Applied Information Technology And Computer Science Vol. 5 No. 1 (2024) 1310-1329

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

Real Time Face Recognition Attendance System Using Python and OpenCV

Yap Jin Wei[1] **and Mohammed Saeed Jawad**[1*]

[1]Fakulti Sains Komputer dan Teknologi Maklumat,
UniversitiTun Hussein Onn Malaysia, Parit Raja, BatuPahat, 86400, MALAYSIA

DOI: https://doi.org/10.30880/aitcs.2024.05.01.074
Received 22 June 2023; Accepted 25 June 2024; Available online 30 August 2024

Abstract: *The Real Time Face Recognition Attendance System using Python and OpenCV is a desktop application that utilizes facial recognition technology to track and manage attendance. The system was developed to improve on the weaknesses of existing attendance management methods which use manual and paper as the medium for attendance management. A prototype model was used to develop this project. Meanwhile, system development is achieved by using VS Code software and a MySQL database. The programming language used is Python and the face recognition algorithm uses the Local Binary Pattern Histogram (LBPH). The face detection is implemented by using the Haar Cascades classifier for object detection. In general, this system can help improve the effectiveness and efficiency of data management in educational institutions and keep it under control in an organized manner.*

Keywords: *Attendance System, Face Recognition, Local Binary Patterns Histograms, Haar Cascades, OpenCV*

1. Introduction

Attendance systems are used to track the presence or absence of people in a certain location and come in three main types: manual, computerized, and biometric. Manual systems use paper records or sign-in sheets, while electronic systems use computers or digital devices. Biometric systems, such as face recognition and fingerprint scanners, use unique physical characteristics to identify people and confirm their presence. Attendance systems can be useful for accurately tracking attendance, but there are privacy concerns and the potential for bias in the algorithms used by certain types of systems.

The human face is a unique representation of a person's identity[1]. Face recognition is a biometric method of identifying a person by comparing a real-time image with an image stored in a database. It is popular due to its ease of use and impressive performance. Face recognition technology is used in various contexts, including border crossings and social media platforms. At border crossings, it is used

*Corresponding author: saeed@uthm.edu.my

to compare a person's image on a biometric passport with their face[2], and Facebook[3] uses it to identify and categorize photos shared by its users.

The traditional method of recording attendance can be time-consuming and inconvenient, especially in large institutions and colleges. Some institutions use QR codes, but these can be difficult to read and require students to use their smartphones to scan them. As a result, many institutions are starting to use newer technologies like iris recognition, fingerprint recognition, and RFID to track attendance. These systems use various types of biometric data to identify individuals and confirm their attendance, making the process more efficient.

The objective of the current study is to create a facial recognition-based attendance system that can accurately track student attendance in the classroom and provide results as needed in real-time. The system is designed for usage in educational facilities and will make use of the LBPH [4]algorithm from OpenCV[5]. The scope of this project is focused on students. There are several modules contained in this system, namely the dataset module, training the dataset module, face recognition module, and marking attendance module.

2. Literature Review

2.1 Haar Cascade Classifier

The Haar classifier is a machine learning object identification algorithm used to detect objects in images and videos. For training in this cascade function, a huge number of positive and negative images are used. The trained classifier is used to recognize certain objects in other images. The algorithm can be explained in four stages:

1) Haar Feature: The initial stage is the collecting of Haar features. Haar feature in the detection window are the result of calculations on adjacent rectangular portions. To determine the differences between and, the pixel intensities of each region must first be summed together. Identifying these aspects in a huge shot can be difficult. This is where integral image comes in, as they can reduce the number of procedures. **Figure 1** shows the Haar Feature originally developed by Viola and Jones[6]. **Figure 2** shows a face is detected using Haar Feature.

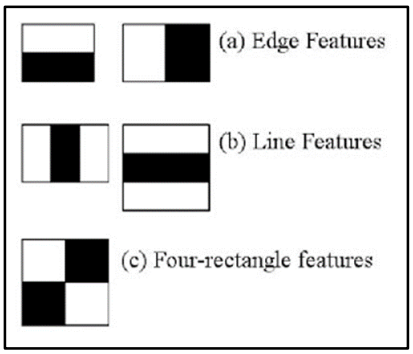


Figure 1: Haar Features

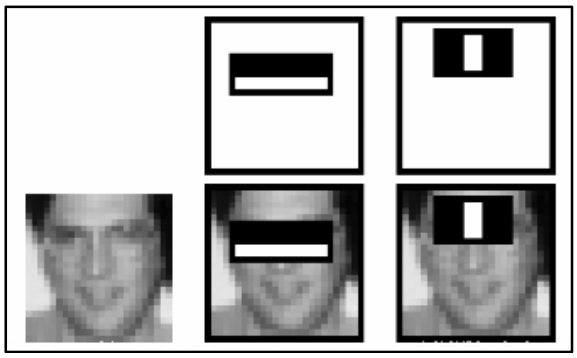


Figure 2: Face detection using Haar Feature

2) Integral Image: An integral image can be used to determine the sum of an image's pixel values rapidly and efficiently[7].

3) Adaboost: AdaBoost is a feature selection approach that enables you to choose a small subset of a big feature set while rejecting any features that perform poorly[8].

4) Cascading Classifier: Each weak classifier functions as a stage in the strong classifier's cascade. This cascade's job is to immediately eliminate non-faces, preventing the loss of significant time and computational effort.

2.2 Local Binary Patterns Histograms (LBPH)

The LBPH (Local Binary Pattern Histogram) algorithm is well-known for its ability to detect a person's face from multiple angles. It works on the principle of local features, comparing binary pixel intensities between the centre pixel and the eight pixels surrounding it. LBP efficiently captures texture information by comparing the threshold values of neighbouring pixels to the centre pixel. LBP, which was first introduced in 1994, has evolved into a highly effective texture classification system, particularly when paired with histograms of directed gradient descriptors. On the same dataset, this combination improves its accuracy. LBP also provides other features such as monotonic grey-scale enhancements and statistical simplicity, allowing for real-time image interpretation in a variety of applications [9]. **Figure 3** shows the LBPH algorithm for face recognition.

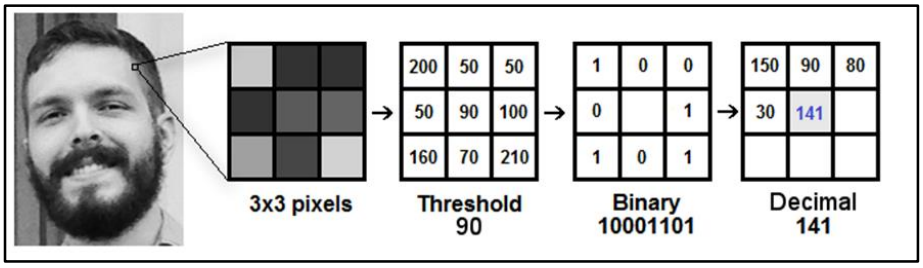


Figure 3: LBPH Algorithm for Face Recognition

This section presents a review of information or ideas related to the face detection attendance management system. The summary of survey from the existing research is summarized in **Table 1**.

Table 1: Comparison between existing systems with proposed system

System/Features	**Smart Attendance System using OPENCV based on Facial Recognition**[10]	**Real Time Automatic Attendance System for Face Recognition Using Face API and OpenCV**[11]	**Facial Recognition Attendance System Using Python and OpenCV**[12]	**Real Time Face Recognition Attendance System using Python and OpenCV**
User Login and Registration	√	√	√	√
Face Recognition Algorithms	Local Binary Patterns Histograms (LBPH)	YOLO V3	EigenFaces	Local Binary Patterns Histograms (LBPH)
Student Details Management	X	X	X	√
Attendance Report	√	√	√	√
Database	√	√	√	√

3. Methodology

The prototype method can carry out the main phases that exist in the system development life cycle including five other phases, namely the design phase, analysis phase, design phase, prototype development phase and implementation phase. **Table 2** shows the activities of the system development phases.

Table 2: Software development activities and their task

Phase	Task	Output
Planning	Proposed the project, determine the project schedule, activities, and output	Project proposal, Gantt Chart
Analysis	Information gathering and analysis	System requirements, Use Case Diagram, Entity Relationship Diagram
Design	Produce system design documents	System architecture, Data dictionary and schema, User interface design
Implementation	System programming code writing Run tests on the system and fix system errors	Program code, Test case
Prototype 1	Detect errors on the system and repair the existing system	System prototype
	Repeat from the planning phase until implementation phase	
Prototype 2	Correct errors once again on the system and update.	System prototype

System requirements analysis involves tasks that are done to understand user needs for a system before the system is developed. Throughout the analysis phase, project requirements refer to the users' perspective during interviews with relevant parties. The processes involved are Use Case Diagram and Class Diagram.

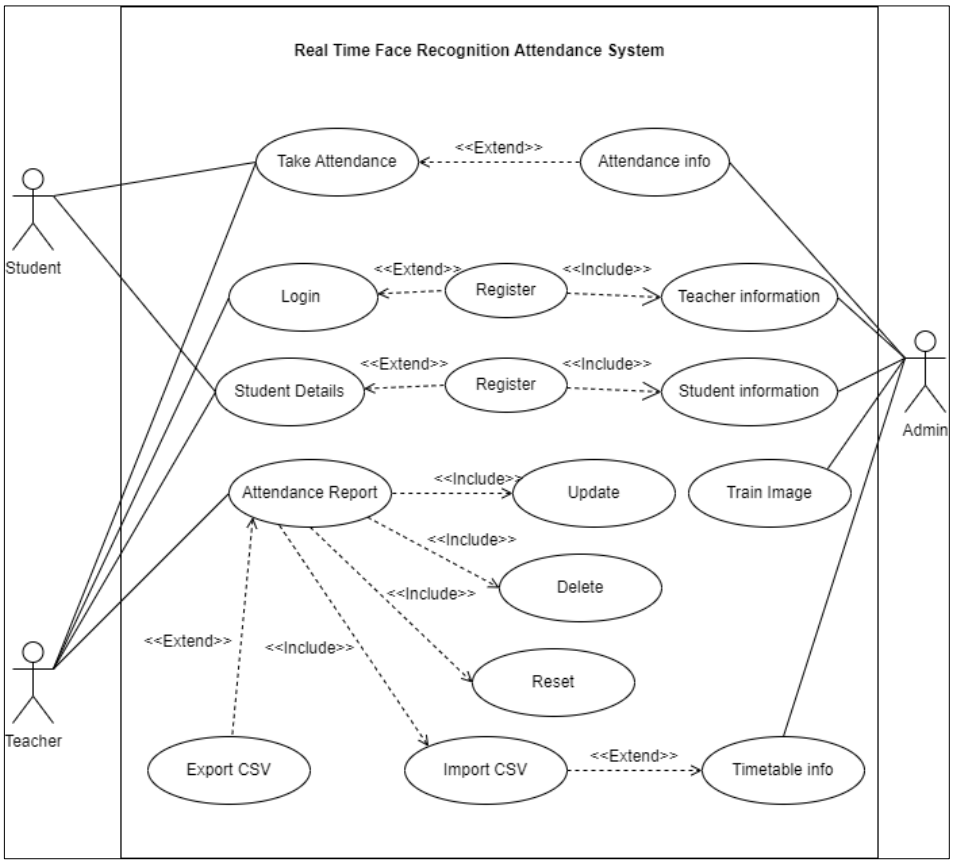


Figure 4: Use Case Diagram

The use case diagram shows an overview of the interaction between the system and its users. The use case diagram also shows the input and output to and from the user and the system. **Figure 4** shows the use case diagram of the developed system. The are three actors which are administrators, student, and teachers. There are 6 use cases that demonstrate how the various components of the real time face recognition attendance system work. Login details, student details, take attendance, attendance report will be verified by teachers will be displayed for admin to see. While the train image is only confirmed by the administrator. Student details and attendance will be confirmed by the student.

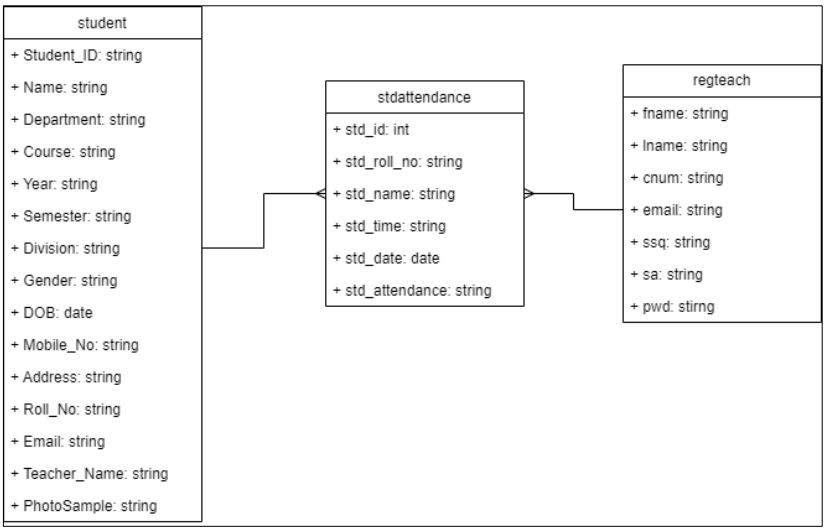


Figure 5: Class Diagram

The Class Diagram is used to assist in developing code for the creation of software applications. **Figure 5** shows the class diagram of the real time face recognition attendance system that includes regteach, student, and stattendance. Teachers represent the regteach. Students represent the student. Attendance represents the stattendance.

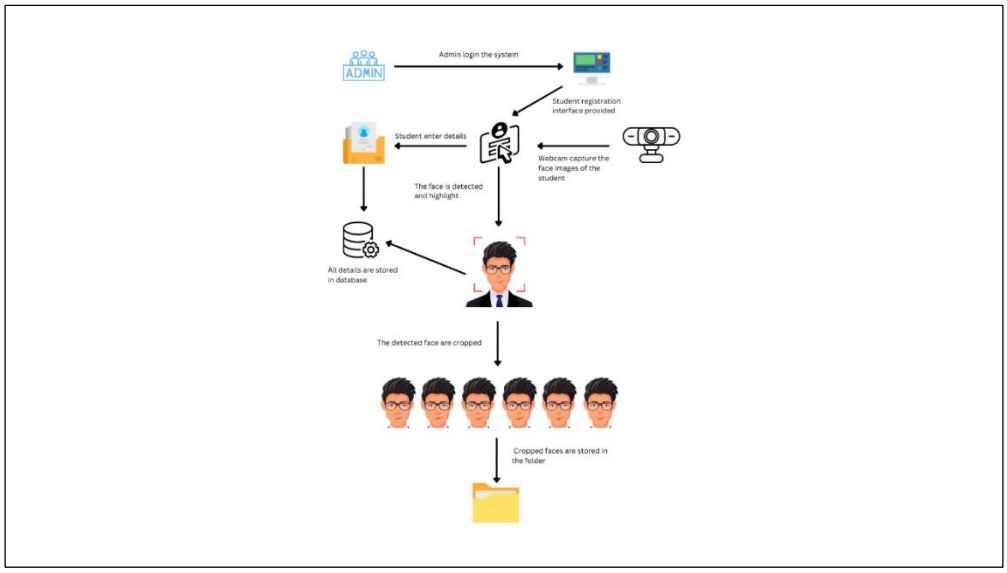


Figure 6: System Architecture Diagram

The system architecture diagram is used to show how each component of module in a system relates to one another. This depicted along of diagram is include the interaction between hardware and software. **Figure 6** shows the overall system architecture diagram of real time face recognition attendance system. Firstly, images of the students who are present in class are captured. Then, a face detection algorithm is utilized to enhance the performance of the system. After detecting a face, pre-processing techniques such as histogram equalization are applied to the extracted face image. A database is created to contain each student's training photos. To accurately recognize faces, feature extraction and categorization approaches are used. Finally, the system updates the database with the recognized student's information, including date, in time, and out time, in the post-processing stage.

The relational schema for the database tables is listed as follows:

i. `regteach (fname, lname, cnum, email, ssq, sa, pwd).`

ii. `stdattendance (std_id, std_roll_no, std_name, std_time, std_date, std_attendance).`

iii. `student (Student_ID, Name, Department, Course, Year, Semester, Division, Gender, DOB, Mobile_No, Address, Roll_No, Email, Teacher_Name, PhotoSample).`

Figure 7 shows the system register interface where the user needs to fill in all the required fields such as first name to register to the system. **Figure 8** shows the system login interface where the user needs to fill in the correct user credentials such as username and their respective passwords to log in to the system. **Figure 9** shows the main page interface of the system, users can click all the buttons to take them to related pages. **Figure 10** shows the student details interface of the system where the user needs to save, update, delete, reset, take pictures, and search for the information.

Figure 11 shows the system train image interface where the user can train the samples for detection and recognition. **Figure 12** shows the system take attendance interface where the user can recognize the face and mark the attendance. **Figure 13** shows the attendance report interface of the system where the user can view, update, reset, import, and export the report.

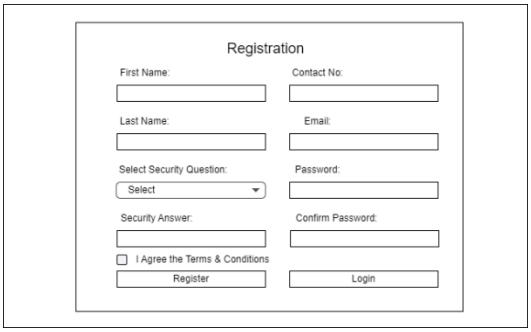


Figure 7: Register Module

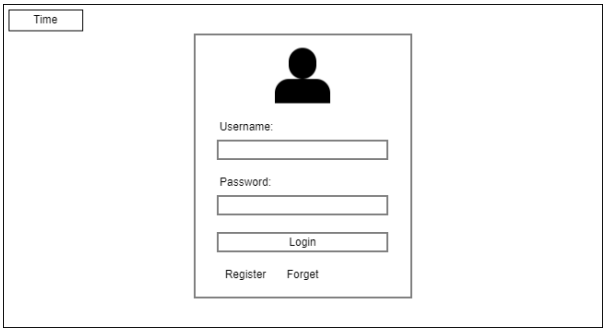


Figure 8: Login Module

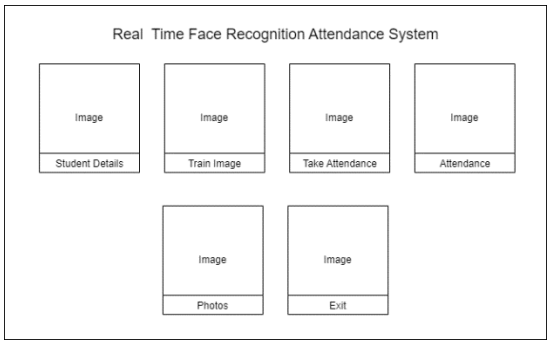


Figure 9: Main Page Module

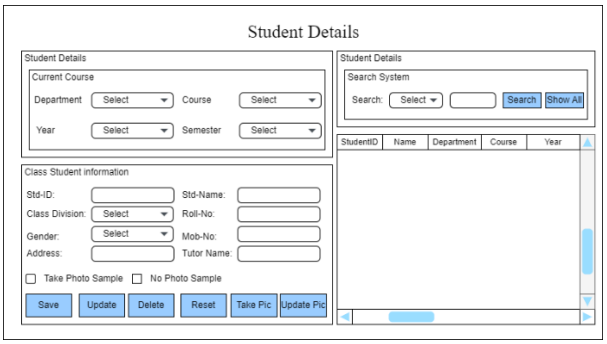


Figure 10: Student Details Module

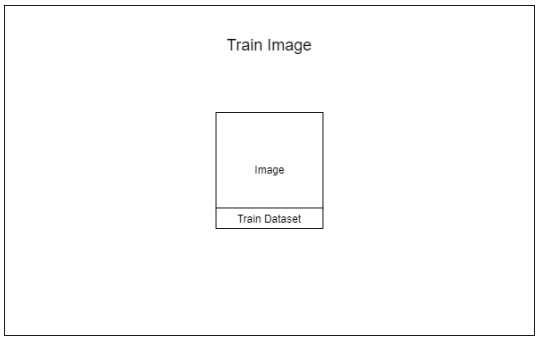


Figure 11: Train Image Module

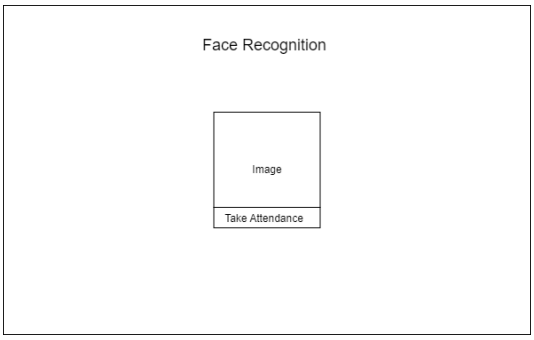


Figure 12: Take Attendance Module

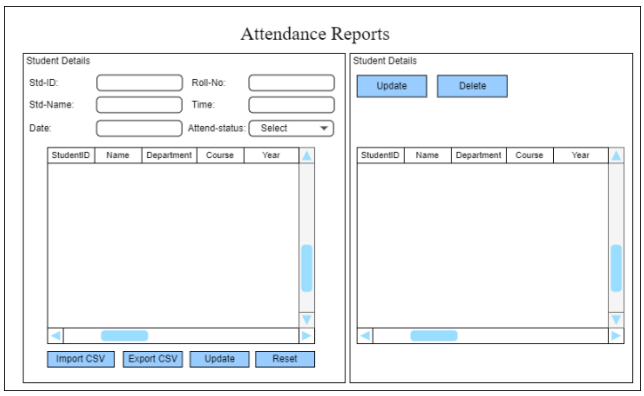


Figure 13: Attendance Report Module

4. Implementation and Testing

This section explains the development of functional modules in the system and its testing.

4.1 System Implementation

System programming has been implemented based on findings from the system analysis and design phase. Program code is provided to aid explanation.

1) Register Module

```
def reg(self):
    if (self.var_fname.get()=="" or self.var_lname.get()=="" or self.var_cnum.get()=="" or self.var_email.get()=="" or self.var_ssq.get
        messagebox.showerror("Error","All Field Required!")
    elif(self.var_pwd.get() != self.var_cpwd.get()):
        messagebox.showerror("Error","Please Enter Password & Confirm Password are Same!")
    elif(self.var_check.get()==0):
        messagebox.showerror("Error","Please Check the Agree Terms and Conditons!")
    else:
        # messagebox.showinfo("Successfully","Successfully Register!")
        try:
            conn = mysql.connector.connect(username='root', password='root',host='localhost',database='face_recognition',port=3306)
            mycursor = conn.cursor()
            query=("select * from regteach where email=%s")
            value=(self.var_email.get(),)
            mycursor.execute(query,value)
            row=mycursor.fetchone()
            if row!=None:
                messagebox.showerror("Error","User already exist,please try another email")
            else:
                mycursor.execute("insert into regteach values(%s,%s,%s,%s,%s,%s,%s,%s)",(
                self.var_fname.get(),
                self.var_lname.get(),
                self.var_cnum.get(),
                self.var_email.get(),
                self.var_ssq.get(),
                self.var_sa.get(),
                self.var_pwd.get()
                ))

                conn.commit()
                conn.close()
                messagebox.showinfo("Success","Successfully Registerd!",parent=self.root)
        except Exception as es:
            messagebox.showerror("Error",f"Due to: {str(es)}",parent=self.root)
```

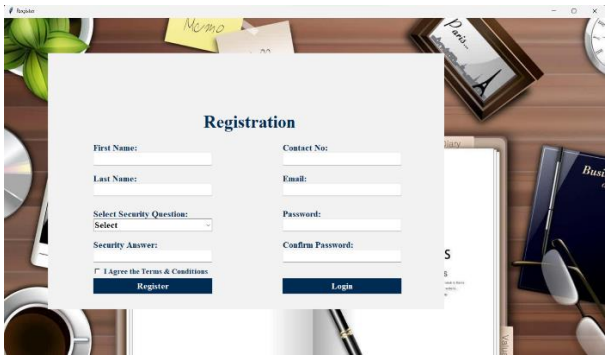


Figure 14: Register Source Code **Figure 15: Register User Interface**

Figure 14 shows the source code for the register function. The user interface consists of an input box that takes first name, last name, contact no, email, password, confirm password, one drop down list, and two buttons. The interface view of this module is shown in **Figure 15**.

2) Login Module

```
def login(self):
    if (self.txtuser.get()=="" or self.txtpwd.get()==""):
        messagebox.showerror("Error","All Field Required!")
    elif(self.txtuser.get()=="admin" and self.txtpwd.get()=="admin"):
        messagebox.showinfo("Sussessfully","Welcome to Face Recognition Attendance System")
    else:
        # messagebox.showerror("Error","Please Check Username or Password !")
        conn = mysql.connector.connect(username='root', password='root',host='localhost',database='face_recognition',port=3306)
        mycursor = conn.cursor()
        mycursor.execute("select * from regteach where email=%s and pwd=%s",(
            self.txtuser.get(),
            self.txtpwd.get()
        ))
        row=mycursor.fetchone()
        if row==None:
            messagebox.showerror("Error","Invalid Username and Password!")
        else:
            open_min=messagebox.askyesno("YesNo","Access only Admin")
            if open_min>0:
                self.new_window=Toplevel(self.root)
                self.app=Face_Recognition_System(self.new_window)
            else:
                if not open_min:
                    return
        conn.commit()
        conn.close()
```

Figure 16: User Login Source Code

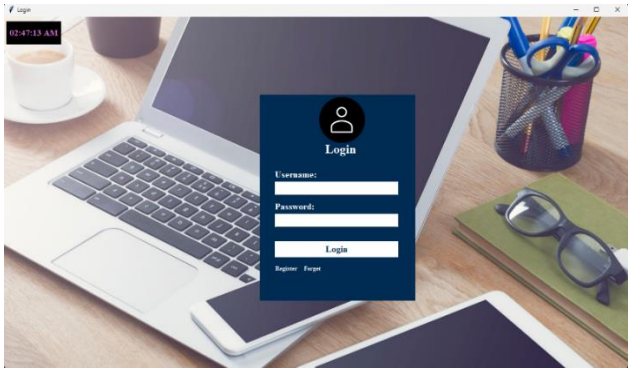


Figure 17: Login User Interface

Figure 16 shows the source code for the login function. The user interface consists of an input box that takes username and password input, as well as two buttons. The interface view of this module is shown in **Figure 17**.

3) Main Page Module

```
# ================Funtion for Open Images Folder=================
    def open_img(self):
        os.startfile("data_img")
# ================Functions Buttons=================
    def student_pannels(self):
        self.new_window=Toplevel(self.root)
        self.app=Student(self.new_window)

    def train_pannels(self):
        self.new_window=Toplevel(self.root)
        self.app=Train(self.new_window)

    def face_rec(self):
        self.new_window=Toplevel(self.root)
        self.app=Face_Recognition(self.new_window)

    def attendance_pannel(self):
        self.new_window=Toplevel(self.root)
        self.app=Attendance(self.new_window)

    def Close(self):
        root.destroy()
```

Figure 18: Main Page Source Code

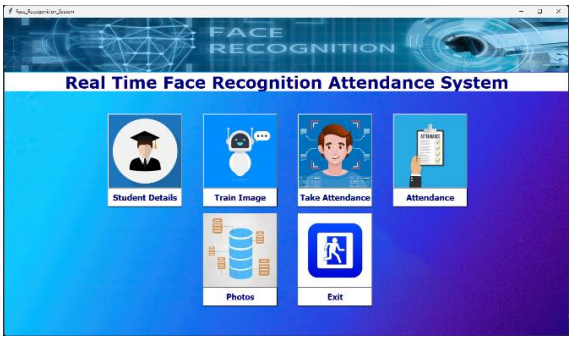


Figure 19: Main Page User Interface

Figure 18 and **Figure 19** show the source code and the user interface for the main page module. The user interface module for the main page function provides a way to navigate between different panels or windows within the GUI application. The design of the user interface and the layout depends on the code used.

4) Student Details Module

Figure 20 and **Figure 21** show the source code and user interface pages for the student details module. This module uses the " add_data" function " generate_dataset " in a class to display message boxes and capturing input from the user. Some of the steps performed in this module include handling business logic such as save, update, delete, reset, take picture, and search student information data.

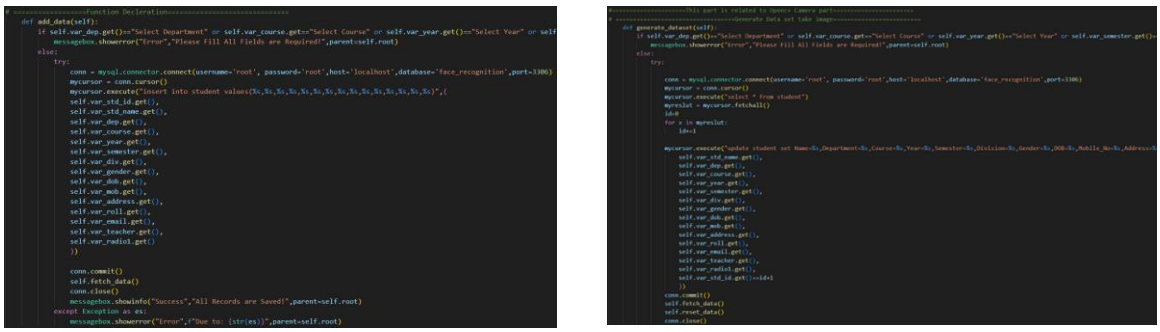


Figure 20: Student Details Source Code

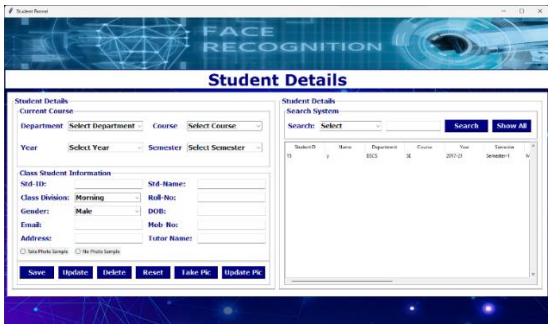


Figure 21: Student Details User Interface

5) Train Image Module

Figure 22 and **Figure 23** show the source code and user interface pages for the train image module. This module uses the " train_classifier " function used to control the train image module.

```python
#================Create Function of Traing==================
def train_classifier(self):
    data_dir=("data_img")
    path=[os.path.join(data_dir,file) for file in os.listdir(data_dir)]

    faces=[]
    ids=[]

    for image in path:
        img=Image.open(image).convert('L') # conver in gray scale
        imageNp = np.array(img,'uint8')
        id=int(os.path.split(image)[1].split('.')[1])

        faces.append(imageNp)
        ids.append(id)

        cv2.imshow("Training",imageNp)
        cv2.waitKey(1)==13

    ids=np.array(ids)

    #================Train Classifier==============
    clf= cv2.face.LBPHFaceRecognizer_create()
    clf.train(faces,ids)
    clf.write("clf.xml")

    cv2.destroyAllWindows()
    messagebox.showinfo("Result","Training Dataset Complated!",parent=self.root)
```

Figure 22: Train Image Source Code

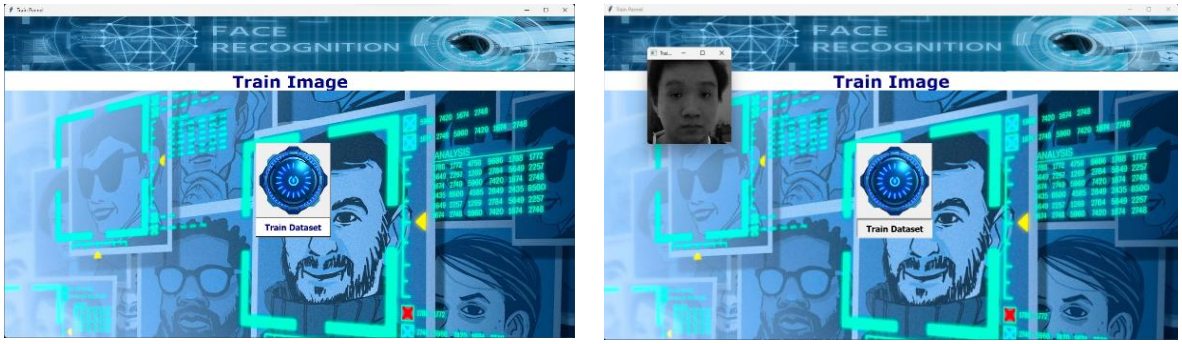


Figure 23: Train Image User Interface

6) Take Attendance Module

Figure 24 and **Figure 25** show the source code and user interface pages for the take attendance module. This module uses the " face_recog " function conducts facial identification on an input image.

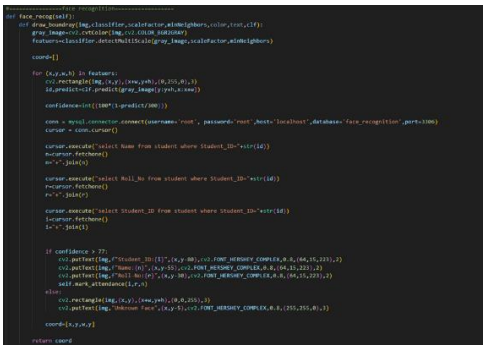


Figure 24: Take Attendance Source Code

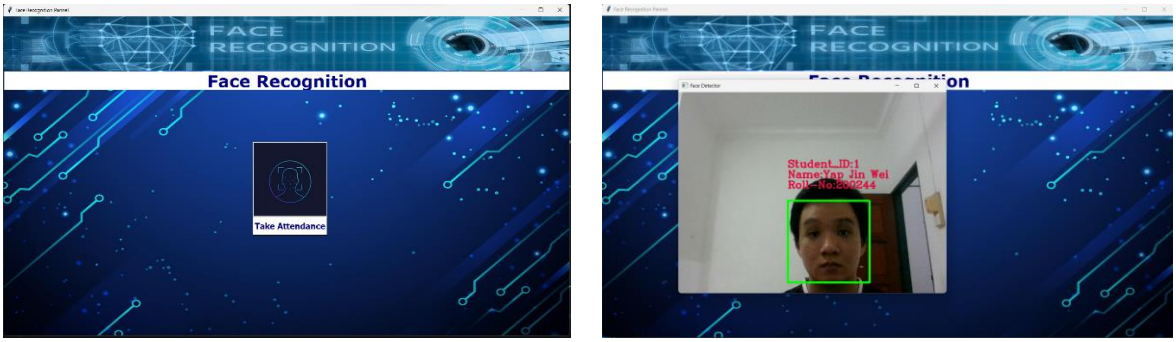


Figure 25: Take Attendance User Interface

7) Attendance Report Module

Figure 26 and **Figure 27** show the source code and user interface pages for the take attendance module. This module uses the "importCsv"and "exportCsv" functions to import attendance data from a CSV file and export the attendance data from the MySQL database to a CSV file.

```
def importCsv(self):
    mydata.clear()
    fln=filedialog.askopenfilename(initialdir=os.ge
    with open(fln) as myfile:
        csvread=csv.reader(myfile,delimiter=",")
        for i in csvread:
            mydata.append(i)
    self.fetchData(mydata)
```

```
def exportCsv(self):
    try:
        if len(mydata)<1:
            messagebox.showerror("Error","No Data Found!",parent=self.root)
            return False
        fln=filedialog.asksaveasfilename(initialdir=os.getcwd(),title="Open CSV"
        with open(fln,mode="w",newline="") as myfile:
            exp_write=csv.writer(myfile,delimiter=",")
            for i in mydata:
                exp_write.writerow(i)
            messagebox.showinfo("Successfuly","Export Data Successfully!")
    except Exception as es:
            messagebox.showerror("Error",f"Due to: {str(es)}",parent=self.root)
```

Figure 26: Attendance Report Source Code

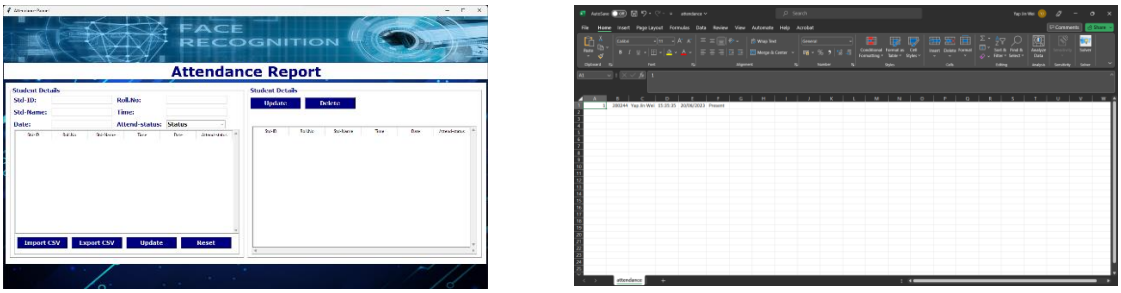


Figure 27: Attendance Report User Interface

4.2 System Testing

In this section, a test will be conducted to evaluate the functionality of each module. The User Acceptance Testing (UAT) method is used to conduct the test. **Table 3** to **Table 10** show the results of functional testing. The test results show that all functional modules can work correctly as intended.

Table 3: Test Case for User Register Module

Account Register Module				
Test Case ID	Description	Expected Decision	Actual Decision	Pass/No Pass
M1-1	Verifies whether the user can register with the correct credentials.	Users should be able to register, and a message should pop up saying that they have successfully registered.	The user successfully registers and a message pop up saying that they have successfully registered.	Pass
M1-2	Check if the system will restrict login when incorrect credentials are entered.	The system should display an error message indicating the register credentials are incorrect and prevent the user from registering.	The system displays an error message and restricts register when incorrect credentials are entered.	Pass

Table 4: Test Case for User Login Module

Account Login Module				
Test Case ID	Description	Expected Decision	Actual Decision	Pass/No Pass
M2-1	Verifies whether the user can log in with the correct credentials.	Users should be able to log in and be directed to the main page.	The user successfully logs in and is directed to the main page.	Pass
M2-2	Check if the system will restrict log in when incorrect credentials are entered.	The system should display an error message indicating the login credentials are incorrect and prevent the user from logging in.	The system displays an error message and restricts login when incorrect credentials are entered.	Pass
M2-3	Check whether the system will reset the password if the user forgets the password.	Users should be able to reset their password and log in again.	User resets password and logs in again.	Pass

Table 5: Test Case for Main Page Module

Main Page Module				
Test Case ID	Description	Expected Decision	Actual Decision	Pass/No Pass
M3-1	Verify the buttons, images, and links display correctly, are interactive, and responsive.	The system should display all buttons, images, and links correctly.	The system displays all buttons, images, and links correctly.	Pass
M3-2	Check the integration with other modules.	The system should be able to interact well with other modules, so that contact and data exchange go smoothly and without mistakes.	The system interacts well with other modules and contact, and data exchange go smoothly and without mistakes.	Pass

Table 6: Test Case for Student Details Module

Student Details Module				
Test Case ID	Description	Expected Decision	Actual Decision	Pass/No Pass
M4-1	Verify a student face towards the web camera to take an image.	The images should be able to capture and store.	The images are captured and stored.	Pass
M4-2	Check the system linked to the database with students' and attendance data.	There system should be displaying the student information and attendance data.	There system display the student information and attendance data.	Pass
M4-3	Check whether the system efficiently stores and retrieves student data.	User should be able to save, update, delete, reset, and search the attendance data.	The user successfully saves, update, delete, reset, and search the attendance data.	Pass

Table 7: Test Case for Train Image Module

Train Image Module				
Test Case ID	Description	Expected Decision	Actual Decision	Pass/No Pass
M5-1	Verify the stored images of a student's face.	The system should be able to train with the images (around 100).	The system successfully trains with the images (around 100).	Pass

Table 8: Test Case for Take Attendance Module

Take Attendance Module				
Test Case ID	Description	Expected Decision	Actual Decision	Pass/No Pass
M6-1	Verify the frontal face of a student for taking attendance.	If the face is recognised, the student's id, name, and roll number must be notified in the GUI.	If the face is recognised, the student's id, name, and roll number notified in the GUI.	Pass
M6-2	Detect and recognize face.	Detection of the face and recognition if records match in the database.	Detection of the face and recognition if records match in the database.	Pass
M6-3	Recognition of faces under low lighting conditions.	The system should be able to recognize the face under low level of lighting conditions.	The system successfully recognizes the face under low level of lighting conditions.	Pass
M6-4	Recognition of faces with and without spectacles.	The system should be able to recognize the face with or without spectacles.	The system recognized the face with or without spectacles.	Pass

Table 9: Test Case for Attendance Report Module

Attendance Report Module				
Test Case ID	Description	Expected Decision	Actual Decision	Pass/No Pass
M7-1	Verify whether the attendance data report can be managed successfully.	The attendance data report should be properly organised and display complete information.	The attendance data report can be managed correctly and displays complete information.	Pass
M7-2	Information retrieval	The selected day's student information and attendance data should be displayed.	The selected day's student information and attendance data are displayed.	Pass

4.2 User Acceptance Test Results

User testing is a technique that will be used to determine the extent to which this system will perform as expected. This testing ensures that all functionalities used work optimally in accordance with the user's demands and preferences.

4.3.1 Register Module

Figure 28 shows a pie chart displaying the percentage of user testing on the register for the administrator of the system respectively. Based on the conducted tests with 1 respondent, it indicates that 100 percent of users passed the registration process in this section. Therefore, this proves that users were able to successfully register during the account registration process.

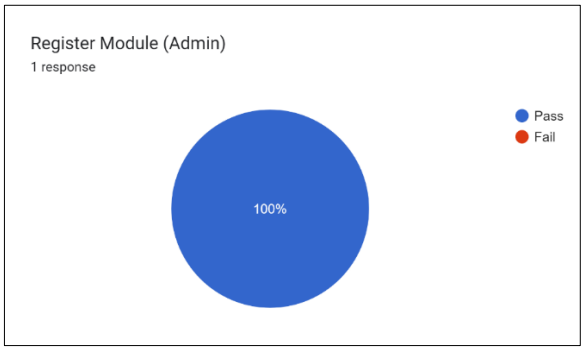


Figure 28: Pie Chart of Register Module (Administrator) Testing

4.3.2 Login Module

Figure 29 shows a pie chart displaying the percentage of user testing on the register for the administrator of the system respectively. Based on the conducted tests with 1 respondent, it indicates that 100 percent of users passed the login process in this section. Therefore, this proves that users were able to successfully log in again using the username and password set during the account registration process.

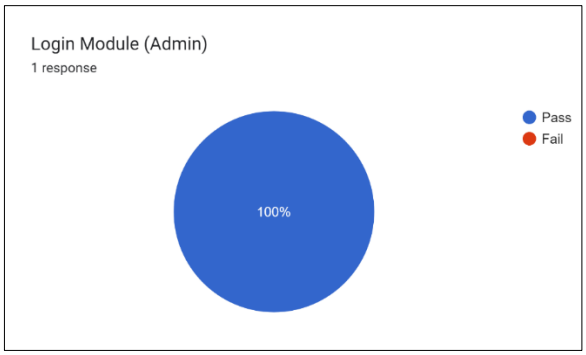


Figure 29: Pie Chart of Login Module (Administrator) Testing

4.3.3 Main Page Module

Figure 30 shows a pie chart displaying the percentage of user testing on the register for the administrator of the system respectively. Based on the conducted tests with 1 respondent, it indicates that 100 percent of users passed the login process in this section. Therefore, this proves that users were able to navigate different modules.

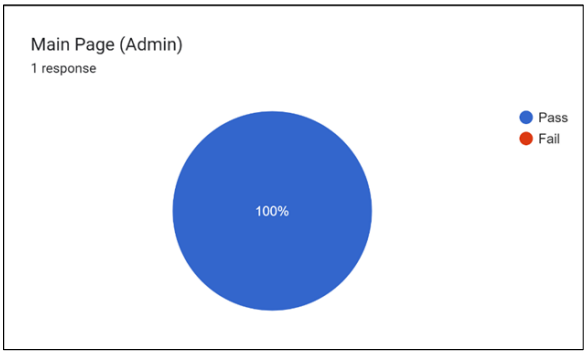


Figure 30: Pie Chart of Main Page Module (Administrator) Testing

4.3.4 Student Details Module

Figure 31 and **Figure 32** show a pie chart displaying the percentage of user testing on the student details module for the student and administrator of the system respectively. Based on the conducted tests with 10 respondents, it indicates that 100 percent of users passed the student details module in this section. For administrator section, 1 respondent passed the student details module. Therefore, this proves that users were able to view their information and take pictures while the administrator can manage the student in student details module.

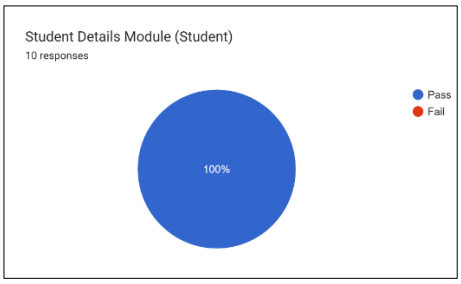


Figure 31: Pie Chart of Student Details Module (Student) Testing

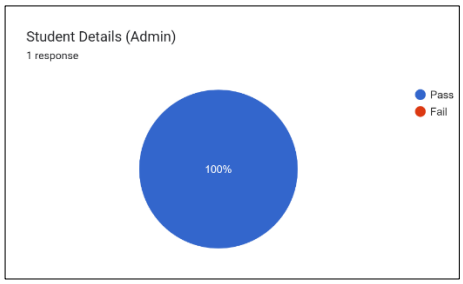


Figure 32: Pie Chart of Student Details Module (Administrator) Testing

4.3.4 Train Image Module

Figure 31 shows a pie chart displaying the percentage of user testing on the train image for the administrator of the system. Based on the conducted tests with 1 respondent, it indicates that 100 percent of users passed the login process in this section. Therefore, this proves that users were able to store images of a student's face.

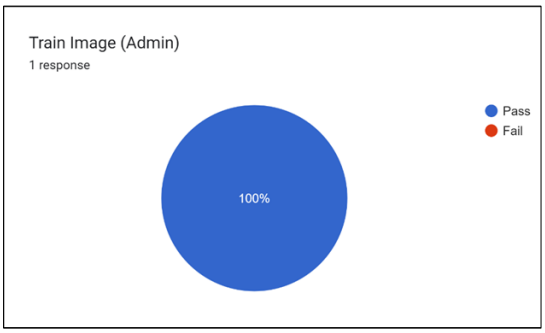


Figure 31: Pie Chart of Train Image Module (Administrator) Testing

4.3.4 Take Attendance Module

Figure 32 and **Figure 33** show a pie chart displaying the percentage of user testing on the take attendance module for the student and administrator of the system respectively. Based on the conducted tests with 10 respondents, it indicates that 100 percent of users passed the take attendance module in this section. For the administrator section, 1 respondent passed take attendance module. Therefore, this proves that users were able verify the frontal face of a student for taking attendance and mark the attendance.

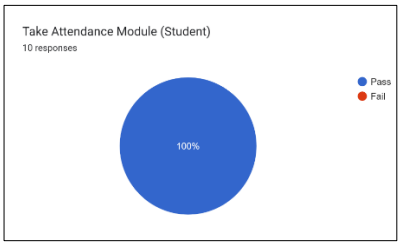


Figure 32: Pie Chart of Take Attendance Module (Student) Testing

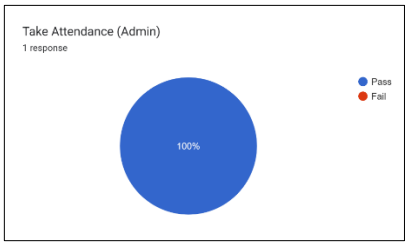


Figure 33: Pie Chart of Take Attendance Module (Administrator) Testing

4.3.4 Attendance Report Module

Figure 34 shows a pie chart displaying the percentage of user testing on the attendance report for the administrator of the system. Based on the conducted tests with 1 respondent, it indicates that 100 percent of users passed the login process in this section. Therefore, this proves that users were able to manage and generate attendance data reports.

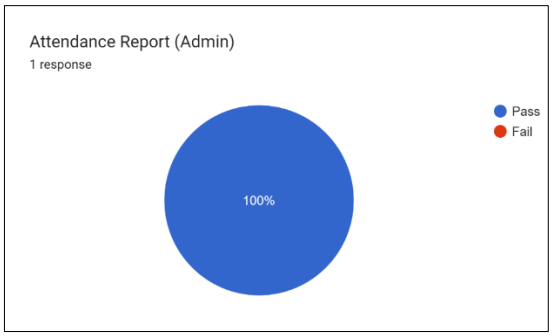


Figure 34: Pie Chart of Take Attendance Module (Administrator) Testing

The system testing is performed following system implementation. This testing was conducted to determine whether the system functions in accordance with the original system design and meets the goals and needs of the users. This testing phase involved a total of 11 respondents, including one administrator and ten students. Based on the results of the test cases, it can be concluded that the users effectively utilized the system. Nevertheless, there were still a few respondents who encountered difficulties or limitations when utilizing the system. Consequently, certain aspects have been enhanced to satisfy the needs of the users. Nevertheless, based on the percentages displayed in the pie chart, it appears that the test was successful overall.

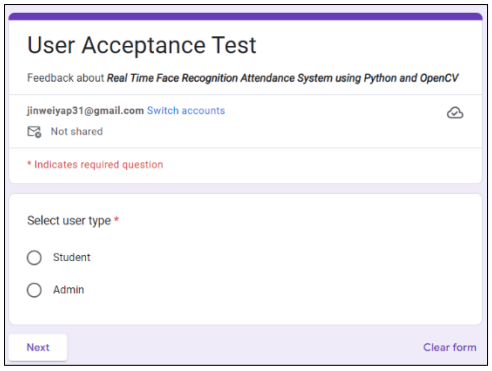
5. Conclusion

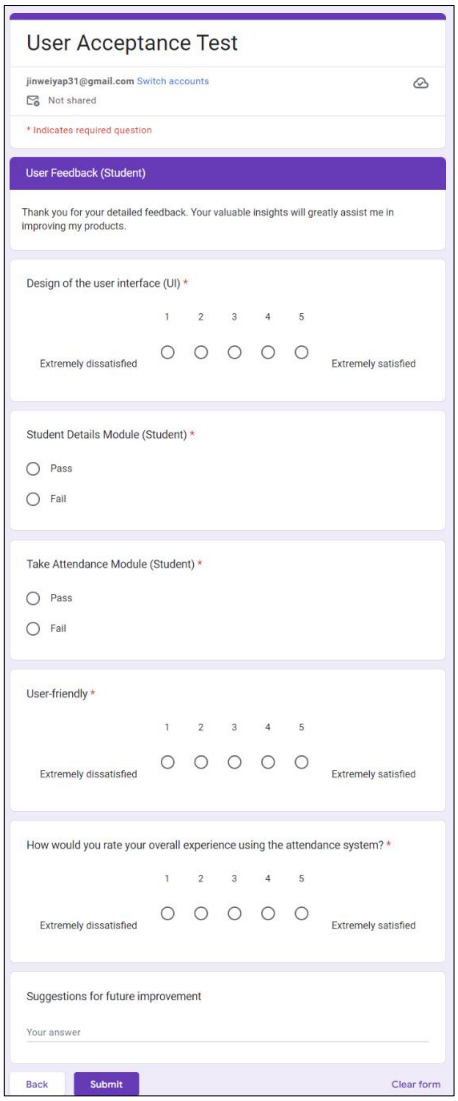
The Real Time Face Recognition Attendance System using Python and OpenCV was developed o manage student attendance. The system was able to accurately record the attendance of students in the classroom and produced the necessary real-time output in an Excel sheet. The system was designed to provide a simple and secure method of taking attendance. The software initially captured images of all authorized persons and stored the data in a database. The system then saved these photographs by mapping them to a facial coordinate structure. The technology identified and marked the presence of registered people the next time they entered the premises. The individual was identified, and their presence was recorded. The proposed system was developed for use by educational institutions.

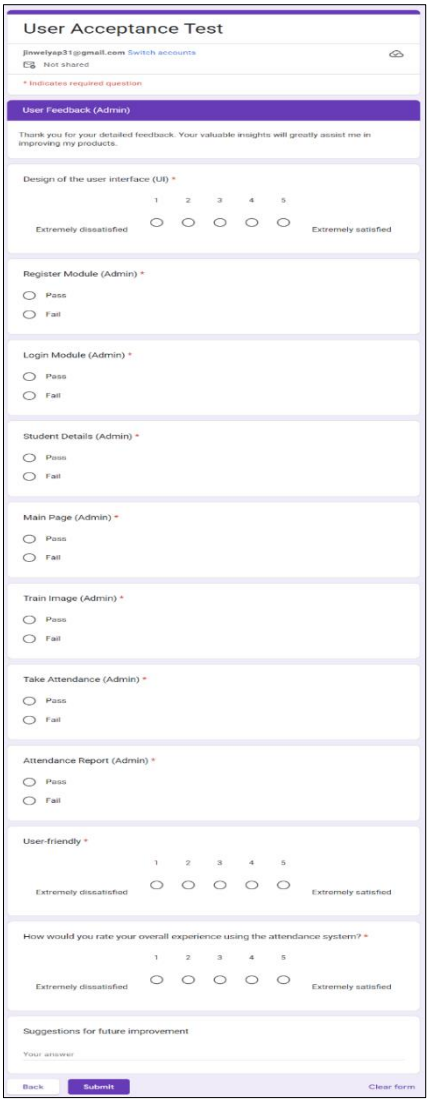
Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Appendix


User Acceptance Test
Feedback about Real Time Face Recognition Attendance System using Python and OpenCV
jinweiyap31@gmail.com Switch accounts
Not shared
* Indicates required question
Select user type *
Student
Admin
Next
Clear form


User Acceptance Test
jinweiyap31@gmail.com Switch accounts
Not shared
* Indicates required question
User Feedback (Student)
Thank you for your detailed feedback. Your valuable insights will greatly assist me in improving my products.
Design of the user interface (UI) *
1 2 3 4 5
Extremely dissatisfied
Extremely satisfied
Student Details Module (Student) *
Pass
Fail
Take Attendance Module (Student) *
Pass
Fail
User-friendly *
1 2 3 4 5
Extremely dissatisfied
Extremely satisfied
How would you rate your overall experience using the attendance system? *
1 2 3 4 5
Extremely dissatisfied
Extremely satisfied
Suggestions for future improvement
Your answer
Back
Submit
Clear form


User Acceptance Test
User Feedback (Admin)
Thank you for your detailed feedback. Your valuable insights will greatly assist me in improving my products.
Design of the user interface (UI) *
Extremely dissatisfied
Extremely satisfied
Register Module (Admin) *
Pass
Fail
Login Module (Admin) *
Pass
Fail
Student Details (Admin) *
Pass
Fail
Main Page (Admin) *
Pass
Fail
Train Image (Admin) *
Pass
Fail
Take Attendance (Admin) *
Pass
Fail
Attendance Report (Admin) *
Pass
Fail
User-friendly *
Extremely dissatisfied
Extremely satisfied
How would you rate your overall experience using the attendance system? *
Extremely dissatisfied
Extremely satisfied
Suggestions for future improvement
Your answer
Back
Submit
Clear form

References

[1] A. S. Redfern and C. P. Benton, "Representation of facial identity includes expression variability," *Vision Res*, vol. 157, pp. 123–131, Apr. 2019, doi: 10.1016/J.VISRES.2018.05.004.

[2] Thales Group, "Facial recognition: top 7 trends (tech, vendors, use cases)," Jun. 12, 2023. [Online]. Available: https://www.thalesgroup.com/en/markets/digital-identity-and security/ government/ biometrics/facial-recognition. [Accessed: 19-Jun-2023].

[3] Thales Group, "Where is facial recognition used?," 2022. [Online]. Available: https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/insp ired/ where-facial-recognition-used [Accessed: 19-Jun-2023].

[4] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3021, pp. 469–481, 2004, doi: 10.1007/978-3-540-24670-1_36/COVER.

[5] OpenCV, "Home - OpenCV," 2021. [Online]. Available: https://opencv.org/. [Accessed: 19-Jun-2023].

[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, doi: 10.1109/CVPR.2001.990517.

[7] Real Python, "Integral Images - Real Python," 2023. [Online]. Available: https://realpython.com/lessons/integral-images/. [Accessed: 19-Jun-2023].

[8] D. Polzer, "AdaBoost, Step-by-Step - Towards Data Science," Medium, Aug. 03, 2022. [Online]. Available: https://towardsdatascience.com/adaboost-in-7-simple-steps-a89dc41ec4. [Accessed: 19-Jun-2023].

[9] A. M. Jagtap, V. Kangale, K. Unune, and P. Gosavi, "A study of LBPH, eigenface, fisherface and haar-like features for face recognition using OpenCV," *Proceedings of the International Conference on Intelligent Sustainable Systems, ICISS 2019*, pp. 219–224, Feb. 2019, doi: 10.1109/ISS1.2019.8907965.

[10] S. Bussa, A. Mani, S. Bharuka, and S. Kaushik, "Smart Attendance System using OPENCV based on Facial Recognition," *International Journal of Engineering Research & Technology*, vol. 9, no. 3, Mar. 2020, doi: 10.17577/IJERTV9IS030122.

[11] S. Khan, A. Akram, and N. Usman, "Real Time Automatic Attendance System for Face Recognition Using Face API and OpenCV," *Wirel Pers Commun*, vol. 113, no. 1, pp. 469–480, Jul. 2020, doi: 10.1007/S11277-020-07224-2.

[12] V. Suresh, S. Chakravarthi Dumpa, C. Deepak Vankayala, and J. Rapa, "Facial Recognition Attendance System Using Python and OpenCv," *Quest Journals Journal of Software Engineering and Simulation*, vol. 5, no. 2, pp. 2321–3809, 2019, Accessed: Jan. 10, 2023. [Online]. Available: www.questjournals.org