target\_size=(64, 64), grayscale=True)

capturedFace = self.img

#self.img = self.img.reshape(1, 64, 64, 3)

self.img = image.img\_to\_array(self.img)

self.img = self.img.astype("float32")

self.img = self.img / 255.0

self.img = np.expand\_dims(self.img, axis=0)

self.prediction = savedModel.predict(self.img)

classIndex = np.argmax(self.prediction, axis=-1)

#identify(?) - wa sure ug mao ba ni

if classIndex[0] == 0:

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

cv2.rectangle(imgOriginal, (x, y - 40),(x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1110089-1",(x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 1:

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

cv2.rectangle(imgOriginal, (x, y - 40),(x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1411439-2",(x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 2:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810007-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 3:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810011-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 4:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810038-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 5:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810130-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 6:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810162-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 7:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810203-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 8:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810221-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 9:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810282-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 10:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810358-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 11:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810490-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 12:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810494-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 13:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810517-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 14:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810546-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 15:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810555-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 16:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810662-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 17:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810771-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 18:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810794-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 19:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810849-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 20:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810861-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 21:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810869-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 22:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1810883-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 23:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1811231-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 24:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1811450-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 25:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1811545-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 26:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1811576-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 27:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1811584-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 28:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1820015-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 29:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1910042-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 30:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1910115-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 31:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1910142-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 32:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1910148-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 33:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1910156-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 34:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1910185-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 35:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1910252-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 36:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1910343-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 37:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1910352-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 38:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1910431-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 39:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1910470-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 40:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1910661-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 41:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1910747-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 42:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1911096-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 43:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1911114-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 44:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1911116-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 45:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1911331-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 46:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1911332-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 47:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1911353-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 48:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1911400-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 49:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1911431-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 50:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1911621-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 51:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1911747-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 52:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1911749-1-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 53:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1911753-2", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 54:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "1911944-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 55:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "2010030-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

elif classIndex[0] == 56:

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

cv2.rectangle(imgOriginal, (x, y - 40), (x + w, y), (0, 255, 0), -2)

cv2.putText(imgOriginal, "2010588-1", (x, y - 10), font, 0.75, (255, 255, 255), 1, cv2.LINE\_AA)

curId = self.mainInterface.lblID.text()

setLabels(classIndex[0], self)

#capture image from webcam(?), display at gui at webcam exit(pwede ra dili, show two guis during runtime[webcam][identity gui])

cv2.imwrite('face.png', capturedFace)

pixmap = QtGui.QPixmap('face.png')

pixmap = pixmap.scaled(self.mainInterface.picStudentPicture.height(), self.mainInterface.picStudentPicture.width(), QtCore.Qt.KeepAspectRatio)

self.mainInterface.picStudentPicture.setPixmap(pixmap)

if(curId != self.mainInterface.lblID.text):

temp = re.findall(r'\d+', self.mainInterface.lblYear.text())

extractedNum = list(map(int, temp))

now = datetime.now()

dateFormat = now.strftime("%m/%d/%Y")

timeFormat = now.strftime("%H:%M %p")

saveStudentDT(self.mainInterface.lblID.text(), self.mainInterface.lblName.text(), str(extractedNum[0]), self.mainInterface.lblCourse.text(), dateFormat, timeFormat)

cv2.imshow("Camera active for scanning", imgOriginal)

#quit by q button

if cv2.waitKey(1) & 0xFF == ord('q'):

break

connection.close()

cap.release()

cv2.destroyAllWindows()

if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

myWin = MyMainWindow()

myWin.show()

sys.exit(app.exec())

**Set Labels**

def setLabels(classNo, self):

if classNo == 0:

self.mainInterface.lblID.setText("1110089-1")

self.mainInterface.lblName.setText("Joseph")

self.mainInterface.lblCourse.setText("BIT HVACR")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 1:

self.mainInterface.lblID.setText("1411439-2")

self.mainInterface.lblName.setText("Yvette")

self.mainInterface.lblCourse.setText("BSTM")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 2:

self.mainInterface.lblID.setText("1810007-1")

self.mainInterface.lblName.setText("Neil Brian")

self.mainInterface.lblCourse.setText("BSInfoTech")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 3:

self.mainInterface.lblID.setText("1810011-2")

self.mainInterface.lblName.setText("Irish")

self.mainInterface.lblCourse.setText("BSCE")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 4:

self.mainInterface.lblID.setText("1810038-1")

self.mainInterface.lblName.setText("Alfred")

self.mainInterface.lblCourse.setText("BIT FoodPrep")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 5:

self.mainInterface.lblID.setText("1810130-1")

self.mainInterface.lblName.setText("Sherman Allen")

self.mainInterface.lblCourse.setText("BSEE")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 6:

self.mainInterface.lblID.setText("1810162-1")

self.mainInterface.lblName.setText("Kent Anderson")

self.mainInterface.lblCourse.setText("BSCrim")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 7:

self.mainInterface.lblID.setText("1810203-1")

self.mainInterface.lblName.setText("John Mellvin")

self.mainInterface.lblCourse.setText("BSCPE")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 8:

self.mainInterface.lblID.setText("1810221-1")

self.mainInterface.lblName.setText("Cliff John")

self.mainInterface.lblCourse.setText("BSEE")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 9:

self.mainInterface.lblID.setText("1810282-1")

self.mainInterface.lblName.setText("John Dave")

self.mainInterface.lblCourse.setText("BSME")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 10:

self.mainInterface.lblID.setText("1810358-1")

self.mainInterface.lblName.setText("Limuelle")

self.mainInterface.lblCourse.setText("BIT FoodPrep")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 11:

self.mainInterface.lblID.setText("1810490-1")

self.mainInterface.lblName.setText("John Michael")

self.mainInterface.lblCourse.setText("BTLED-ICT")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 12:

self.mainInterface.lblID.setText("1810494-1")

self.mainInterface.lblName.setText("Neil Kenneth")

self.mainInterface.lblCourse.setText("BIT Electrical")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 13:

self.mainInterface.lblID.setText("1810517-2")

self.mainInterface.lblName.setText("Levelyn")

self.mainInterface.lblCourse.setText("BSHM")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 14:

self.mainInterface.lblID.setText("1810546-2")

self.mainInterface.lblName.setText("Joey Mae")

self.mainInterface.lblCourse.setText("BSInfoTech")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 15:

self.mainInterface.lblID.setText("1810555-1")

self.mainInterface.lblName.setText("Jomar")

self.mainInterface.lblCourse.setText("BIT Automotive")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 16:

self.mainInterface.lblID.setText("1810662-1")

self.mainInterface.lblName.setText("Dariel")

self.mainInterface.lblCourse.setText("BIT RAC")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 17:

self.mainInterface.lblID.setText("1810771-2")

self.mainInterface.lblName.setText("Willcie Marie")

self.mainInterface.lblCourse.setText("BTLED-ICT")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 18:

self.mainInterface.lblID.setText("1810794-2")

self.mainInterface.lblName.setText("Hanna Germaine")

self.mainInterface.lblCourse.setText("BSFT")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 19:

self.mainInterface.lblID.setText("1810849-2")

self.mainInterface.lblName.setText("Eden Vina")

self.mainInterface.lblCourse.setText("BSFT")

self.mainInterface.lblYear.setText("4th year")

elif classNo == 20:

self.mainInterface.lblID.setText("1810861-2")

self.mainInterface.lblName.setText("March Tresha")

self.mainInterface.lblCourse.setText("BSFT")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 21:

self.mainInterface.lblID.setText("1810869-1")

self.mainInterface.lblName.setText("Mark Joseph")

self.mainInterface.lblCourse.setText("BIT FoodPrep")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 22:

self.mainInterface.lblID.setText("1810883-2")

self.mainInterface.lblName.setText("Richelle")

self.mainInterface.lblCourse.setText("BTLED")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 23:

self.mainInterface.lblID.setText("1811231-1")

self.mainInterface.lblName.setText("Royeth")

self.mainInterface.lblCourse.setText("BSHM")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 24:

self.mainInterface.lblID.setText("1811450-2")

self.mainInterface.lblName.setText("Charlyn")

self.mainInterface.lblCourse.setText("BSHM")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 25:

self.mainInterface.lblID.setText("1811545-2")

self.mainInterface.lblName.setText("Hera")

self.mainInterface.lblCourse.setText("BSCPE")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 26:

self.mainInterface.lblID.setText("1811576-2")

self.mainInterface.lblName.setText("April Lyn")

self.mainInterface.lblCourse.setText("BSTM")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 27:

self.mainInterface.lblID.setText("1811584-1")

self.mainInterface.lblName.setText("Jonathan")

self.mainInterface.lblCourse.setText("BIT Electrical")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 28:

self.mainInterface.lblID.setText("1819165-1")

self.mainInterface.lblName.setText("Erwin")

self.mainInterface.lblCourse.setText("BIT Electrical")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 29:

self.mainInterface.lblID.setText("1820015-1")

self.mainInterface.lblName.setText("Rommel Lyndon")

self.mainInterface.lblCourse.setText("BSCPE")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 30:

self.mainInterface.lblID.setText("1910042-1")

self.mainInterface.lblName.setText("Dave Diofrey")

self.mainInterface.lblCourse.setText("BSCrim")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 31:

self.mainInterface.lblID.setText("1910115-1")

self.mainInterface.lblName.setText("Ryan")

self.mainInterface.lblCourse.setText("BSInfoTech")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 32:

self.mainInterface.lblID.setText("1910142-2")

self.mainInterface.lblName.setText("Jean Kristel Grey")

self.mainInterface.lblCourse.setText("BSInfoTech")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 33:

self.mainInterface.lblID.setText("1910148-2")

self.mainInterface.lblName.setText("Margarit Mellen")

self.mainInterface.lblCourse.setText("BSInfoTech")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 34:

self.mainInterface.lblID.setText("1910185-2")

self.mainInterface.lblName.setText("Jondelyn Mae")

self.mainInterface.lblCourse.setText("BSInfoTech")

self.mainInterface.lblYear.setText("3rd year")

elif classNo == 35:

self.mainInterface.lblID.setText("1910252-2")

self.mainInterface.lblName.setText("Maryjoy")

self.mainInterface.lblCourse.setText("BIT FoodPrep")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 36:

self.mainInterface.lblID.setText("1910343-1")

self.mainInterface.lblName.setText("Jayson")

self.mainInterface.lblCourse.setText("BSInfoTech")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 37:

self.mainInterface.lblID.setText("1910470-2")

self.mainInterface.lblName.setText("Maden Mar")

self.mainInterface.lblCourse.setText("BSInfoTech")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 38:

self.mainInterface.lblID.setText("1910661-2")

self.mainInterface.lblName.setText("Joserose")

self.mainInterface.lblCourse.setText("BEED")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 39:

self.mainInterface.lblID.setText("1910747-2")

self.mainInterface.lblName.setText("Sharett Ann")

self.mainInterface.lblCourse.setText("BSCE")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 40:

self.mainInterface.lblID.setText("1911096-1")

self.mainInterface.lblName.setText("Ralf Jeric")

self.mainInterface.lblCourse.setText("BIT Electrical")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 41:

self.mainInterface.lblID.setText("1911114-1")

self.mainInterface.lblName.setText("Kenetnbel Jhon")

self.mainInterface.lblCourse.setText("BSME")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 42:

self.mainInterface.lblID.setText("1911116-1")

self.mainInterface.lblName.setText("Terry")

self.mainInterface.lblCourse.setText("BIT Electrical")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 43:

self.mainInterface.lblID.setText("1911331-1")

self.mainInterface.lblName.setText("Nheil John")

self.mainInterface.lblCourse.setText("BIT Automotive")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 44:

self.mainInterface.lblID.setText("1911332-1")

self.mainInterface.lblName.setText("Kianu")

self.mainInterface.lblCourse.setText("BSInfoTech")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 45:

self.mainInterface.lblID.setText("1911353-1")

self.mainInterface.lblName.setText("John Ian")

self.mainInterface.lblCourse.setText("BIT Electrical")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 46:

self.mainInterface.lblID.setText("1911400-2")

self.mainInterface.lblName.setText("Sharzyl Mae")

self.mainInterface.lblCourse.setText("BSTM")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 47:

self.mainInterface.lblID.setText("1911431-1")

self.mainInterface.lblName.setText("Nica")

self.mainInterface.lblCourse.setText("BIT FoodPrep")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 48:

self.mainInterface.lblID.setText("1911621-1")

self.mainInterface.lblName.setText("Dino")

self.mainInterface.lblCourse.setText("BIT HVACR")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 49:

self.mainInterface.lblID.setText("1911735-2")

self.mainInterface.lblName.setText("Lord Angelu")

self.mainInterface.lblCourse.setText("BSInfoTech")

self.mainInterface.lblYear.setText("Year: 4th year")

elif classNo == 50:

self.mainInterface.lblID.setText("1911747-1")

self.mainInterface.lblName.setText("Harold")

self.mainInterface.lblCourse.setText("BSInfoTech")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 51:

self.mainInterface.lblID.setText("1911749-1")

self.mainInterface.lblName.setText("Ruel")

self.mainInterface.lblCourse.setText("BSInfoTech")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 52:

self.mainInterface.lblID.setText("1911753-2")

self.mainInterface.lblName.setText("Leilane")

self.mainInterface.lblCourse.setText("BSTM")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 53:

self.mainInterface.lblID.setText("1911902-2")

self.mainInterface.lblName.setText("Christine Joyce")

self.mainInterface.lblCourse.setText("BSInfoTech")

self.mainInterface.lblYear.setText("Year: 4rd year")

elif classNo == 54:

self.mainInterface.lblID.setText("1911944-1")

self.mainInterface.lblName.setText("Jerome")

self.mainInterface.lblCourse.setText("BIT Drafting")

self.mainInterface.lblYear.setText("Year: 3rd year")

elif classNo == 55:

self.mainInterface.lblID.setText("2010030-1")

self.mainInterface.lblName.setText("Gerald")

self.mainInterface.lblCourse.setText("BSEE")

self.mainInterface.lblYear.setText("Year: 2nd year")

elif classNo == 56:

self.mainInterface.lblID.setText("2010588-1")

self.mainInterface.lblName.setText("Rovec")

self.mainInterface.lblCourse.setText("BSInfoTech")

self.mainInterface.lblYear.setText("Year: 2nd year")

**Main GUI**

from PyQt5 import QtCore, QtGui, QtWidgets

import res

class Ui\_mainWindow(object):

def setupUi(self, mainWindow):

mainWindow.setObjectName("mainWindow")

mainWindow.setFixedSize(591, 415)

self.centralwidget = QtWidgets.QWidget(mainWindow)

self.centralwidget.setObjectName("centralwidget")

self.picStudentPicture = QtWidgets.QLabel(self.centralwidget)

self.picStudentPicture.setGeometry(QtCore.QRect(20, 60, 291, 281))

self.picStudentPicture.setAutoFillBackground(True)

self.picStudentPicture.setFrameShape(QtWidgets.QFrame.StyledPanel)

self.picStudentPicture.setText("")

self.picStudentPicture.setObjectName("picStudentPicture")

self.lblID = QtWidgets.QLabel(self.centralwidget)

self.lblID.setGeometry(QtCore.QRect(90, 350, 131, 21))

font = QtGui.QFont()

font.setFamily("BankGothic")

font.setPointSize(14)

font.setBold(True)

font.setWeight(75)

self.lblID.setFont(font)

self.lblID.setObjectName("lblID")

self.lblName = QtWidgets.QLabel(self.centralwidget)

self.lblName.setGeometry(QtCore.QRect(330, 70, 221, 61))

font = QtGui.QFont()

font.setFamily("BankGothic")

font.setPointSize(14)

font.setBold(True)

font.setWeight(75)

self.lblName.setFont(font)

self.lblName.setStyleSheet("color:rgb(255, 255, 255)")

self.lblName.setTextFormat(QtCore.Qt.PlainText)

self.lblName.setScaledContents(False)

self.lblName.setWordWrap(True)

self.lblName.setObjectName("lblName")

self.btnScan = QtWidgets.QPushButton(self.centralwidget)

self.btnScan.setGeometry(QtCore.QRect(360, 350, 171, 31))

font = QtGui.QFont()

font.setFamily("BankGothic")

font.setPointSize(14)

font.setBold(True)

font.setWeight(75)

self.btnScan.setFont(font)

self.btnScan.setStyleSheet("QPushButton#btnScan{\n"

" background-color:qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, \n"

" stop:0 rgba(162, 162, 162, 255), \n"

" stop:0.55 rgba(230, 230, 230, 255), \n"

" stop:0.98 rgba(162, 162, 162, 255));\n"

" border-radius: 5px;\n"

" color: rgba(0,0,0,255);\n"

"}\n"

"QPushButton#btnScan:hover{\n"

" background-color:qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, \n"

" stop:0 rgba(162, 162, 162, 255), \n"

" stop:0.55 rgba(230, 230, 230, 255), \n"

" stop:0.98 rgba(162, 162, 162, 255));\n"

"}\n"

"QPushButton#btnScan:pressed{\n"

" padding-left: 5px;\n"

" padding-top: 5px;\n"

" background-color:rgba(105, 118, 132, 200);\n"

"}")

self.btnScan.setAutoDefault(False)

self.btnScan.setDefault(False)

self.btnScan.setFlat(False)

self.btnScan.setObjectName("btnScan")

self.lblCourse = QtWidgets.QLabel(self.centralwidget)

self.lblCourse.setGeometry(QtCore.QRect(330, 130, 221, 51))

font = QtGui.QFont()

font.setFamily("BankGothic")

font.setPointSize(14)

font.setBold(True)

font.setWeight(75)

self.lblCourse.setFont(font)

self.lblCourse.setStyleSheet("color:rgb(255, 255, 255)")

self.lblCourse.setWordWrap(True)

self.lblCourse.setObjectName("lblCourse")

self.lblYear = QtWidgets.QLabel(self.centralwidget)

self.lblYear.setGeometry(QtCore.QRect(330, 190, 221, 41))

font = QtGui.QFont()

font.setFamily("BankGothic")

font.setPointSize(14)

font.setBold(True)

font.setWeight(75)

self.lblYear.setFont(font)

self.lblYear.setStyleSheet("color:rgb(255, 255, 255)")

self.lblYear.setWordWrap(True)

self.lblYear.setObjectName("lblYear")

self.bgLabel = QtWidgets.QLabel(self.centralwidget)

self.bgLabel.setGeometry(QtCore.QRect(0, 0, 591, 401))

self.bgLabel.setStyleSheet("border-image: url(:/images/images/facebg.jpg);\n"

"border-radius:5px;")

self.bgLabel.setText("")

self.bgLabel.setObjectName("bgLabel")

self.bgLabelTransparent = QtWidgets.QLabel(self.centralwidget)

self.bgLabelTransparent.setGeometry(QtCore.QRect(326, 60, 231, 281))

self.bgLabelTransparent.setStyleSheet("background-color: rgba(0,0,0,100);\n"

"border-radius: 20px;")

self.bgLabelTransparent.setText("")

self.bgLabelTransparent.setObjectName("bgLabelTransparent")

self.logoCCSIT = QtWidgets.QLabel(self.centralwidget)

self.logoCCSIT.setGeometry(QtCore.QRect(70, 10, 41, 41))

self.logoCCSIT.setStyleSheet("border-image: url(:/images/images/CCSIT logo.png)")

self.logoCCSIT.setText("")

self.logoCCSIT.setObjectName("logoCCSIT")

self.logoSLSU = QtWidgets.QLabel(self.centralwidget)

self.logoSLSU.setGeometry(QtCore.QRect(20, 10, 41, 41))

self.logoSLSU.setStyleSheet("border-image: url(:/images/images/SLSU logo.png)")

self.logoSLSU.setText("")

self.logoSLSU.setObjectName("logoSLSU")

self.bgLabel.raise\_()

self.bgLabelTransparent.raise\_()

self.picStudentPicture.raise\_()

self.lblID.raise\_()

self.lblName.raise\_()

self.btnScan.raise\_()

self.lblCourse.raise\_()

self.lblYear.raise\_()

self.logoSLSU.raise\_()

self.logoCCSIT.raise\_()

mainWindow.setCentralWidget(self.centralwidget)

self.menubar = QtWidgets.QMenuBar(mainWindow)

self.menubar.setGeometry(QtCore.QRect(0, 0, 591, 21))

self.menubar.setObjectName("menubar")

self.menuFile = QtWidgets.QMenu(self.menubar)

self.menuFile.setObjectName("menuFile")

self.menuView = QtWidgets.QMenu(self.menuFile)

self.menuView.setObjectName("menuView")

mainWindow.setMenuBar(self.menubar)

self.actionScanned = QtWidgets.QAction(mainWindow)

self.actionScanned.setObjectName("actionScanned")

self.actionRegister = QtWidgets.QAction(mainWindow)

self.actionRegister.setObjectName("actionRegister")

self.actionViewRegistered = QtWidgets.QAction(mainWindow)

self.actionViewRegistered.setObjectName("actionViewRegistered")

self.actionViewScanned = QtWidgets.QAction(mainWindow)

self.actionViewScanned.setObjectName("actionViewScanned")

self.menuView.addAction(self.actionViewRegistered)

self.menuView.addAction(self.actionViewScanned)

self.menuFile.addAction(self.actionRegister)

self.menuFile.addAction(self.menuView.menuAction())

self.menubar.addAction(self.menuFile.menuAction())

self.retranslateUi(mainWindow)

QtCore.QMetaObject.connectSlotsByName(mainWindow)

def retranslateUi(self, mainWindow):

\_translate = QtCore.QCoreApplication.translate

mainWindow.setWindowTitle(\_translate("mainWindow", "Student Entrance Masked Face Recognition and Verification System"))

self.lblID.setText(\_translate("mainWindow", "xxxxxxx-x"))

self.lblName.setText(\_translate("mainWindow", "Name"))

self.btnScan.setText(\_translate("mainWindow", "Scan"))

self.lblCourse.setText(\_translate("mainWindow", "Course"))

self.lblYear.setText(\_translate("mainWindow", "Year"))

self.menuFile.setTitle(\_translate("mainWindow", "File"))

self.menuView.setTitle(\_translate("mainWindow", "View"))

self.actionScanned.setText(\_translate("mainWindow", "Scanned"))

self.actionRegister.setText(\_translate("mainWindow", "Register"))

self.actionViewRegistered.setText(\_translate("mainWindow", "All Records"))

self.actionViewScanned.setText(\_translate("mainWindow", "Scanned"))

if \_\_name\_\_ == "\_\_main\_\_":

import sys

app = QtWidgets.QApplication(sys.argv)

mainWindow = QtWidgets.QMainWindow()

ui = Ui\_mainWindow()

ui.setupUi(mainWindow)

mainWindow.show()

sys.exit(app.exec\_())

**Register GUI**

from PyQt5 import QtCore, QtGui, QtWidgets

from sympy import count\_roots

import res1

import pymysql

connection = pymysql.connect(host = "localhost", user = "root", password = "", database = "capstone")

cursor = connection.cursor()

class Ui\_registerForm(object):

def setupUi(self, registerForm):

registerForm.setObjectName("registerForm")

registerForm.resize(443, 261)

self.lblIDNumber = QtWidgets.QLabel(registerForm)

self.lblIDNumber.setGeometry(QtCore.QRect(50, 50, 121, 16))

font = QtGui.QFont()

font.setFamily("BankGothic")

font.setPointSize(14)

font.setBold(True)

font.setWeight(75)

self.lblIDNumber.setFont(font)

self.lblIDNumber.setObjectName("lblIDNumber")

self.txtYear = QtWidgets.QPlainTextEdit(registerForm)

self.txtYear.setGeometry(QtCore.QRect(170, 120, 211, 31))

font = QtGui.QFont()

font.setFamily("BankGothic")

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.txtYear.setFont(font)

self.txtYear.setObjectName("txtYear")

self.txtCourse = QtWidgets.QPlainTextEdit(registerForm)

self.txtCourse.setGeometry(QtCore.QRect(170, 160, 211, 31))

font = QtGui.QFont()

font.setFamily("BankGothic")

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.txtCourse.setFont(font)

self.txtCourse.setObjectName("txtCourse")

self.btnClear = QtWidgets.QPushButton(registerForm)

self.btnClear.setGeometry(QtCore.QRect(280, 210, 75, 31))

font = QtGui.QFont()

font.setFamily("BankGothic")

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.btnClear.setFont(font)

self.btnClear.setStyleSheet("QPushButton#btnClear{\n"

" background-color:qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, \n"

" stop:0 rgba(162, 162, 162, 255), \n"

" stop:0.55 rgba(230, 230, 230, 255), \n"

" stop:0.98 rgba(162, 162, 162, 255));\n"

" border-radius: 5px;\n"

" color: rgba(0,0,0,255);\n"

"}\n"

"QPushButton#btnClear:hover{\n"

" background-color:qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, \n"

" stop:0 rgba(162, 162, 162, 255), \n"

" stop:0.55 rgba(230, 230, 230, 255), \n"

" stop:0.98 rgba(162, 162, 162, 255));\n"

"}\n"

"QPushButton#btnClear:pressed{\n"

" padding-left: 5px;\n"

" padding-top: 5px;\n"

" background-color:rgba(105, 118, 132, 200);\n"

"}")

self.btnClear.setObjectName("btnClear")

self.btnSave = QtWidgets.QPushButton(registerForm)

self.btnSave.setGeometry(QtCore.QRect(190, 210, 75, 31))

font = QtGui.QFont()

font.setFamily("BankGothic")

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.btnSave.setFont(font)

self.btnSave.setStyleSheet("QPushButton#btnSave{\n"

" background-color:qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, \n"

" stop:0 rgba(162, 162, 162, 255), \n"

" stop:0.55 rgba(230, 230, 230, 255), \n"

" stop:0.98 rgba(162, 162, 162, 255));\n"

" border-radius: 5px;\n"

" color: rgba(0,0,0,255);\n"

"}\n"

"QPushButton#btnSave:hover{\n"

" background-color:qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, \n"

" stop:0 rgba(162, 162, 162, 255), \n"

" stop:0.55 rgba(230, 230, 230, 255), \n"

" stop:0.98 rgba(162, 162, 162, 255));\n"

"}\n"

"QPushButton#btnSave:pressed{\n"

" padding-left: 5px;\n"

" padding-top: 5px;\n"

" background-color:rgba(105, 118, 132, 200);\n"

"}")

self.btnSave.setObjectName("btnSave")

self.lblName = QtWidgets.QLabel(registerForm)

self.lblName.setGeometry(QtCore.QRect(50, 90, 71, 16))

font = QtGui.QFont()

font.setFamily("BankGothic")

font.setPointSize(14)

font.setBold(True)

font.setWeight(75)

self.lblName.setFont(font)

self.lblName.setObjectName("lblName")

self.lblYear = QtWidgets.QLabel(registerForm)

self.lblYear.setGeometry(QtCore.QRect(50, 130, 47, 13))

font = QtGui.QFont()

font.setFamily("BankGothic")

font.setPointSize(14)

font.setBold(True)

font.setWeight(75)

self.lblYear.setFont(font)

self.lblYear.setObjectName("lblYear")

self.lblCourse = QtWidgets.QLabel(registerForm)

self.lblCourse.setGeometry(QtCore.QRect(50, 170, 101, 16))

font = QtGui.QFont()

font.setFamily("BankGothic")

font.setPointSize(14)

font.setBold(True)

font.setWeight(75)

self.lblCourse.setFont(font)

self.lblCourse.setObjectName("lblCourse")

self.txtIDNumber = QtWidgets.QPlainTextEdit(registerForm)

self.txtIDNumber.setGeometry(QtCore.QRect(170, 40, 211, 31))

font = QtGui.QFont()

font.setFamily("BankGothic")

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.txtIDNumber.setFont(font)

self.txtIDNumber.setObjectName("txtIDNumber")

self.txtName = QtWidgets.QPlainTextEdit(registerForm)

self.txtName.setGeometry(QtCore.QRect(170, 80, 211, 31))

font = QtGui.QFont()

font.setFamily("BankGothic")

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.txtName.setFont(font)

self.txtName.setObjectName("txtName")

self.label = QtWidgets.QLabel(registerForm)

self.label.setGeometry(QtCore.QRect(0, 0, 441, 261))

self.label.setStyleSheet("border-image: url(:/img/images/hexbg.jpg);\n"

"border-radius: 5px;")

self.label.setText("")

self.label.setObjectName("label")

self.label.raise\_()

self.lblIDNumber.raise\_()

self.txtYear.raise\_()

self.txtCourse.raise\_()

self.btnClear.raise\_()

self.btnSave.raise\_()

self.lblName.raise\_()

self.lblYear.raise\_()

self.lblCourse.raise\_()

self.txtIDNumber.raise\_()

self.txtName.raise\_()

self.btnClear.clicked.connect(self.clearInput)

self.btnSave.clicked.connect(self.saveStudent)

self.retranslateUi(registerForm)

QtCore.QMetaObject.connectSlotsByName(registerForm)

def clearInput(self):

self.txtIDNumber.setPlainText("")

self.txtName.setPlainText("")

self.txtCourse.setPlainText("")

self.txtYear.setPlainText("")

def saveStudent(self):

idNumber = self.txtIDNumber.toPlainText()

name = self.txtName.toPlainText()

year = self.txtYear.toPlainText()

course = self.txtCourse.toPlainText()

insertSQL = "INSERT INTO faces(ID, NAME, YEAR, COURSE, VERIFIED) VALUES (%s, %s, %s, %s, %s);"

val = (idNumber, name, year, course, '0')

cursor.execute(insertSQL, val)

connection.commit()

connection.close()

print(idNumber)

print(name)

print(year)

print(course)

def retranslateUi(self, registerForm):

\_translate = QtCore.QCoreApplication.translate

registerForm.setWindowTitle(\_translate("registerForm", "Register Form"))

self.lblIDNumber.setText(\_translate("registerForm", "ID Number"))

self.btnClear.setText(\_translate("registerForm", "Clear"))

self.btnSave.setText(\_translate("registerForm", "Save"))

self.lblName.setText(\_translate("registerForm", "Name"))

self.lblYear.setText(\_translate("registerForm", "Year"))

self.lblCourse.setText(\_translate("registerForm", "Course"))

if \_\_name\_\_ == "\_\_main\_\_":

import sys

app = QtWidgets.QApplication(sys.argv)

registerForm = QtWidgets.QDialog()

ui = Ui\_registerForm()

ui.setupUi(registerForm)

registerForm.show()

sys.exit(app.exec\_())

**View Registered Student GUI**

from PyQt5 import QtCore, QtGui, QtWidgets

import pymysql

import ast

connection = pymysql.connect(host= "localhost", user= "root", password= "", database= "capstone")

cursor = connection.cursor()

def convert(mydata):

def cvt(data):

try:

return ast.literal\_eval(data)

except Exception:

return str(data)

return tuple(map(cvt, mydata))

class Ui\_viewRecords(object):

def loadData(self):

with connection:

rows = cursor.execute("SELECT \* FROM faces")

data = cursor.fetchall()

for row in data:

self.addTable(convert(row))

cursor.close()

def addTable(self, columns):

rowPosition = self.tblData.rowCount()

self.tblData.insertRow(rowPosition)

for i, column in enumerate(columns):

self.tblData.setItem(rowPosition, i, QtWidgets.QTableWidgetItem(str(column)))

def setupUi(self, viewRecords):

viewRecords.setObjectName("viewRecords")

viewRecords.resize(579, 432)

self.tblData = QtWidgets.QTableWidget(viewRecords)

self.tblData.setGeometry(QtCore.QRect(10, 10, 561, 411))

self.tblData.setFrameShape(QtWidgets.QFrame.StyledPanel)

self.tblData.setFrameShadow(QtWidgets.QFrame.Sunken)

self.tblData.setRowCount(0)

self.tblData.setColumnCount(5)

self.tblData.setObjectName("tblData")

item = QtWidgets.QTableWidgetItem()

self.tblData.setHorizontalHeaderItem(0, item)

item = QtWidgets.QTableWidgetItem()

self.tblData.setHorizontalHeaderItem(1, item)

item = QtWidgets.QTableWidgetItem()

self.tblData.setHorizontalHeaderItem(2, item)

item = QtWidgets.QTableWidgetItem()

self.tblData.setHorizontalHeaderItem(3, item)

item = QtWidgets.QTableWidgetItem()

self.tblData.setHorizontalHeaderItem(4, item)

self.loadData()

self.tblData.setColumnWidth(0, 70)

self.tblData.setColumnWidth(1, 200)

self.tblData.setColumnWidth(2, 50)

self.tblData.setColumnWidth(3, 150)

self.tblData.setColumnWidth(4, 50)

self.retranslateUi(viewRecords)

QtCore.QMetaObject.connectSlotsByName(viewRecords)

def retranslateUi(self, viewRecords):

\_translate = QtCore.QCoreApplication.translate

viewRecords.setWindowTitle(\_translate("viewRecords", "Student Records"))

item = self.tblData.horizontalHeaderItem(0)

item.setText(\_translate("viewRecords", "ID"))

item = self.tblData.horizontalHeaderItem(1)

item.setText(\_translate("viewRecords", "Name"))

item = self.tblData.horizontalHeaderItem(2)

item.setText(\_translate("viewRecords", "Year"))

item = self.tblData.horizontalHeaderItem(3)

item.setText(\_translate("viewRecords", "Course"))

item = self.tblData.horizontalHeaderItem(4)

item.setText(\_translate("viewRecords", "Verified"))

if \_\_name\_\_ == "\_\_main\_\_":

import sys

app = QtWidgets.QApplication(sys.argv)

viewRecords = QtWidgets.QDialog()

ui = Ui\_viewRecords()

ui.setupUi(viewRecords)

viewRecords.show()

sys.exit(app.exec\_())

**View Scanned Students GUI**

from PyQt5 import QtCore, QtGui, QtWidgets

import pymysql

import ast

connection = pymysql.connect(host= "localhost", user= "root", password= "", database= "capstone")

cursor = connection.cursor()

def convert(mydata):

def cvt(data):

try:

return ast.literal\_eval(data)

except Exception:

return str(data)

return tuple(map(cvt, mydata))

class Ui\_viewScanned(object):

def loadData(self):

with connection:

rows = cursor.execute("SELECT \* FROM scanned")

data = cursor.fetchall()

for row in data:

self.addTable(convert(row))

cursor.close()

def addTable(self, columns):

rowPosition = self.tblData.rowCount()

self.tblData.insertRow(rowPosition)

for i, column in enumerate(columns):

self.tblData.setItem(rowPosition, i, QtWidgets.QTableWidgetItem(str(column)))

def setupUi(self, viewScanned):

viewScanned.setObjectName("viewScanned")

viewScanned.resize(541, 385)

self.tblData = QtWidgets.QTableWidget(viewScanned)

self.tblData.setGeometry(QtCore.QRect(10, 10, 521, 361))

self.tblData.setFrameShape(QtWidgets.QFrame.StyledPanel)

self.tblData.setFrameShadow(QtWidgets.QFrame.Sunken)

self.tblData.setRowCount(0)

self.tblData.setColumnCount(6)

self.tblData.setObjectName("tblData")

item = QtWidgets.QTableWidgetItem()

self.tblData.setHorizontalHeaderItem(0, item)

item = QtWidgets.QTableWidgetItem()

self.tblData.setHorizontalHeaderItem(1, item)

item = QtWidgets.QTableWidgetItem()

self.tblData.setHorizontalHeaderItem(2, item)

item = QtWidgets.QTableWidgetItem()

self.tblData.setHorizontalHeaderItem(3, item)

item = QtWidgets.QTableWidgetItem()

self.tblData.setHorizontalHeaderItem(4, item)

item = QtWidgets.QTableWidgetItem()

self.tblData.setHorizontalHeaderItem(5, item)

self.loadData()

self.tblData.setColumnWidth(0, 70)

self.tblData.setColumnWidth(1, 134)

self.tblData.setColumnWidth(2, 50)

self.tblData.setColumnWidth(3, 85)

self.tblData.setColumnWidth(4, 70)

self.tblData.setColumnWidth(5, 70)

self.retranslateUi(viewScanned)

QtCore.QMetaObject.connectSlotsByName(viewScanned)

def retranslateUi(self, viewScanned):

\_translate = QtCore.QCoreApplication.translate

viewScanned.setWindowTitle(\_translate("viewScanned", "Scan List"))

item = self.tblData.horizontalHeaderItem(0)

item.setText(\_translate("viewScanned", "ID"))

item = self.tblData.horizontalHeaderItem(1)

item.setText(\_translate("viewScanned", "Name"))

item = self.tblData.horizontalHeaderItem(2)

item.setText(\_translate("viewScanned", "Year"))

item = self.tblData.horizontalHeaderItem(3)

item.setText(\_translate("viewScanned", "Course"))

item = self.tblData.horizontalHeaderItem(4)

item.setText(\_translate("viewScanned", "Date"))

item = self.tblData.horizontalHeaderItem(5)

item.setText(\_translate("viewScanned", "Time"))

if \_\_name\_\_ == "\_\_main\_\_":

import sys

app = QtWidgets.QApplication(sys.argv)

viewScanned = QtWidgets.QDialog()

ui = Ui\_viewScanned()

ui.setupUi(viewScanned)

viewScanned.show()

sys.exit(app.exec\_())

**Appendix B**

**EVALUATION TOOL**

**System Evaluation (ISO 9126)**

**Instructions:** Please evaluate the “Student Entrance Masked Face Recognition and Verification System” using the scale shown below.

Check(/) the appropriate score. Thank You.

Alphonse Victor N. Antegra Jannie Fleur V. Oraño

Programmer Adviser

**Qualitative Description per Functionality Indicator**

|  |  |
| --- | --- |
| **Limits of Scale** | **Qualitative Description** |
| 4.21 – 5.00 | Fully Functional |
| 3.21 – 4.20 | Mostly Functional |
| 2.61 – 3.20 | Functional |
| 1.81 – 2.60 | Slightly Functional |
| 1.0 – 1.8 | Not Functional |

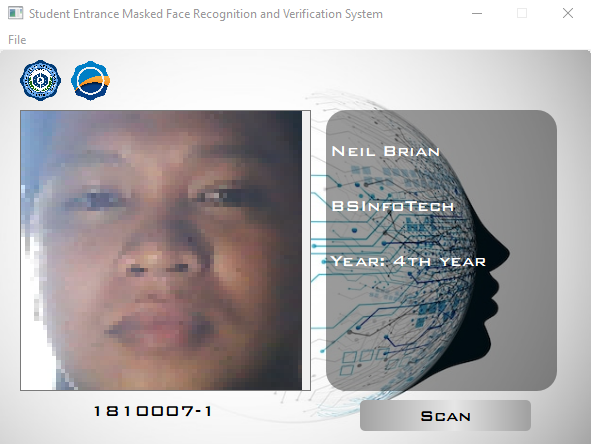
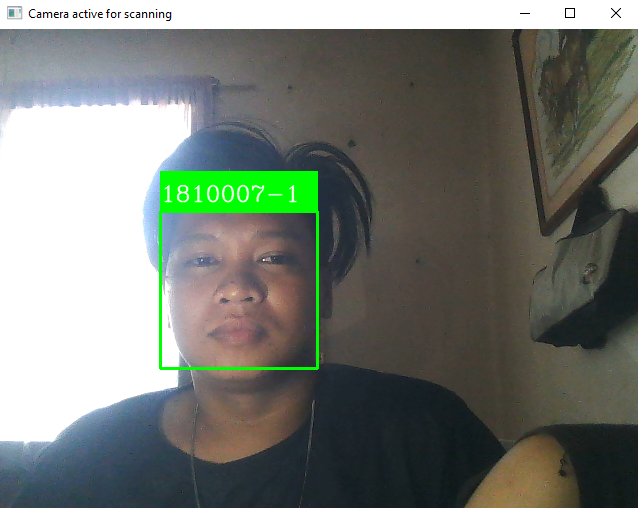
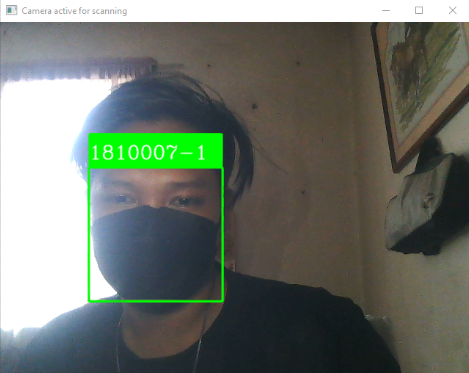
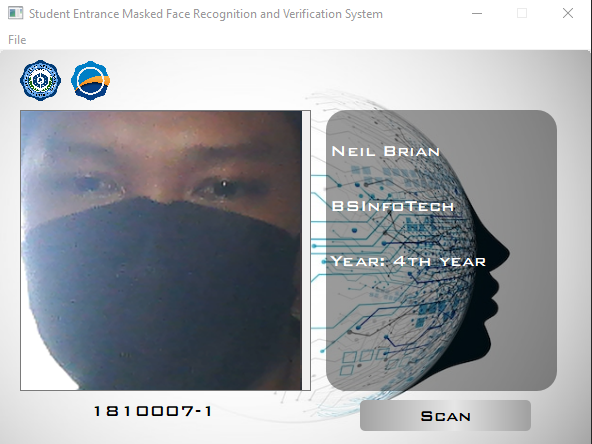
**Qualitative Description per Usability Indicator**

|  |  |
| --- | --- |
| **Limits of Scale** | **Qualitative Description** |
| 4.21 – 5.00 | Fully Usable |
| 3.21 – 4.20 | Mostly Usable |
| 2.61 – 3.20 | Usable |
| 1.81 – 2.60 | Slightly Usable |
| 1.0 – 1.8 | Not Usable |

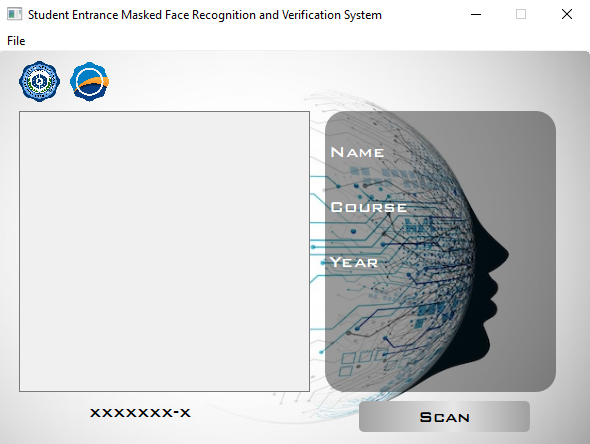
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Criteria** | | **Score** | | | | |
| *Characteristics* | *Sub-Characteristics* | **1** | **2** | **3** | **4** | **5** |
| Functionality | The application performs the required functionalities |  |  |  |  |  |
| The application provides the expected result |  |  |  |  |  |
| Usability | The graphical user interface of the application is easy to use or navigate |  |  |  |  |  |
| The displayed results of the application are understandable |  |  |  |  |  |
| The application is portable |  |  |  |  |  |
| Based on the utilized online test dataset, the application can recognize the students |  |  |  |  |  |
| The application is helpful |  |  |  |  |  |

**Appendix C**

**SAMPLE INPUT AND OUTPUT**

****

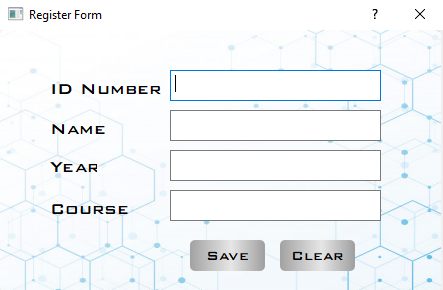
**Appendix C**

**USER’S GUIDE**

**To scan students**

**1.** Click the scan button.

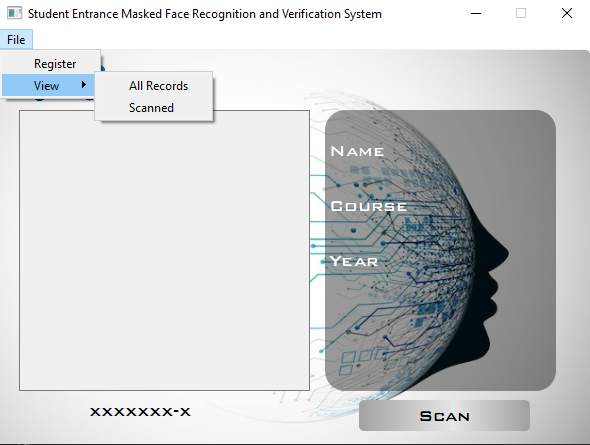
2. Let the user face the camera.

****

**To register student**

**1.** Click File.

2. Click Register.



**To view All records/Scanned students**

**1.** Click File.

**2.** Click view.

**3.** Click All records / scanned.

**Appendix E**

**OTHER RELEVANT DOCUMENTS**

**Testing Result** (test.csv)

|  |  |
| --- | --- |
| Filename | Predictions |
| 1110089-1\IMG\_20211029\_103147.jpg | 1110089-1 |
| 1110089-1\IMG\_20211029\_103150.jpg | 1110089-1 |
| 1110089-1\IMG\_20211029\_103421.jpg | 1110089-1 |
| 1110089-1\IMG\_20211029\_103424.jpg | 1110089-1 |
| 1110089-1\IMG\_20211029\_103427.jpg | 1110089-1 |
| 1411439-2\IMG\_20211029\_141139.jpg | 1411439-2 |
| 1411439-2\IMG\_20211029\_141146.jpg | 1411439-2 |
| 1411439-2\IMG\_20211029\_141151.jpg | 1411439-2 |
| 1411439-2\IMG\_20211029\_141308.jpg | 1411439-2 |
| 1411439-2\IMG\_20211029\_141310.jpg | 1411439-2 |
| 1810007-1\IMG\_20211108\_154201.jpg | 1810007-1 |
| 1810007-1\IMG\_20211108\_154214.jpg | 1810007-1 |
| 1810007-1\IMG\_20211108\_154258.jpg | 1810007-1 |
| 1810007-1\IMG\_20211108\_154458.jpg | 1810007-1 |
| 1810007-1\IMG\_20211108\_154501.jpg | 1810007-1 |
| 1810007-1\IMG\_20211108\_154813.jpg | 1810007-1 |
| 1810007-1\IMG\_20211108\_154816.jpg | 1810007-1 |
| 1810007-1\IMG\_20211108\_154831.jpg | 1810007-1 |
| 1810007-1\IMG\_20211108\_154834.jpg | 1810007-1 |
| 1810007-1\IMG\_20211108\_154836.jpg | 1810007-1 |
| 1810011-2\IMG\_20211108\_103341.jpg | 1810011-2 |
| 1810011-2\IMG\_20211108\_103401.jpg | 1810011-2 |
| 1810011-2\IMG\_20211108\_103511.jpg | 1810011-2 |
| 1810011-2\IMG\_20211108\_103517.jpg | 1810011-2 |
| 1810011-2\IMG\_20211108\_103524.jpg | 1810011-2 |
| 1810038-1\received\_1065260960902945.jpeg | 1810038-1 |
| 1810038-1\received\_401269321703351.jpeg | 1810038-1 |
| 1810038-1\received\_405549261075765.jpeg | 1810038-1 |
| 1810038-1\received\_438026647663307.jpeg | 1810038-1 |
| 1810038-1\received\_572146727349759.jpeg | 1810038-1 |
| 1810130-1\IMG\_20211029\_152902.jpg | 1810130-1 |
| 1810130-1\IMG\_20211029\_152906.jpg | 1810130-1 |
| 1810130-1\IMG\_20211029\_152920.jpg | 1810130-1 |
| 1810130-1\IMG\_20211029\_153057.jpg | 1810130-1 |
| 1810130-1\IMG\_20211029\_153155.jpg | 1810130-1 |
| 1810162-1\IMG\_20211108\_103748.jpg | 1810162-1 |
| 1810162-1\IMG\_20211108\_103800.jpg | 1810162-1 |
| 1810162-1\IMG\_20211108\_103819.jpg | 1810162-1 |
| 1810162-1\IMG\_20211108\_104055.jpg | 1810162-1 |
| 1810162-1\IMG\_20211108\_104123.jpg | 1810162-1 |
| 1810203-1\IMG\_20211029\_105251.jpg | 1810203-1 |
| 1810203-1\IMG\_20211029\_105254.jpg | 1810203-1 |
| 1810203-1\IMG\_20211029\_105257.jpg | 1810203-1 |
| 1810203-1\IMG\_20211029\_105510.jpg | 1810203-1 |
| 1810203-1\IMG\_20211029\_105513.jpg | 1810203-1 |
| 1810221-1\IMG\_20211108\_084824.jpg | 1810221-1 |
| 1810221-1\IMG\_20211108\_084829.jpg | 1810221-1 |
| 1810221-1\IMG\_20211108\_084845.jpg | 1810221-1 |
| 1810221-1\IMG\_20211108\_085210.jpg | 1810221-1 |
| 1810221-1\IMG\_20211108\_085222.jpg | 1810221-1 |
| 1810282-1\IMG\_20211029\_090838.jpg | 1810282-1 |
| 1810282-1\IMG\_20211029\_090903.jpg | 1810282-1 |
| 1810282-1\IMG\_20211029\_090921.jpg | 1810282-1 |
| 1810282-1\IMG\_20211029\_091014.jpg | 1810282-1 |
| 1810282-1\IMG\_20211029\_091021.jpg | 1810282-1 |
| 1810358-1\IMG\_20211108\_095012.jpg | 1810358-1 |
| 1810358-1\IMG\_20211108\_095019.jpg | 1810358-1 |
| 1810358-1\IMG\_20211108\_095747.jpg | 1810358-1 |
| 1810358-1\IMG\_20211108\_095833.jpg | 1810358-1 |
| 1810358-1\IMG\_20211108\_095841.jpg | 1810358-1 |
| 1810490-1\250208853\_216363403951243\_8477346511775780526\_n.jpg | 1810490-1 |
| 1810490-1\250749207\_460347645508801\_2518776721339003517\_n.jpg | 1810490-1 |
| 1810490-1\251365982\_354813513111844\_6622617993126039727\_n.jpg | 1810490-1 |
| 1810490-1\252042030\_372767734633059\_7471281436438439541\_n.jpg | 1810490-1 |
| 1810490-1\252958272\_215705127303509\_3187968455893334036\_n.jpg | 1810490-1 |
| 1810494-1\IMG\_20211029\_100703.jpg | 1810494-1 |
| 1810494-1\IMG\_20211029\_100813.jpg | 1810494-1 |
| 1810494-1\IMG\_20211029\_100818.jpg | 1810494-1 |
| 1810494-1\IMG\_20211029\_100941.jpg | 1810494-1 |
| 1810494-1\IMG\_20211029\_100945.jpg | 1810494-1 |
| 1810517-2\IMG\_20211029\_092358.jpg | 1810517-2 |
| 1810517-2\IMG\_20211029\_092402.jpg | 1810517-2 |
| 1810517-2\IMG\_20211029\_092837.jpg | 1810517-2 |
| 1810517-2\IMG\_20211029\_092841.jpg | 1810517-2 |
| 1810517-2\IMG\_20211029\_092846.jpg | 1810517-2 |
| 1810546-2\IMG\_20211107\_095320.jpg | 1810546-2 |
| 1810546-2\IMG\_20211107\_095324.jpg | 1810546-2 |
| 1810546-2\IMG\_20211107\_095514.jpg | 1810546-2 |
| 1810546-2\IMG\_20211107\_095521.jpg | 1810546-2 |
| 1810546-2\IMG\_20211107\_095533.jpg | 1810546-2 |
| 1810555-1\249675711\_685226569123621\_3456206877896688202\_n (1).jpg | 1810555-1 |
| 1810555-1\250518628\_613996886448933\_875690580748590651\_n.jpg | 1810555-1 |
| 1810555-1\250780010\_6699167776822284\_4131237571432379355\_n.jpg | 1810555-1 |
| 1810555-1\250803570\_404874197881723\_8270945493087003837\_n.jpg | 1810555-1 |
| 1810555-1\250814823\_254125800071919\_2231646221638786368\_n.jpg | 1911116-1 |
| 1810662-1\IMG\_20211105\_091900.jpg | 1810662-1 |
| 1810662-1\IMG\_20211105\_091910.jpg | 1810662-1 |
| 1810662-1\IMG\_20211105\_091918.jpg | 1810662-1 |
| 1810662-1\IMG\_20211105\_092244.jpg | 1810662-1 |
| 1810662-1\IMG\_20211105\_092345.jpg | 1810662-1 |
| 1810771-2\254276487\_313247026983307\_1867958408296381539\_n.jpg | 1810771-2 |
| 1810771-2\254276487\_623607542339390\_1410814822335508905\_n.jpg | 1810771-2 |
| 1810771-2\254293562\_414864290117523\_7873364847636614006\_n.jpg | 1810771-2 |
| 1810771-2\254350209\_240222638096494\_4659918053162615221\_n.jpg | 1810771-2 |
| 1810771-2\254724737\_4460918850688384\_6191331359452849426\_n.jpg | 1810771-2 |
| 1810794-2\received\_303376094701215.jpeg | 1810794-2 |
| 1810794-2\received\_307770874225966.jpeg | 1810794-2 |
| 1810794-2\received\_581575683062803.jpeg | 1810794-2 |
| 1810849-2\IMG\_20211107\_085154.jpg | 1810849-2 |
| 1810849-2\IMG\_20211107\_085249.jpg | 1810849-2 |
| 1810849-2\IMG\_20211107\_085252.jpg | 1810849-2 |
| 1810849-2\IMG\_20211107\_085619.jpg | 1810849-2 |
| 1810849-2\IMG\_20211107\_085621.jpg | 1810849-2 |
| 1810861-2\IMG\_20211108\_153218.jpg | 1810861-2 |
| 1810861-2\IMG\_20211108\_153246.jpg | 1810861-2 |
| 1810861-2\IMG\_20211108\_153318.jpg | 1810861-2 |
| 1810861-2\IMG\_20211108\_153754.jpg | 1810861-2 |
| 1810861-2\IMG\_20211108\_153817.jpg | 1810861-2 |
| 1810869-1\IMG\_20211029\_102426.jpg | 1810869-1 |
| 1810869-1\IMG\_20211029\_102429.jpg | 1810869-1 |
| 1810869-1\IMG\_20211029\_102534.jpg | 1810869-1 |
| 1810869-1\IMG\_20211029\_102537.jpg | 1810869-1 |
| 1810869-1\IMG\_20211029\_102540.jpg | 1810869-1 |
| 1810883-2\248716721\_439590727739718\_3403265657743283867\_n.jpg | 1810883-2 |
| 1810883-2\249849465\_4996524807038310\_731514916704846154\_n.jpg | 1810883-2 |
| 1810883-2\249858710\_3516794801880082\_4058137133806587156\_n.jpg | 1810883-2 |
| 1810883-2\250279548\_896853374305548\_2878404648635794253\_n.jpg | 1810883-2 |
| 1810883-2\250690970\_1784824638380102\_1944672583568270758\_n.jpg | 1810883-2 |
| 1811231-1\IMG\_20211108\_145146.jpg | 1811231-1 |
| 1811231-1\IMG\_20211108\_145201.jpg | 1811231-1 |
| 1811231-1\IMG\_20211108\_145312.jpg | 1811231-1 |
| 1811231-1\IMG\_20211108\_145558.jpg | 1811231-1 |
| 1811231-1\IMG\_20211108\_145710.jpg | 1811231-1 |
| 1811450-2\IMG\_20211029\_135122.jpg | 1811450-2 |
| 1811450-2\IMG\_20211029\_135125.jpg | 1811450-2 |
| 1811450-2\IMG\_20211029\_135245.jpg | 1811450-2 |
| 1811450-2\IMG\_20211029\_135248.jpg | 1811450-2 |
| 1811450-2\IMG\_20211029\_135334.jpg | 1811450-2 |
| 1811545-2\IMG\_20211107\_103929.jpg | 1811545-2 |
| 1811545-2\IMG\_20211107\_104026.jpg | 1811545-2 |
| 1811545-2\IMG\_20211107\_104038.jpg | 1811545-2 |
| 1811545-2\IMG\_20211107\_104245.jpg | 1811545-2 |
| 1811545-2\IMG\_20211107\_104321.jpg | 1811545-2 |
| 1811576-2\IMG\_20211029\_133726.jpg | 1811576-2 |
| 1811576-2\IMG\_20211029\_133736.jpg | 1811576-2 |
| 1811576-2\IMG\_20211029\_133829.jpg | 1811576-2 |
| 1811576-2\IMG\_20211029\_133921.jpg | 1811576-2 |
| 1811576-2\IMG\_20211029\_133933.jpg | 1811576-2 |
| 1811584-1\IMG\_20211108\_102531.jpg | 1811584-1 |
| 1811584-1\IMG\_20211108\_102544.jpg | 1811584-1 |
| 1811584-1\IMG\_20211108\_102731.jpg | 1811584-1 |
| 1811584-1\IMG\_20211108\_102746.jpg | 1811584-1 |
| 1811584-1\IMG\_20211108\_102753.jpg | 1811584-1 |
| 1819165-1\248392465\_930784084510187\_4431370517887633707\_n.jpg | 1819165-1 |
| 1819165-1\248405823\_614242533111561\_6670196566752914642\_n.jpg | 1819165-1 |
| 1819165-1\248667782\_952944648946009\_6006779850568611799\_n.jpg | 1819165-1 |
| 1819165-1\249011086\_410380980580354\_2932983832412505676\_n.jpg | 1819165-1 |
| 1819165-1\249385473\_1243822162751009\_4429510048448601557\_n.jpg | 1819165-1 |
| 1820015-1\IMG\_20211105\_143116.jpg | 1820015-1 |
| 1820015-1\IMG\_20211105\_143126.jpg | 1820015-1 |
| 1820015-1\IMG\_20211105\_143158.jpg | 1820015-1 |
| 1820015-1\IMG\_20211105\_143606.jpg | 1820015-1 |
| 1820015-1\IMG\_20211105\_143618.jpg | 1820015-1 |
| 1820015-1\IMG\_20211105\_143739.jpg | 1820015-1 |
| 1910034-1\254093355\_473799950632347\_7869062380672016509\_n.jpg | 1910034-1 |
| 1910034-1\254349319\_1109205246282237\_5938263215014642126\_n.jpg | 1910034-1 |
| 1910034-1\254364017\_840120609988546\_7525690121912992557\_n.jpg | 1910034-1 |
| 1910034-1\254422715\_2767111680253583\_6746964124695904725\_n.jpg | 1910034-1 |
| 1910034-1\254484182\_628109758216733\_5726641290315179309\_n.jpg | 1910034-1 |
| 1910042-1\IMG\_20211029\_105856.jpg | 1910042-1 |
| 1910042-1\IMG\_20211029\_105858.jpg | 1910042-1 |
| 1910042-1\IMG\_20211029\_110008.jpg | 1910042-1 |
| 1910042-1\IMG\_20211029\_110012.jpg | 1910042-1 |
| 1910042-1\IMG\_20211029\_110016.jpg | 1910042-1 |
| 1910115-1\20211105\_173927.jpg | 1910115-1 |
| 1910115-1\20211105\_173936.jpg | 1910115-1 |
| 1910115-1\20211105\_173941.jpg | 1910115-1 |
| 1910115-1\20211105\_174118.jpg | 1910115-1 |
| 1910115-1\20211105\_174125.jpg | 1910115-1 |
| 1910142-2\20211105\_172252.jpg | 1910142-2 |
| 1910142-2\20211105\_172308.jpg | 1910142-2 |
| 1910142-2\20211105\_172321.jpg | 1910142-2 |
| 1910142-2\20211105\_172414.jpg | 1910142-2 |
| 1910142-2\20211105\_172421.jpg | 1910142-2 |
| 1910142-2\20211105\_172430.jpg | 1910142-2 |
| 1910148-2\20211105\_172655.jpg | 1910148-2 |
| 1910148-2\20211105\_172701.jpg | 1910148-2 |
| 1910148-2\20211105\_172707.jpg | 1910148-2 |
| 1910148-2\20211105\_172936.jpg | 1910148-2 |
| 1910148-2\20211105\_172940.jpg | 1910148-2 |
| 1910185-2\20211105\_173453.jpg | 1910185-2 |
| 1910185-2\20211105\_173510.jpg | 1910185-2 |
| 1910185-2\20211105\_173516.jpg | 1910185-2 |
| 1910185-2\20211105\_173710.jpg | 1910185-2 |
| 1910185-2\20211105\_173714.jpg | 1910185-2 |
| 1910252-2\IMG\_20211029\_094221.jpg | 1910252-2 |
| 1910252-2\IMG\_20211029\_094254.jpg | 1910252-2 |
| 1910252-2\IMG\_20211029\_094409.jpg | 1910252-2 |
| 1910252-2\IMG\_20211029\_094419.jpg | 1910252-2 |
| 1910252-2\IMG\_20211029\_094441.jpg | 1910252-2 |
| 1910343-1\20211105\_174303.jpg | 1910343-1 |
| 1910343-1\20211105\_174309.jpg | 1910343-1 |
| 1910343-1\20211105\_174328.jpg | 1910343-1 |
| 1910343-1\20211105\_174421.jpg | 1910343-1 |
| 1910343-1\20211105\_174533.jpg | 1910343-1 |
| 1910661-2\IMG\_20211107\_100417.jpg | 1910661-2 |
| 1910661-2\IMG\_20211107\_100426.jpg | 1910661-2 |
| 1910661-2\IMG\_20211107\_100433.jpg | 1910661-2 |
| 1910661-2\IMG\_20211107\_100606.jpg | 1910661-2 |
| 1910661-2\IMG\_20211107\_100613.jpg | 1910661-2 |
| 1910747-2\IMG\_20211108\_100846.jpg | 1910747-2 |
| 1910747-2\IMG\_20211108\_100904.jpg | 1910747-2 |
| 1910747-2\IMG\_20211108\_101130.jpg | 1910747-2 |
| 1910747-2\IMG\_20211108\_101136.jpg | 1910747-2 |
| 1910747-2\IMG\_20211108\_101143.jpg | 1910747-2 |
| 1911096-1\IMG\_20211107\_162737.jpg | 1911096-1 |
| 1911096-1\IMG\_20211107\_162803.jpg | 1911096-1 |
| 1911096-1\IMG\_20211107\_162855.jpg | 1911096-1 |
| 1911096-1\IMG\_20211107\_162902.jpg | 1911096-1 |
| 1911096-1\IMG\_20211107\_162905.jpg | 1911096-1 |
| 1911114-1\IMG\_20211108\_164936.jpg | 1911114-1 |
| 1911114-1\IMG\_20211108\_165005.jpg | 1911114-1 |
| 1911114-1\IMG\_20211108\_165041.jpg | 1911114-1 |
| 1911114-1\IMG\_20211108\_165226.jpg | 1911114-1 |
| 1911114-1\IMG\_20211108\_165311.jpg | 1911114-1 |
| 1911116-1\IMG\_20211107\_163622.jpg | 1911116-1 |
| 1911116-1\IMG\_20211107\_163632.jpg | 1911116-1 |
| 1911116-1\IMG\_20211107\_163636.jpg | 1911116-1 |
| 1911116-1\IMG\_20211107\_163945.jpg | 1911116-1 |
| 1911116-1\IMG\_20211107\_164005.jpg | 1911116-1 |
| 1911331-1\IMG\_20211105\_094941.jpg | 1911331-1 |
| 1911331-1\IMG\_20211105\_094949.jpg | 1911331-1 |
| 1911331-1\IMG\_20211105\_094952.jpg | 1911331-1 |
| 1911331-1\IMG\_20211105\_095138.jpg | 1911331-1 |
| 1911331-1\IMG\_20211105\_095144.jpg | 1911331-1 |
| 1911331-1\IMG\_20211105\_095232.jpg | 1911331-1 |
| 1911332-1\IMG\_20211105\_105929.jpg | 1911332-1 |
| 1911332-1\IMG\_20211105\_105944.jpg | 1911332-1 |
| 1911332-1\IMG\_20211105\_110059.jpg | 1911332-1 |
| 1911332-1\IMG\_20211105\_110119.jpg | 1911332-1 |
| 1911332-1\IMG\_20211105\_110124.jpg | 1911332-1 |
| 1911332-1\IMG\_20211105\_110155.jpg | 1911332-1 |
| 1911353-1\IMG\_20211029\_101538.jpg | 1911353-1 |
| 1911353-1\IMG\_20211029\_101540.jpg | 1911353-1 |
| 1911353-1\IMG\_20211029\_101653.jpg | 1911353-1 |
| 1911353-1\IMG\_20211029\_101704.jpg | 1911353-1 |
| 1911353-1\IMG\_20211029\_101707.jpg | 1911353-1 |
| 1911400-2\IMG\_20211029\_142857.jpg | 1911400-2 |
| 1911400-2\IMG\_20211029\_142906.jpg | 1911400-2 |
| 1911400-2\IMG\_20211029\_142944.jpg | 1911400-2 |
| 1911400-2\IMG\_20211029\_143109.jpg | 1911400-2 |
| 1911400-2\IMG\_20211029\_143129.jpg | 1911400-2 |
| 1911431-1\IMG\_20211107\_155225.jpg | 1911431-1 |
| 1911431-1\IMG\_20211107\_155230.jpg | 1911431-1 |
| 1911431-1\IMG\_20211107\_155402.jpg | 1911431-1 |
| 1911431-1\IMG\_20211107\_155409.jpg | 1911431-1 |
| 1911431-1\IMG\_20211107\_155419.jpg | 1911431-1 |
| 1911621-1\IMG\_20211029\_102813.jpg | 1911621-1 |
| 1911621-1\IMG\_20211029\_102845.jpg | 1911621-1 |
| 1911621-1\IMG\_20211029\_102901.jpg | 1911621-1 |
| 1911621-1\IMG\_20211029\_103003.jpg | 1911621-1 |
| 1911621-1\IMG\_20211029\_103006.jpg | 1911621-1 |
| 1911735-2\received\_179429894373133.jpeg | 1911735-2 |
| 1911735-2\received\_203229491919022.jpeg | 1911735-2 |
| 1911735-2\received\_228718002607085.jpeg | 1911735-2 |
| 1911735-2\received\_426299415807576.jpeg | 1911735-2 |
| 1911735-2\received\_792740431515896.jpeg | 1911735-2 |
| 1911747-1\IMG\_20211108\_101805.jpg | 1911747-1 |
| 1911747-1\IMG\_20211108\_101813.jpg | 1911747-1 |
| 1911747-1\IMG\_20211108\_101817.jpg | 1911747-1 |
| 1911747-1\IMG\_20211108\_102128.jpg | 1911747-1 |
| 1911747-1\IMG\_20211108\_102136.jpg | 1911747-1 |
| 1911749-1\IMG\_20211107\_155857.jpg | 1911749-1 |
| 1911749-1\IMG\_20211107\_155903.jpg | 1911749-1 |
| 1911749-1\IMG\_20211107\_160054.jpg | 1911749-1 |
| 1911749-1\IMG\_20211107\_160102.jpg | 1911749-1 |
| 1911749-1\IMG\_20211107\_160114.jpg | 1911749-1 |
| 1911753-2\IMG\_20211029\_142337.jpg | 1911753-2 |
| 1911753-2\IMG\_20211029\_142342.jpg | 1911753-2 |
| 1911753-2\IMG\_20211029\_142413.jpg | 1911753-2 |
| 1911753-2\IMG\_20211029\_142613.jpg | 1911753-2 |
| 1911753-2\IMG\_20211029\_142628.jpg | 1911753-2 |
| 1911753-2\IMG\_20211029\_142637.jpg | 1911753-2 |
| 1911902-2\IMG20210128134129.jpg | 1810221-1 |
| 1911902-2\IMG20210620183621.jpg | 1911902-2 |
| 1911902-2\IMG20211031103645.jpg | 1911902-2 |
| 1911902-2\IMG20211031110123.jpg | 1911902-2 |
| 1911902-2\IMG20211031110130.jpg | 1911902-2 |
| 1911944-1\IMG\_20211108\_151426.jpg | 1911944-1 |
| 1911944-1\IMG\_20211108\_151441.jpg | 1911944-1 |
| 1911944-1\IMG\_20211108\_151449.jpg | 1911944-1 |
| 1911944-1\IMG\_20211108\_151727.jpg | 1911944-1 |
| 1911944-1\IMG\_20211108\_151751.jpg | 1911944-1 |
| 2010030-1\IMG\_20211107\_164823.jpg | 2010030-1 |
| 2010030-1\IMG\_20211107\_164826.jpg | 2010030-1 |
| 2010030-1\IMG\_20211107\_164927.jpg | 2010030-1 |
| 2010030-1\IMG\_20211107\_164955.jpg | 2010030-1 |
| 2010030-1\IMG\_20211107\_165007.jpg | 2010030-1 |
| 2010588-1\IMG\_20211029\_095507.jpg | 2010588-1 |
| 2010588-1\IMG\_20211029\_095539.jpg | 2010588-1 |
| 2010588-1\IMG\_20211029\_095548.jpg | 2010588-1 |
| 2010588-1\IMG\_20211029\_095749.jpg | 2010588-1 |
| 2010588-1\IMG\_20211029\_095808.jpg | 2010588-1 |

**Appendix F**

**WORKING TITLE FORM**

Republic of the Philippines

**SOUTHERN LEYTE STATE UNIVERSITY**

Sogod, Southern Leyte

Website: [www.slsuonline.edu.ph](http://www.slsuonline.edu.ph)

Email: [slsumaincampus@gmail.com](mailto:slsumaincampus@gmail.com)

[op@slsuonline.edu.ph](mailto:op@slsuonline.edu.ph)

Telefax No. (053) 382-3294

***College of Computer Studies and Information Technology***

**Proponents/Researchers:**

|  |
| --- |
| 1) Alcesto, Myrelle R. |
| 2) Antegra, Alphonse Victor N. |
| 3) Antipolo, Neil Brian S. |
| 4) Calivoso, Jesheryll P. |

**Proposed Project Title:**

|  |
| --- |
| **Student Entrance Masked Face Recognition and Verification System** |

|  |  |
| --- | --- |
| **Submitted by:**      \_*\_\_*  \_ALCESTO, MYRELLE R. \_ \_\_\_  (Signature of Project Manager over printed name)    Date: \_\_\_JULY 2, 2021\_\_\_\_\_\_\_ | **Noted:**      \_\_\_\_\_\_MS. JANNIE FLEUR V. ORAÑO\_\_\_\_\_\_  (Signature of Adviser over printed name)    Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **Recommending Approval:**      \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Signature of Patent Searcher over printed name)    Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Approved:**      \_\_\_\_\_\_\_\_\_MR. ALEX C. BACALLA, DIT\_\_\_\_\_\_\_\_  (Signature of the Dean over printed name)    Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**CURRICULUM VITAE**

**Personal Data**

**Name:** ALCESTO, Myrelle R.

**Address:** Bunga, Padre Burgos, Southern Leyte

**Date of Birth:** September 15, 1999

**Email Address:** [myrellerufila@gmail.com](mailto:myrellerufila@gmail.com)

**Status:** Single

**Nationality:** Filipino

**Religion:** Roman Catholic

**Educational Background**

**Elementary Education**

2006-2012 Santo Rosario Elementary School

**Secondary Education**

2012-2016 San Roque National High School

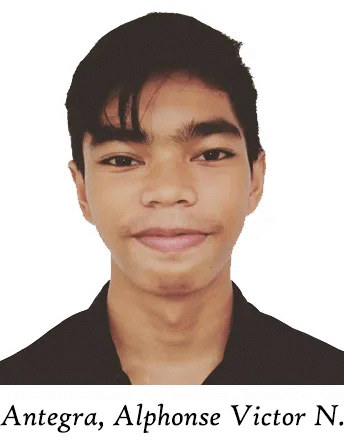
2016-2018 Saint Joseph College

**Tertiary Education**

2018-Present Southern Leyte State University – Main Campus

**CURRICULUM VITAE**

**Personal Data**

**Name:** ANTEGRA, Alphonse Victor N.

**Address:** Isagani, Maasin City, Southern Leyte

**Date of Birth:** July 2, 2000

**Email Address:** [alphonseantegra@gmail.com](mailto:alphonseantegra@gmail.com)

**Status:** Single

**Nationality:** Filipino

**Religion:** Roman Catholic

**Educational Background**

**Elementary Education**

2006-2012 Saint Joseph College

**Secondary Education**

2012-2016 Saint Joseph College

2016-2018 Saint Joseph College

**Tertiary Education**

2018-Present Southern Leyte State University – Main Campus

**CURRICULUM VITAE**

**Personal Data**

**Name:** ANTIPOLO, Neil Brian S.

**Address:** Looc, Tomas Oppus, Southern Leyte

**Date of Birth:** April 4, 1999

**Email Address:** [louxiaofensiao@gmail.com](mailto:louxiaofensiao@gmail.com)

**Status:** Single

**Nationality:** Filipino

**Religion:** Roman Catholic

**ANTIPOLO, Neil Brian S.**

**Educational Background**

**Elementary Education**

2006-2012 Maslog Elementary School

**Secondary Education**

2012-2016 SLSU Laboratory High School

2016-2018 Southern Leyte State University Senior High School– Tomas Oppus

**Tertiary Education**

2018-Present Southern Leyte State University – Main Campus

**CURRICULUM VITAE**

**Personal Data**

**Name:** CALIVOSO, Jesheryll P.

**Address:** Caridad Sur, Matalom, Leyte

**Date of Birth:** August 13, 1999

**Email Address:** [calivoso.jesheryll08@gmail.com](mailto:calivoso.jesheryll08@gmail.com)

**Status:** Single

**Nationality:** Filipino

**Religion:** Roman Catholic

**Educational Background**

**Elementary Education**

2006-2012 Januario P. Olo Memorial Elementary School

**Secondary Education**

2012-2016 Matalom National High School

2016-2018 Matalom National High School

**Tertiary Education**

2018-Present Southern Leyte State University – Main Campus

**GLOSSARY**

**Convolutional Neural Network** – a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data.

**Deep Learning** – is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.

**Detection** - the action or process of identifying the presence of something concealed.

**Face Mask** - a protective mask covering the nose and mouth or nose and eyes.

**Face Recognition -** a method of identifying or verifying the identity of an individual using their face. Face recognition systems can be used to identify people in photos, video, or in real-time.

**Pandemic –** an outbreak of a pandemic disease

**Verification** – it is the process of establishing the truth, accuracy, or validity of something.

**Haar-Cascaded Detection-**  is a machine learning-based approach where a lot of positive and negative images are used to train the classifier.