An Access Control System using RFID and Face Verification with QR Code-based Enrollment

Ekarin Suethanuwong*, and Supachoke Sukkasame**

Received: December 21, 2022
Revised: April 11, 2023
Accepted: July 26, 2023

* Corresponding Author: Ekarin Suethanuwong, E-mail: ekarin.s@psu.ac.th
DOI: 10.14416/j.it.2023.v2.002

Abstract

The objective of this paper is to propose the system architecture and implementation of an access control system using RFID technology and face verification. The implemented system was employed to control user's access at the tripod turnstile gate in the library of the Prince of Songkla University (Trang Campus). The access control system is divided into two parts: enrolment and access control. In the enrolment part, the system obtains the user information from three processes: (1) the basic user information (e.g., full name and university identification number) is received via a SOAP web service, (2) the face image of a user is detected by using the camera of a mobile phone, (3) the UID data of a RFID card is read in a form of QR code by using our enrolment board. In the access control part, the RFID reader board forwards the UID data reading from an RFID card into the main controller board via RS-232 communication. The main controller board is used to send the UID data into the Flask web server in order to retrieve the corresponding face image encoding, and then verify with faces that are detected from the webcam at the tripod turnstile gate. The main controller board also controls the tripod turnstile gate via the push-pull solenoid and receives external signals from the proximity sensor and limit switch. The performance of the implemented system using the Raspberry Pi 4 Model B board as the main controller board in terms of verification accuracy and response time was measured, and also compared with the NVIDIA Jetson Nano board.

Keywords: Access Control, RFID, Face Verification, JSON Web Token, Raspberry Pi, NVIDIA Jetson Nano.

1. Introduction

Nowadays, the university library at Prince of Songkla University (Trang Campus) is a place that contains a collection of books, periodicals, journals, magazines, and education media for university students, staffs, and public people to read, borrow, or refer to. From the past until now, the library's access control has used only a push button to unlock the tripod turnstile gate as shown in Figure 1. In other words, a user who requires to access at the tripod turnstile gate only presses the push button. The function of the existing system cannot categorize people who are going to access into the library, i.e., it cannot block unauthorized people. The librarians have to periodically monitor people who are going to access into the library at the tripod turnstile gate in order to prevent public people who would like to use the library's services without paying the library fee. Moreover, it would be even more difficult to deal with if unauthorized people dress like the university's students even though the librarians monitor at the tripod turnstile gate more frequently.

Figure 1. *Tripod Turnstile Gate with the Push Button.*

* *Department of Information and Digital Technology Management, Faculty of Commerce and Management, Prince of Songkla University.*
** *Department of Digital Business, Faculty of Commerce and Management, Prince of Songkla University.*

Currently, every student and staff in the university must have their own identification cards, called as university cards. The university cards of the students are also ATM (Automated Teller Machine) cards, i.e., the university cards can be used to withdraw an amount of money from their bank accounts. Furthermore, the university cards of the students and staffs are attached with RFID (Radio Frequency Identification) tags, later called as RFID cards, so that they can utilize the RFID cards for user identification. However, the RFID cards cannot identify RFID card holders as RFID card owners correctly, if people use the RFID cards of other persons in order to get right to do something or access some places. On the other hand, personal identification using the biometric technology (e.g., fingerprint, face, iris, and so forth) is much more difficult to be used by persons who are not the biometric owners because the biometric identification employs a personal identity that is unique for each person [1].

As mentioned previously, the biometric technology seems to be promising to be used in access control systems. In [1-8], they show that the biometric technology has been widely adopted in access control systems. In contrast to the RFID technology [9], the biometric technology cannot provide all correct results for user identification, especially identifying with a large number of people, meaning that there might be some false results of user identification [10]. For examples, it could be that face recognition algorithms are unable to correctly identify people with very similar face images especially twins [11]. Variant lighting conditions during face image detection processes could cause face recognition algorithms to produce incorrect results for user identification [12]. For user identification using fingerprint, there might be some wrong outputs when users have their unclear friction ridges [13], [14]. On the contrary, user identification by using RFID cards produces accurate results due to the existence of error detection in RFID communication, and provides results faster than using biometric technology that needs recognition processes [1], [4]. Therefore, combining both RFID and biometric technology in access control systems could eliminate the drawbacks of each other.

Nowadays, fingerprint and face are the most widely used biometric technologies in access control systems [15], [16]. With fingerprint recognition, it is not suitable for access control systems in situations that there are many users who need to access at the same time. This is because each user needs some amount of time to place his or her finger on a fingerprint reader. In addition, there needs to be avoidance to contact a common fingerprint capturing device together in the pandemic situation of the COVID-19 (coronavirus disease 2019). On the contrary, using face recognition in access control systems, human faces can be detected and captured by a video camera immediately without touching it. Moreover, with the recent advancements in fingerprint and face recognition algorithms, the accuracies are achievable beyond 95 percentages for user identification [15], [17]. Such recognition accuracies were done under controlled quality images and certain environment conditions. In other words, the recognition results might produce different accuracies, depending on input image quality and environmental conditions especially lighting [12]. The completely correct user identification in access control systems using biometrics technology under a large number of users is needed and remains problematic. In this paper, our proposed system architecture therefore relies on user verification [1], [15], not identification, so that highly accurate matching results are achievable independently of a large number of users.

More recently, there has been advancement in small single-board computers, i.e., high computational power such as Raspberry Pi, Banana Pi, and NVIDIA Jetson Nano [15], [18-19]. The hardware specifications of the recent single-board computers are powerful enough to make processing times acceptable for real-time face verification [15], [19]. For example, the Raspberry Pi 4 Model B board that was first released in 2019 has a 1.5 GHz 64-bit quad-core ARM Cortex-A72 processor with on-board 802.11ac Wi-Fi, Bluetooth 5, one Gigabit Ethernet port, up to 8 GB of RAM, and dual-monitor support via two micro-HDMI ports [18]. In this paper, a small single-board computer was used to execute the face detection and verification functions in a real-time manner. The system architecture was designed for the university students and

staffs to be able to do self-enrolments by using their own mobile devices. All basic user information of the university students and staffs such as full names and university ID numbers can be retrieved by means of a SOAP (Simple Object Access Protocol) web service in which the SOAP web service provider of the university information system is called through the local area network of the university. The rest of this paper is organized as follows. The section 2 mentions existing attendance or access control systems incorporated with the RFID technology and face recognition along with their disadvantages. The proposed system architecture is described in the section 3. The section 4 explains the implementation of an access control system by using our proposed system architecture. The section 5 provides the performance evaluation of the implemented system and the comparative study with other existing systems. The conclusion and future works are given in the section 6.

2. Related Works

Akbar et al. [20] proposed and implemented a model of an automated attendance system using face identification incorporated with RFID. The operation of the system starts with identifying a face image detected from the webcam. If the face image can be identified with all face images existing in the database, the corresponding UID (Unique Identifier) in the database will be taken to compare the UID obtained from the RFID tag by using a RFID reader connected to the Arduino Uno R3 board. The main drawback of this model is that the identification process will take a larger amount of time if a number of face images in the database increases. In addition, their proposed system is a standalone system, i.e., it operates on its own without being connected to a computer network. In other words, this model was not designed to adopt the database in common if the automated attendance system is used for more than one place.

Sanath et al. [21] proposed a model of a smart attendance system using RFID and face recognition. The procedure of the proposed model begins with a validation of UID from the RFID tag of an employee by using the RFID reader connected to the ESP8266 board. If the UID is valid against its database, the temperature of the employee will be then compared with a specified threshold level. If the high temperature of the employee is found, an alert message with the employee's information will be sent to the concerned authorities. The OpenCV module with the HAAR Cascade classifier is used to detect faces in an image. The CNN architecture model is used for face recognition. Like [20], the disadvantage of this model is that an employee with a normal temperature goes through the process of the face identification, not verification. In addition, there is no information about time it takes for each process of their attendance system.

Kanna et al. [22] proposed and implemented an attendance system for students and staffs by using RFID technology and face recognition. The Raspberry Pi board (Pi 4 Model B) is used to receive a video streaming from a webcam (Logitech C270 HD), and then transfers it to the Amazon Web Service (AWS) cloud for image analysis (i.e., face detection and recognition) using the Amazon Rekognition API service. In the proposed system, the video processing for face recognition together with RFID detection must be done in a scheduled time interval. During this scheduled time interval, all detected face images by the Amazon Rekognition API service are stored in a specified folder in the AWS cloud. Once the RFID tag has been detected, the Raspberry Pi board takes its UID value to search for the corresponding user information in the Firebase database. If the UID value is found in the database, the identity photo from the database is taken and identified with the face images within the specified folder in the AWS cloud. Even though, the proposed system aims to improve the speed of the existing attendance management processes, there is no measurement result about processing time. The downside of the proposed system is that the time it takes for the matching process in the scheduled time interval will be increased if a number of detected faces in the AWS folder is larger, meaning that this proposed system is not suitable for a large number of users.

Kariapper [23] developed an automatic attendance system with two-factor verification that combines RFID and face identification processes. Once the RFID tag has been detected by the MFRC522 module, the UID value is read by the Arduino microcontroller. The camera module then takes a student picture for the face detection and identification processes. The MTCNN (Multi-Task Cascaded Convolutional Network) model was used for the face detection process. As a result of the MTCNN model, the facial landmark localization is found, i.e., left eye, right eye, nose, left mouth corner, and right mouth corner. If the users obtained from searching the UID value in the database and identifying the detected face are matched, the attendance will be granted. Similar to [20-22], the shortcoming of this automatic attendance system is that the face identification process verifies with all face images in the database. In other words, the face identification process takes a long and unacceptable time when a number of students who have already registered to use this system is large. This work did not also provide measurement results about the time it takes for the identification process. Like [20], the system is a standalone system that does not share the database in common to use in other classrooms at the same time.

Wahyudono and Ogi [24] implemented an access control system with a two-factor authentication based on RFID and face recognition using the LBP (Local Binary Pattern) algorithm. In this system, the Raspberry Pi (3B Model) board was used as a main microcontroller board. If the results from both RFID and face authentication processes are successful, the authentication dashboard using the 7-inch LCD touch screen will show the words "Access Granted!" together with the user information, and the system then activates the relay module to open the door gate. In the performance testing, the RFID authentication and face recognition result in the accuracies of 100% and 80%, respectively. In addition, the average times of RFID authentication and face recognition are 0.03 and 6.3885 seconds, respectively. So, the overall average time (i.e., 6.4185 seconds) is not a promising result to apply for an access control system with a large number of users.

Leyu et al. [15] presented the design and implementation of an access control system that combines RFID with double biometric technologies, (i.e., face and fingerprint authentications). The system is broken down into two parts: (1) the card-issuing part, and (2) the card reading and verification part. In the first part, the main control module (i.e., Raspberry Pi 4 Model B) receives the extracted face and fingerprint feature information of a user from the CSI camera of the Raspberry Pi board and the AS608 optical fingerprint recognition module, respectively. Such information is encrypted with the AES algorithm and stored into the MFRC522 chip of the RFID card. The system saves the user's basic information into a database via the Windows client software. In the second part, the open source OpenCV and Dlib libraries were adopted for face detection and recognition. For the performance testing, the card-issuing process achieved an average time of ten seconds. In addition, the system took approximately three seconds for automatic verification. The main distinction of this access control system from others in [20-24] is to enhance the authentication security by using double biometric types (i.e., face and fingerprint) and to store both encrypted biometric data of a user in a RFID card. The drawback of this system is that it cannot be used in a case where a RFID card is not allowed to write any data into it, e.g., some organizations using RFID-tagged organization cards combined with ATM cards have strict policies to not allow to write any data into the cards. In this paper, the system architecture and implementation of an access control system using RFID technology and face verification that eliminate the problems mentioned in this section are proposed.

3. System Architecture

The proposed system architecture of an access control system using RFID technology and face verification was designed under the following constraints. In the first constraint, the basic information of the university students and staffs, later called as university users, already exists in the information system of the university and being able to be accessed via a SOAP web service. SOAP is a lightweight protocol

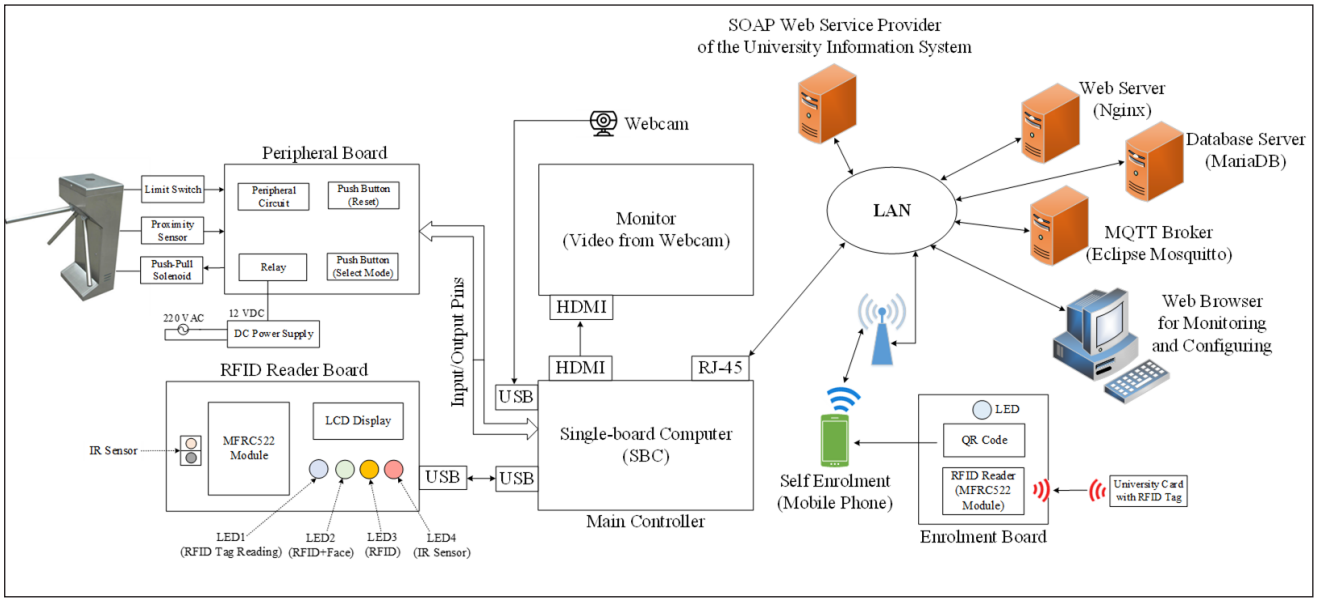

Figure 2. *The Proposed System Architecture of an Access Control System using RFID and Face Verification.*

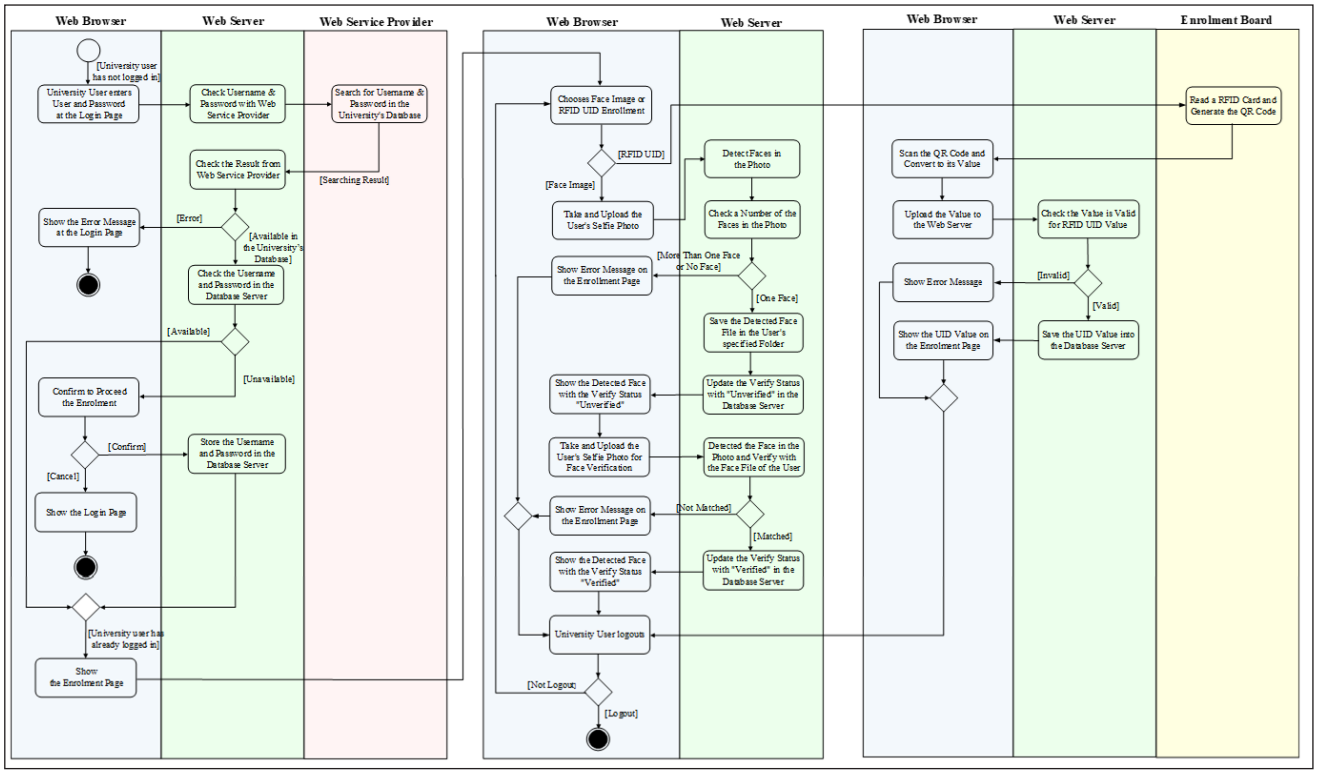

Figure 3. *The Activity Diagram of the Enrolment Process.*

intended for exchanging XML-based information between different applications with different platforms [25]. In the second constraint, the university users must enroll themselves by using their university accounts (username and password) in order to use the access control system. In other words, there is no any intervention of officials or librarians in the enrolment process of the university users. In the third constraint, the data transmission of the access control system over the local area network of the university must be secure in term of both confidentiality and integrity. In the last constraint, university and public users who has just passed the tripod turnstile gate can be monitored by the librarians or the authorized officials via web browsers.

The system architecture of an access control system was

proposed as depicted in Figure 2. It consists of two main parts: enrolment and access control parts. In the enrolment part, the university users need to enroll themselves into the database on their own in order to obtain the following user information: (1) basic user information from the university's information system via SOAP web service, (2) a face image of each university user from the webcam of the user's computer device e.g., mobile phone, (3) the UID data of a RFID card from an electronic device that converts the UID data into a form of QR (Quick Response) code. In case of public users, the enrolment process must be done by librarians. In the access control part, the university and public users can access the library at the tripod turnstile gate if they authenticate themselves with their own RFID cards and faces successfully.

3.1 User Information from SOAP Web Service

In the first process of the enrolment part as depicted in the left side of Figure 3, A university user starts to enroll himself/herself by making use of his/her own university account, i.e., username and password, to log in at the login page of our web application. The university user is then prompted with a dialog box to confirm his/her enrolment decision. If the university user makes his/her decision to proceed the enrolment process, the backend process at the web server that performs a web service client using the SOAP protocol authenticates his/her login account with the web service provider of the university information system. In this way, the web server can retrieve the basic user information, i.e., full name, university identification number (user ID), and user position (either student or staff). If the authentication result is valid, the basic user information will be returned to the enrolment page of the web application. Otherwise, an error message will be shown on the login page of the web application.

3.2 Face Image and QR Code-based UID

In the second process of the enrolment part as described in the middle of Figure 3, a university user who has already logged in at the login page of the web application can take his/her face photo or make selfie at the enrolment page of the web application by using a camera e.g., the front camera of a mobile phone. The enrolment page sends the selfie photo to the web server to detect and crop only his/her face part by using a face detection algorithm in which the HOG (Histogram of Oriented Gradients) function in the Dlib library is employed [26]. HOG that is one of the widely used face detection model is a feature descriptor that counts the occurrences of gradient orientation in localized portions of an image. With HOG, the image is broken down into small regions. The gradient and orientation of each region are then calculated. Histograms are generated for each region individually to form the structure of an object [27], [28]. If the face part can be detected successfully, the face image will be cropped and saved into a file, and then stored in the web server under a specified folder, named with his/her username. The web server will also return the cropped face image to the enrolment page. Only one face in the photo is valid, otherwise an error message is informed to be invalid with more than one face. Once the cropped face image has returned at the enrolment page, the face verification status of that face image is stored in the database and displayed as "unverified" on the enrolment page.

After the previously mentioned face enrolment process, the university user must verify his/her face image with the face image from the front camera of his/her mobile phone at the enrolment page in order to change the face verification status to be the "verified" status. This ensures that the face verification algorithm can produce the same correct result at the tripod turnstile gate. In the third process of the enrolment as depicted in the right side of Figure 3, the enrolment page needs to obtain the UID value of a RFID card by a dedicated electronic board called as *enrolment board* that reads the UID value from a RFID card, and then converts it into the QR code image on an LCD (Liquid Crystal Display) display. Again, at the enrolment page, the university user makes use of the camera on his/her mobile phones to read and translate the QR code image to the UID value. The enrolment part is completed on the condition that the three above processes of the enrolment have been done successfully. Once the enrolment part has completed, the university user receives a permission to access into the library at the tripod turnstile gate.

3.3 Access Control with RFID and Face Verification

In the access control part, it is mainly composed of following components: RFID reader board, main controller, peripheral board, web server, database server, MQTT (Message Queue Telemetry Transport) broker as depicted in Figure 2. To describe the mechanism of the access control part, a user who requires to access into the library at the tripod turnstile gate needs to place his/her RFID card over the RFID reader board. If the RFID card is scanned successfully, the RFID reader board will obtain its UID data and notify two things to the user, i.e., the buzzer will sound shortly, and the light emitting diode (LED1) on the RFID reader board will blink once. The microcontroller in the RFID reader board hands off the UID data to the main controller via a USB port by using RS-232 communication. Then, the main controller that is responsible for controlling the tripod turnstile gate encode the UID data by means of JWT (JSON Web Token) [29] for the sake of data integrity. The main controller and the web application on the web server use the same secret key of JWT for encoding and decoding. Note that the UID data must be based on the JavaScript Object Notation (JSON) format. The identification value of the main controller is uniquely and randomly defined in a set of 32-byte long characters, called as main controller identifier (shortly called as main controller ID). The main controller ID and encoded UID data are transmitted to the web server over the university's local area network by means of RESTful web service. The web service provider on the web server provides functions via web service endpoints. With regard to data confidentiality, the data communication between the main controller and web server is based on the HTTPS (Hypertext Transfer Protocol Secure) protocol.

Once the web application on the web server has received the transmitted data from the main controller, it takes the main controller ID as a device identifier to look up the corresponding secret key in the database. With the secret key, the web application uses it as a symmetric key to decode the encoded UID data. If the decoding process is done successfully, the web application will use the UID data to retrieve the corresponding username in the database. The username is then used to search for the face image file under the corresponding folder in the web server. If the face image file is found, and the *face_locations* and *face_encodings* functions in the *face_recognition* module built using the Dlib library will be utilized to convert the face locations from the face image file into the face image encoding data that is later brought into the face verification process at the main controller. The web application returns the information in a form of JSON to the main controller over the university's local area network. The information consists of the university ID number, the UID data, and the face image encoding data of the corresponding UID data. There are three fail scenarios that the main controller cannot receive the face image encoding data as follows. In the first scenario, the information from the web application does not arrive at the main controller within an expected time interval that are configured as three seconds. In this scenario, the main controller concludes that the web server is down regardless of its causes: network or web server problems. The second scenario is that the UID data cannot be found in the database. The web application responses the main controller with the *No UID* result in JSON format. For the last scenario, the enrolment of the user has not been completed yet, meaning that the enrolment status of the user in the database shows the zero value, otherwise it is one. This might be caused by the nonexistence of his/her face image file in the web server. In this scenario, the web application replies the main controller with the *Incomplete Enrolment* result in JSON format.

Once the user data together with the face image encoding data has arrived at the main controller, the timer of the face verification process, called as verification timer, starts working, and then the face verification process begins to detect the user's face image from the webcam as shown in Figure 2. In the Dlib library, two face detection functions, i.e., HOG and CNN are given [28]. The HOG function was selected in our system architecture because it runs much faster than the other one but a bit less accuracy [26], [28]. In addition, our proposed system architecture adopts face verification

instead of face identification. In the process of face verification, the encoding data of the detected face image is compared with only the face image of an enrolled user. On the contrary, in the face identification process, the detected face image is checked against the face images of all enrolled users, i.e., it would take very much longer time to be accomplished.

At the main controller, as soon as the user's face image has been detected, the face image encoding data obtained from the web server is taken to do a matching process with the detected face image. The face verification makes use of the *compare_faces* and *face_distance* functions in the *face_recognition* module, which is built using the Dlib library, to perform a face matching. As a result of the face verification process, if both face images are matched, the main controller will stop the verification timer, and then activate the push-pull solenoid by changing the *normally open* state of the relay's contact into the close state in order to unlock the tripod turnstile gate. The push-pull solenoid remains unlocked until a specified time called as *open gate timeout* has elapsed. However, once the user has passed through the tripod turnstile gate within the open gate timeout, its mechanical part will be moved and come near the proximity sensor. As a result, the proximity sensor changes an output voltage level at its output pin that connects to the main controller, indicating that the user has already passed the tripod turnstile gate. The main controller then deactivates the push-pull solenoid so that the tripod turnstile gate returns to be locked again. In addition, a librarian can also make and keep the tripod turnstile gate unlocked, as long as the librarian twists and hold the key at the tripod turnstile gate's lock. In this regard, twisting the key makes the *normally open* contact of the limit switch become closed. In such that, the main controller detects an input voltage level changed from the limit switch, and then commands to unlock the tripod turnstile gate.

3.4 Controlling and Configuration Setting

In our system architecture, rebooting the main controller and opening the tripod turnstile gate as well as configuring the access control modes of the main controllers can be done via the monitoring page of the web application. There are three access control modes: (1) RFID and face verification, (2) Only RFID, and (3) IR sensor. In the first mode, the main controller will unlock the tripod turnstile gate if both RFID card and face are verified to be valid. In the second mode, the tripod turnstile gate will be unlocked by using only RFID card. With this mode, the processing speed is increased in comparison with the first mode due to no face verification process. It is useful when many enrolled users would like to access into the library simultaneously. In the last mode, the tripod turnstile gate will be unlocked if any object or hand comes close to the IR sensor. This mode will be meaningful if users who need to pass through the tripod turnstile gate do not have their own RFID cards with them.

To control and configure the main controller remotely, a web browser sends the command data or access control mode together with the main controller ID via an AJAX (Asynchronous JavaScript and XML) call. The web application on the web server receives the main controller ID, and then uses it to search for the corresponding secret key in the database. The main controller ID is a unique number for IoT device identification. In the database, the IoTNumberID field as shown in Table 1 is used to store the main controller ID. The web application forwards the command data or access control mode from the web browser to the main controller via the MQTT broker. Note that MQTT is a lightweight publish-subscribe protocol which was designed for communications between devices with resource constraints and limited network bandwidth, such as Internet of Things [30]. The web application publishes messages to the MQTT broker for the main controller with the MQTT topic, i.e., *psutrang/library/gate/iot001*. The main controller must subscribe at the MQTT broker in order to be able to receive all messages transmitted with that MQTT topic. With regard to configuring the access control mode, it must be updated into the database server before forwarding it to the main controller via the MQTT broker.

In a viewpoint of data integrity, by using JWT, data that is

sent between the web application and the main controller via the MQTT broker as illustrated in Table 2 must be encrypted with a shared secret key for the signature part. The 32-byte shared secret key of JWT, later called as token key, is generated by the *token_urlsafe (32)* method in the secret module of the Python library. The token key of the main controller is unique and stored in the TokenKey field of the tb_iotconfig table as shown in Table 1. Note that PyJWT is a Python library which allows to encode and decode JSON Web Tokens. In addition, the MQTT broker is configured to require client authentication using a valid username and password before a connection is permitted.

Table 1. *The tb_iotconfig Table of the Database.*

Field Name	Description
id	The private key of the table
IoTNumberID	The main controller ID (001)
IoTName	Entry Gate of PSU Trang Library
Operation	Access control mode (0/1/2)
TokenKey	The secret key of JWT (32 bytes)
MqttTopic	psutrang/library/gate/iot001
Description	IoT device for the entry gate

Table 2. *Command and Configuration Data.*

Command or Configuration	Data in JSON Format
Reboot	{"IoTCommand": "REBOOT"}
Open Gate	{"OpenGate": "ON"}
Access Control Mode	{"OperationMode": 0}

At the main controller, messages received from the web server are decoded by means of JWT. Note that the token key of the main controller is embedded in its programming code. The main controller then interprets the decoded messages. If the REBOOT is found, the main controller will restart itself. In the case of the OpenGate, the main controller will change a logic level to activate a relay that is connected with the push-pull solenoid on the peripheral board in order to unlock the tripod turnstile gate. For the OperationMode, the access control mode's value at the main controller will be updated. After that, the main controller sends the access control mode's value to the RFID reader board in order to switch on one of the LED (Light Emitting Diode) indicators (i.e., LED2, LED3, LED4) that corresponds to the updated access control mode.

3.5 Monitoring

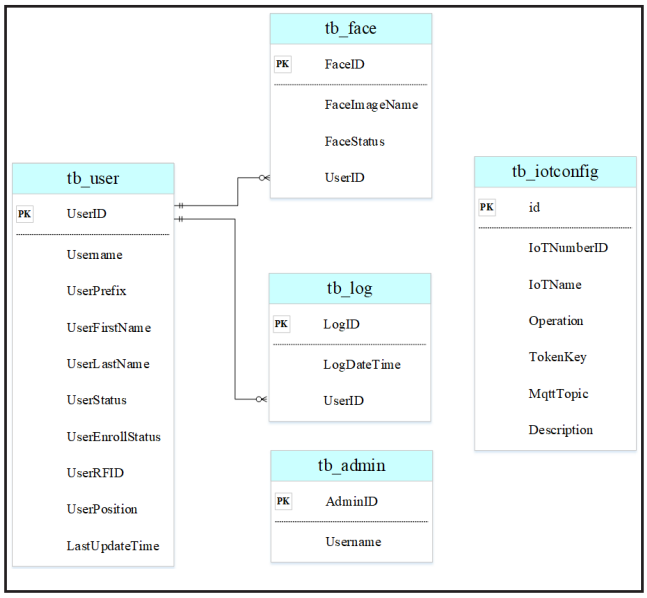

Figure 4. *ER Diagram of the Access Control System.*

In our proposed system architecture, two operations of the main controller must be monitored by librarians or authorized officers via a web browser. The first operation monitoring is to show the connection status between the web browser and the main controller as well as the access control mode of the main controller. The main controller periodically sends the following data with a time interval of 30 seconds: {"IoTNumberID": "001", "data": encodedPayload}. The encodedPayload data is derived from the raw data encoded by means of JWT. The following raw data is the access control mode in JSON format: {"OperationMode": operation Mode data}. The operationMode data is an integer number that represents the access control mode's value. Note that the main controller makes use of the *requests.post()* method of the Python library to send the data to the web server.

Once the web application at the web server has received the data, the main controller ID (i.e., the IoTNumberID data) in the data is used to search for the token key in the database. By means of JWT, the received data is decoded and validated

by using the token key. As soon as the decoding process has been successful, the operation field's value in the tb_iotconfig table of the database is updated with the access control mode's value. Note that the ER (Entity Relationship) diagram of the database is shown as Figure 4. The web application then sends the access control mode's value to the monitoring pages of web browsers in a broadcast manner via the Socket.IO library [31]. In the web browsers, the *setInterval* function of JavaScript with a time interval of 60 seconds is employed to check the connection status. If the access control mode's value from the main controller arrives at the web browser via the web server within such a time interval, the setInterval function will be reset, and the *Connected* status together with the access control mode's value will be shown on the monitoring page. Otherwise, the *Unconnected* status will be displayed.

The second operation monitoring is to store the recent users, who have passed through the tripod turnstile gate with either the RFID and face verification mode or the only RFID mode, in the database, and then display them on web browsers. However, with the IR sensor mode, users who have accessed through the tripod turnstile gate cannot be monitored via web browsers. Once a user has passed the tripod turnstile gate using the RFID card with or without face verification, the main controller will send an access event log to the web server for storing it in the database. The access event log that is based on JSON format consists of the main controller ID and encoded payload data that is encoded by means of JWT. The web application on the web server uses the main controller ID to retrieve the corresponding token key from the database. The encoded payload data is decoded and validated by means of JWT with the corresponding token key. A record consisting of the user ID from the access event log along with the current date and time reading from the web server is inserted into the tb_log table of the database. At the end, the web application forwards the access event log together with the user details, e.g., full name and user position to the monitoring pages at web browsers in a broadcast manner via the Socket.IO library.

4. Implementation

The purposed system architecture was implemented in both access control and enrolment parts. In the first part, the implementation of the main controller board and RFID reader board are shown in Figure 5 and Figure 6, respectively. Regarding the main controller board, the Raspberry Pi 4 Model B, shortly called Raspberry Pi, was adopted as the main controller. This Raspberry Pi board has the following main components: 1.5GHz 64-bit quad-core ARM Cortex-A72 CPU, RAM 8GB, 40-pin GPIO header, 2.4GHz and 5.0GHz IEEE 802.11ac wireless, 1xGigabit Ethernet port, 2xUSB 3.0 ports, 2xUSB 2.0 ports, and 2xmicro-HDMI ports. The Raspberry Pi board receives a video stream via a USB port from the C270 HD Logitech webcam with a frame rate of 30 fps. Python was chosen as a programming language on the Raspberry Pi board. The application developed on the Raspberry Pi board was set to automatically run every time after the operating system, called Raspberry Pi OS, has been booted. The programming code for the application was written on the Thonny Python IDE program. The application running on the Raspberry Pi board continuously takes the video frames from the webcam via the USB port and shows on the monitor via the micro-HDMI port. The GPIO pins of the Raspberry Pi board for the main controller board were defined to connect peripheral components as depicted in Figure 7.

The ESP32 DevKit V1 module [32] was selected as the microcontroller of the RFID reader board. The Arduino programming style was used to develop the application on the ESP32 DevKit V1 module by using the PlatformIO program as an Integrated Development Environment (IDE) program. The Raspberry Pi board communicates with the microcontroller by using RS-232 communication via USB ports. The MFRC522 module that operates at a frequency of 13.56MHz [9] was adopted into both the RFID reader board and the enrolment board. The GPIO pins of the ESP32 DevKit V1 module for the RFID reader board were defined to connect peripheral components as depicted in Figure 8.

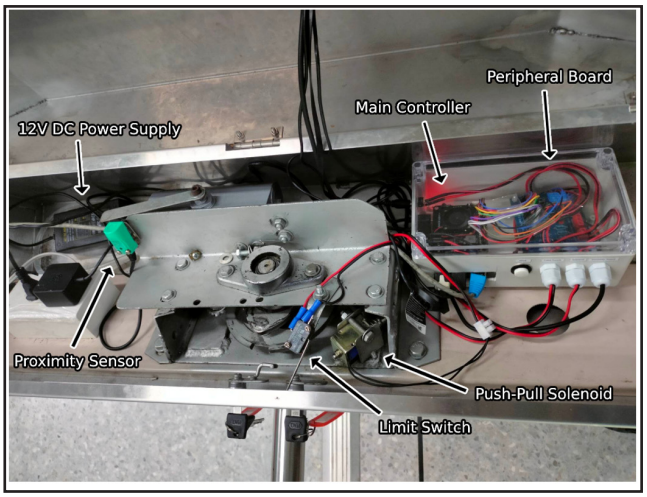

Figure 5. *Inside View of the Tripod Turnstile Gate.*

Figure 6. *Outside View of the Tripod Turnstile Gate.*

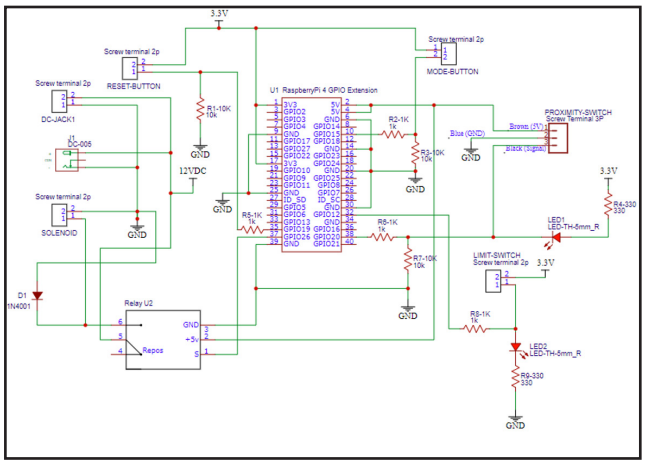

Figure 7. *The Schematic of the Main Controller Board.*

The hardware of the enrolment part as shown in Figure 9 can be described as follows. The ESP32 DevKit V1 module was employed as the microcontroller of the enrolment board. The 1.8-inch 128x160 TFT LCD module with the ST7735 driver [33] was chosen to be an LCD module for displaying a QR Code. The RFID reader module of the enrolment board also adopted the MFRC522 module. The application on the ESP32 DevKit V1 module was developed on the PlatformIO program as an IDE program by using the Arduino programming style. The GPIO pins of the ESP32 DevKit V1 module for the enrolment board were defined to connect peripheral components as shown in Figure 10.

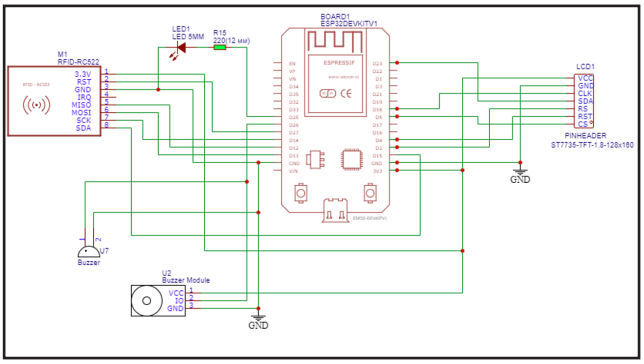

Figure 8. *The Schematic of the RFID Reader Board.*

Figure 9. *The Hardware of the Enrolment Part.*

The web server, MQTT broker, and database server were installed on the Ubuntu 20.04.4 LTS operating system and located in the same virtual machine with the same domain name (i.e., libaccess.trang.psu.ac.th) but different port numbers. The virtual machine is mainly composed of the 8-core CPU, RAM 8 GB, and HDD 50GB. The software development of the web application on the web server is based on Flask that is a micro web framework written in Python. The web server adopted Nginx as a reverse proxy while the eventlet library was applied for a WSGI server. The database server and MQTT broker were implemented by using free and open-source software, i.e., MariaDB and Eclipse Mosquitto, respectively. The MQTT broker was configured for client authentication

using a set of username and password. For the web application development, the status "Connected: (RFID and Face Verification)" appears on the monitoring page as shown in Figure 11, meaning that the main controller board at the tripod turnstile gate can communicate and send the access control mode to the monitoring page. Otherwise, the monitoring page displays the status "Unconnected!" instead.

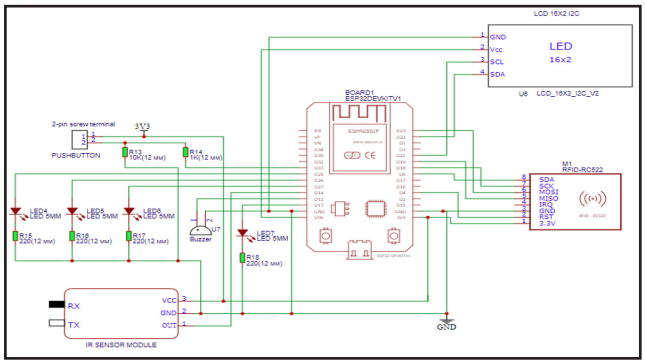

Figure 10. *The Schematic of the Enrolment Board.*

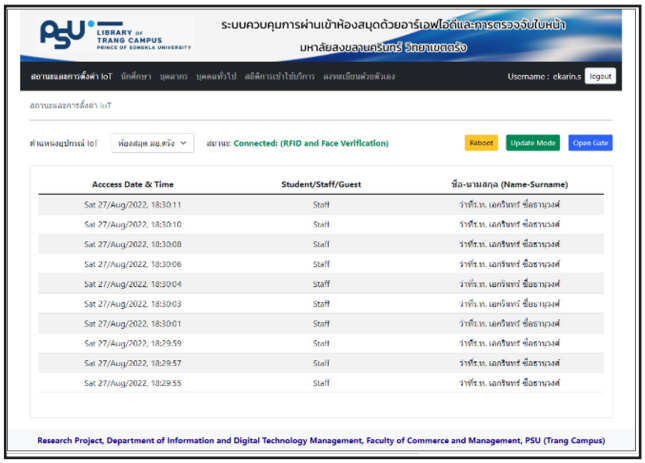

Figure 11. *The Monitoring Page of the Web Application.*

In the Figure 12, the enrolment's status shows the word "Completed!" at the top right corner. It denotes that the university user that owns this login account already provided his/her UID value and face to the access control system. With this status, the user can use the user's RFID card to pass the tripod turnstile gate. Otherwise, the enrolment's status shows the word "Incompleted!", instead. In addition, we also implemented the history page with a filtering ability for showing the previous access events at the tripod turnstile gate. For guest users who are neither students nor staffs in the university, librarians at the university use the guest page to enroll for the guest users because they do not have their own university accounts to login the web application for self-enrolment.

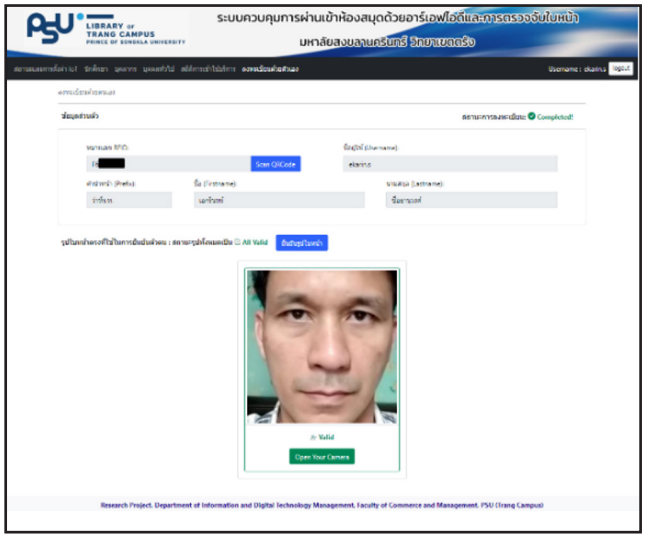

Figure 12. *The Enrolment Page of the Web Application.*

5. Performance

The performance of the implemented system was evaluated in terms of verification accuracy and response time. According to [15], the accuracy of the deep learning face recognition model in the Dlib library was measured by using the LFW dataset to be 99.38% with a distance threshold of 0.6. In our work, we did not repeat the well-done accuracy measurement for the deep learning face recognition model. However, the verification accuracy of the implemented system was measured in order to see the achievable verification accuracy based on our face images and conditions. The tolerance value of the *compare_faces* function in the *face_recognition* module built using the Dlib library was set to a lower value (0.45) than the default value (0.6) for better face similarity comparison. The timeout parameters shown in Table 3 were configured in the main controller for the performance evaluation. The OPENGATE_TIMEOUT parameter denotes an open gate timeout that is a maximum time interval for unlocking the tripod turnstile gate since the Raspberry Pi board started to activate the push-pull solenoid. The FACERECOG_TIMEOUT parameter is maximum time interval of the verification timer for the face verification process.

In the experiment setups, the number of users were set to be 20.

Each user tested the verification process of the implemented system with the user's RFID card and all the other RFID cards. This means that each user performed 20 times the verification process in a certain distance between the webcam and user. The distances were varied with 50, 75, 100, 125, 150 cm. The light intensity levels measured by the lux meter (UNI-T UT383 Mini Light Meter) at the user faces in the distances of 50, 75, 100, 125, 150 cm. were 310, 301, 290, 283, 271 lx, respectively. For evaluating the verification accuracy, the following values: GAR, FRR, FAR, and ERR [34] were calculated in percentage from the experimental results and shown in Table 4. Note that GAR and ERR are calculated from 1 – FRR, and an average of FRR and FAR, respectively.

Table 3. *Timeout Parameters in the Main Controller.*

Timeout Parameter	Time (seconds)
OPENGATE_TIMEOUT	5
FACERECOG_TIMEOUT	0.3

Table 4. *Verification Accuracies in Various Distances.*

Metrics (%)	Distances (cm.)				
	50	**75**	**100**	**125**	**150**
GAR	100	100	100	**N**	**N**
FRR	0	0	0	**N**	**N**
FAR	0	0	0	**N**	**N**
ERR	0	0	0	**N**	**N**
N denotes that no face can be found in face detection					

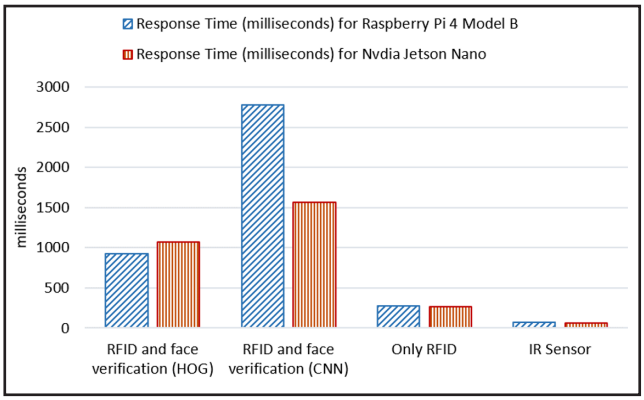

Figure 13. *Response Times of Access Control Process.*

With regard to the response time, the elapsed times of the access control process in the main controller board were calculated by subtracting the start time from the end time. In the access control process, the start time is a point in time that the main controller receives either UID or command data from the RFID reader board via RS-232 communication, whereas the end time is a point in time that the main controller starts to activate the push-pull solenoid for unlocking the tripod turnstile gate. These start and end times were obtained by the perf_counter() function of the time module in the Python library. The perf_counter() function's execution returns a float value of time in seconds. The response times of the access control process for all access control modes were such elapsed times measured and calculated accordingly. Each response time as shown in Figure 13 were averaged from 20 response time values under the same experiment setup.

In addition, the implemented system with the other main controller (i.e., NVIDIA Jetson Nano 4GB B01 Dev Kit) was tested for comparison purpose. The CNN function of the Dlib library was also brought into comparison between both main controllers in the access control process with the RFID and face verification mode. As shown in Figure 13, both main controllers provided almost the same response times of the access control process in all access control modes except for the CNN function. With the CNN function, the NVIDIA Jetson Nano board (1.566 seconds) resulted in almost two times faster response time than the Raspberry Pi board (2.783 seconds). Note that the round-trip time (RTT) delay between the main controller and web server was 0.723 milliseconds. It was measured by using the ping command and averaged from 20 ICMP request packets. With the IR sensor mode, both Raspberry Pi and NVIDIA Jetson Nano boards provided the fastest response times in comparison with other access control modes, i.e., 0.6794, and 0.5965 seconds, respectively. In the Only RFID mode, the response times of the access control process in both Raspberry Pi and NVIDIA Jetson Nano board showed about four times faster than using the RFID and face verification mode with the HOG function. Furthermore, we also compare

Table 5. *Comparisons between our Proposed System and Other Existing Attendance/Access Control Systems.*

Systems	Biometric Processor	Techniques	Database	Biometric Recognition Type	Response Time (Seconds)	Data Security
Our System	Server and Raspberry Pi 4 Model B	RFID + Face	Online	Verification	0.923	HTTPS, JWT
Leyu et al. [15]	Raspberry Pi 4 Model B	RFID + Face + Fingerprint	Offline[1]	Verification	≈ 3	AES Encryption
Akbar et al. [20]	Laptop	RFID + Face	Offline	Identification	N/A	N/A
Sanath et al. [21]	Google Colab[3]	RFID + Face	Online[2]	Identification	N/A	N/A
Kanna et al. [22]	Amazon Rekognition[4]	RFID + Face	Online[2]	Identification	N/A	N/A
Kariapper [23]	Desktop Computer	RFID + Face	Offline	Identification	N/A	N/A
Wahyudono and Ogi [24]	Raspberry Pi 3 Model B	RFID + Face	Offline	Identification	6.4185	N/A

[1]Biometric data of each user is stored in his/her own RFID card [2]Firebase database for cloud storage

[3]Cloud-based service that allows the execution of Python code [4]AWS cloud-based software as a computer vision platform

our access control system using the HOG function with other existing similar systems mentioned in the section of the related works as shown in Table 5.

6. Conclusion

In this paper, we proposed the system architecture of an access control system using RFID and face verification with QR code-based enrollment. The implementation of the proposed system architecture for the library at the Prince of Songkla University (Trang Campus) was described in detail. We also measured and calculated the performance of the implemented system in terms of verification accuracy and response time. Based on our experiment setups, the implemented system with the Raspberry Pi 4 Model B board can achieve the GAR value of 100% within a distance of 100 cm. The faces cannot be detected if the distance is over 100 cm. However, the implemented system is typically used within a distance of 75 cm. Moreover, the response time of the access control process for the implemented system with the Raspberry Pi Mode 4 board using the HOG functions of the Dlib library for the RFID and face verification mode does not go beyond 1 second. This is acceptable for us to apply for the access control system at the library in the university. Nevertheless, our proposed system cannot prevent face spoofing by using a photo, video, or mask for an authorized person's face. In the future work, we will investigate and combine the optimal technique of face spoofing detection, called as face anti-spoofing [35], into our proposed system.

7. Acknowledgement

This research was supported by Prince of Songkla University (Trang Campus) under the grant number: CAM6403072S

8. References

[1] Vandana and N. Kaur. "A Study of Biometric Identification and Verification System." *Proceedings*

of 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), India, pp. 60-64, 2021.

[2] K. Neeraja, P. R. Chandra Rao, S. Maloji, and M. A. Hussain. "Implementation of Security System for Bank using OpenCV and RFID." *International Journal of Engineering & Technology (UAE)*, Vol. 7, No. 2, pp. 187-192, 2018.

[3] A. D. Deshmukh, M. G. Nakrani, D. L. Bhuyar, and U. B. Shinde. "Face Recognition Using OpenCV Based On IoT for Smart Door." *Proceedings of the International Conference on Sustainable Computing in Science, Technology & Management (SUSCOM-2019),* India, pp. 1066-1073, 2019.

[4] M. Boroš and F. Lenko. "Importance of electronic access control systems and the need for their testing." *Proceedings of 2020 International Conference on Diagnostics in Electrical Engineering (Diagnostika),* Czech Republic, pp. 1-4, 2020.

[5] O. Kainz, J. Drozd, M. Michalko, and F. Jakab. "RASPBERRY PI-BASED ACCESS CONTROL USING FACE RECOGNITION." *Acta Electrotechnica et Informatica,* Vol. 19, No. 4, pp. 15-20, 2019.

[6] Y. Wu and Y. Mao. "Research Progress of Biometric Identification Technology Applied to Access Control System." *Proceedings of 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE),* China, pp. 279-282, 2020.

[7] R. K. Uskenbayeva, A. A. Kuandykov, A. A. Kuatbayeva, A. B. Kassymova, G. K. Kuatbayeva, B. K. Zhussipbek, and Zh. Khamzina. "Development biometric IoT access control system for employees at the example of KazPost branch." *Proceedings of 2021 IEEE 23rd Conference on Business Informatics (CBI),* Italy, pp. 202-206, 2021.

[8] K. Selvaraj, S. Alagarsamy, and M. Dhilipkumar. "Raspberry Pi based Automatic Door Control System." *Proceedings of 2021 3rd International Conference on Signal Processing and Communication (ICPSC),* India, pp. 652-656, 2021.

[9] U. Farooq, M. U. Hasan, M. Amar, A. Hanif, and M. U. Asad. "RFID Based Security and Access Control System." *IACSIT International Journal of Engineering and Technology,* Vol. 6, No. 4, pp. 309-314, 2014.

[10] C. Busch, A. Czajka, F. Deravi, P. Drozdowski, M. Gomez-Barrero, G. Hasse, O. Henniger, E. Kindt, J. Kolberg, A. Nouak, K. Raja, R. Ramachandra, C. Rathgeb, J. Salomon, and R. Veldhuis. "A response to the European Data Protection Supervisor 'Misunderstandings in Biometrics' by the European Association for Biometrics." *IET biometrics,* Vol. 11, No. 1, pp. 79-86, 2021.

[11] K. W. Bowyer and P. J. Flynn. "Biometric identification of identical twins: A survey." *Proceedings of 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS),* USA, pp. 1-8, 2016.

[12] M. R. Faraji and X. Qi. "Face recognition under varying illuminations with multi-scale gradient maximum response." *Neurocomputing,* Vol. 38, pp. 87-100, 2018.

[13] X. Yin, Y. Zhu, and J. Hu. "A Survey on 2D and 3D Contactless Fingerprint Biometrics: A Taxonomy, Review, and Future Directions." *IEEE Open Journal of the Computer Society,* Vol. 2, pp. 370-381, 2021.

[14] F. Al-alem, M. A. Alsmirat, and M. Al-Ayyoub. "On the road to the Internet of Biometric Things: A survey of fingerprint acquisition technologies and fingerprint databases." *Proceedings of the 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA),* Morocco, pp. 1-6, 2016.

[15] Z. Leyu, Z. Xinyou, F. Yunjia, L. Shuyao, B. Jun, and H. Xijia. "Design and Implementation of RFID Access Control System Based on Multiple Biometric

Features." *Proceedings of the 2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP),* Chengdu, China, pp. 570-575, 2021.

[16] M. Abdul-Al, G. K. Kyeremeh, N. O. Parchin, R. A. Abd-Alhameed, R. Qahwaji, and J. Rodriguez. "Performance of Multimodal Biometric Systems Using Face and Fingerprints (Short Survey)." *Proceedings of the 2021 IEEE 26th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD),* Portugal, pp. 1-6, 2021.

[17] M. B. Patel, S. M. Parikh, and A. R. Patel. "An improved approach in fingerprint recognition algorithm." *Smart Computational Strategies: Theoretical and Practical Aspects,* Springer, Singapore, pp. 135-151, 2019.

[18] A. A. Süzen, B. Duman, and B. Şen. "Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN." *Proceedings of the 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA),* Turkey, pp. 1-5, 2020.

[19] T. Lindner, D. Wyrwał, M. Białek, and P. Nowak. "Face recognition system based on a single-board computer." *Proceedings of the 2020 International Conference Mechatronic Systems and Materials (MSM),* Poland, pp. 1-6, 2020.

[20] M. S. Akbar, P. Sarker, A. T. Mansoor, A. M. Al Ashray, and J. Uddin, "Face Recognition and RFID Verified Attendance System," *Proceedings of the 2018 International Conference on Computing, Electronics and Communications Engineering (iCCECE),* pp. 168-172, 2018.

[21] K. Sanath, M. K, M. Rajan, V. Balamurugan, and M. E. Harikumar, "RFID and Face Recognition based Smart Attendance System," *Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication (ICCMC),* pp. 492-499, 2021.

[22] P. V. Kanna, K. V. Anusuya, and P. Vaishnavi. "Smart Attendance System using Face Recognition and RFID Technology," *Proceedings of the First International Conference on Combinatorial and Optimization (ICCAP),* 2021.

[23] RKAR. Kariapper, "Attendance System using RFID, IoT and Machine Learning: A Two-Factor Verification Approach," *Journal of Advanced Research in Dynamical and Control Systems,* Vol. 12, pp. 3285-3297, 2021.

[24] B. Wahyudono and D. Ogi. "Implementation of Two Factor Authentication based on RFID and Face Recognition using LBP Algorithm on Access Control System," *Proceedings of the 2020 International Conference on ICT for Smart Society (ICISS)*, pp. 1-6, 2020.

[25] A. W. Mohamed and A. M. Zeki. "Web services SOAP optimization techniques." *Proceedings of the 2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS),* Bahrain, pp. 1-5, 2017.

[26] Suwarno and Kevin. "Analysis of Face Recognition Algorithm: Dlib and OpenCV." *International Journal of Electronics,* Vol. 4, No. 1, pp. 173-184, 2020.

[27] H. Ahamed, I. Alam, and M. M. Islam. "HOG-CNN Based Real Time Face Recognition." *Proceedings of 2018 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE),* Bangladesh, pp. 1-4, 2018.

[28] A. Jadhav, S. Lone, S. Matey, T. Madamwar, and S. Jakhete. "Survey on Face Detection Algorithms." *International Journal of Innovative Science and Research Technology,* Vol. 6, No. 2, pp. 291-297, 2021.

[29] S. Ahmed and Q. Mahmood. "An authentication based scheme for applications using JSON web token." Proceedings of the 22nd International Multitopic Conference (INMIC), Pakistan, pp. 1-6, 2019.

[30] B. Mishra and A. Kertesz, "The Use of MQTT in M2M and IoT Systems: A Survey." *IEEE Access,*

Vol. 8, pp. 201071-201086, 2020.

[31] C. K. Rajak, U. Soni, B. Biswas, and A. K. Shrivastava. "Real-time web based Timing display Application for Test Range Applications." *Proceedings of the 2nd International Conference on Range Technology (ICORT),* India, pp. 1-6, 2021.

[32] D. P. Hutabarat, R. Susanto, B. Prasetya, B. Linando, and S. M. N. Arosha. "Smart system for maintaining aquascape environment using internet of things based light and temperature controller." *International Journal of Electrical and Computer Engineering (IJECE),* Vol. 2, No. 1, pp. 896-902, 2022.

[33] K. R. Dabhade, K. S. Mulik, and H. Jerath. "IOT based Wearable Smart Health Band Assistance." *International Journal of Engineering Research & Technology (IJERT),* Vol. 9, No. 11, 2020.

[34] N. Hassan, D. A. Ramli, and S. A. Suandi. "Fusion of Face and Fingerprint for Robust Personal Verification System." *International Journal of Machine Learning and Computing,* Vol. 4, No. 4, pp. 371-375, 2014.

[35] Z. Yu, Y. Qin, X. Li, C. Zhao, Z. Lei, and G. Zhao. "Deep Learning for Face Anti-Spoofing: A Survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 45, No. 5, pp. 5609-5631, 2023.