VIETNAM NATIONAL UNIVERSITY OF HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

———— * ————

THESIS PROPOSAL

Development of IoT-Based Solution For Checking Student's Attendance

Council: Computer Engineering
Supervise: PhD. Pham Hoang Anh
Reviewer: PhD. Le Trong Nhan
—o0o—
Student 1: Nguyen Truong Thanh (1652557)
Student 2: Nguyen Anh Quan (1552310)

Ho Chi Minh City, 4/2020

Declaration

We declare that the thesis has been composed by ourselves and that the work has not be submitted for any other degree or professional qualification. We confirm that the work submitted is my own, except where work which has formed part of jointly-authored publications has been included. Our contribution and those of the other authors to this work have been explicitly indicated below. We confirm that appropriate credit has been given within this thesis where reference has been made to the work of others.

Acknowledgments

With many thanks to our supervisor PhD. Pham Hoang Anh for his guidance during this research. To our families, who always support us throughout the process. Furthermore, to our friends, for giving us many ideas to improve the attendance system. Finally, to our team and teammate, we spent all of time to complete the thesis in the last 2 semesters.

Abstract

Attendance management is a significant part for each school, university or company. There is numerous approach to check attendance, for example, manually checking participation. Besides, on the market has different systems use fingerprint technology, radio frequency identification (RFID) and face detection. Huge numbers of them utilized those innovations independently but it is not suitable for places like education institutes.

The purpose of this research is to design and develop a new system applied different technologies together and enable the Internet of Things to the system. Therefore, the system uses both face recognition and radio frequency identification (RFID) to find out a solution for checking student's attendance. The system has a MySQL database to store entire data of the system. Python programming language is used to program the system and OpenCV for face recognition.

One primary finding of the research is to distinguish students with various edges of faces in the wake of getting the RFID tag value. While getting attendance, the student should put their RFID card on the card reader, and afterward, the system recognizes the student with the tag value. The system at that point recovers student's enlistment subtleties and start contrast student's face and spared appearances of that person. On the off chance that the framework distinguishes the student, at that point it checks for the subject's time slot to guarantee that the student goes to the right class. In the unlikely chance that every of them is correct, the system marks attendance for the student and sends a notification email to the him or her, so the student can know their participation checked effectively for the class. There is a web application additionally created to enroll courses and subjects on the framework and to see attendance details.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Topic introduction

Attendance marking is a significant activity in any university or a class. Traditionally or manually marking attendance is very time consuming and troublesome undertaking when there is a huge gathering of students. Normally lecturers are not going to mark the attendance of students and students are asked to sign the signing sheet. So it is not extremely precise. One can also sign on behalf of another. Sometimes students can forge to sign the attendance. If the teacher takes attendance by reading the names of each student, this will take a long time. Traditional checking attendance takes a lot of time if the instructor wants to take attendance more than once and the accuracy is not high. Therefore, there are many problems when manually taking attendance. There are many researchers who have tried to come up with solutions but have not really found an exact solution. Some researchers use fingerprints to solve problems, but it is not practical for universities or similar places. The scanner can be hard to recognize if the fingerprint contains water, oil or dust. It is not practical to give fingerprints when there is a huge number of students. It can make long queues and waste time. Some others use RFID technology. Using only RFID is not really accurate because a student can give his or her friend a card to take attendance. In such case, the fake attendance will be marked. Another existing solution is marked attendance using face recognition. Using facial recognition is a good solution, but it is difficult to identify a person correctly. Each person's face can change over time. They may come up with different hair styles different beard styles. In such cases, the appearance of students may change. Now there is another problem: if everyone has to wear a mask, it is harder for the system to identify. Besides, the accuracy of the system will depend on the brightness of the environment. So using RFID and face recognition separately is not very accurate. If we use both of them together, then we can come up with a good accurate solution. Although previous researchers use these technologies they use them separately.

When it comes to both technologies, one problem is how to connect them together. Researchers have previously used different modules such as Arduino and Raspberry Pi to connect them into a main operator. But all have major performance issues. Using an Arduino, we can not do a lot of work because it does not have a huge amount of memory. To implement technology such as image processing, we need enough memory to run the system. If a Raspberry Pi is used, there must be a way to optimize its workload. Otherwise, the main controller will not work properly. Some researchers have tried to implement it with Arduino and personal computers, but using Arduino is not mandatory for such a system. The newly developed system uses only one Raspberry Pi as the main controller. It needs some additional components to perform facial recognition and Radio Frequency Identification communication (RFID) but they will not be very expensive.

While executing a system like this, customers need to spend enormous usage costs. Existing frameworks are required to have appropriately wired systems administration and there ought to be the principle spot to keep up the framework. For instance, on the off chance that we consider a fingerprint based attendance system, they are networked into the main place and a database. It is required to have an individual with great specialized information to actualize the system. Along these lines, clients need to do some installment to those folks as well. There, when actualizing such a framework, the organization needs to spend a lot of starting usage cost. The new framework is actualized to use as a plug and play device. Clients can purchase the gadget from the shop, place it in an important spot, and simply use it. In this way, no expenses for wires and other system gadgets and no expenses for specialized individuals. Along these lines, the new framework will be a very cost-effective and exact framework for accurate system for student's attendance management.

1.2 Purpose of the topic

The purpose of the topic is find out the solution for checking student's attendance, and apply to universities, educational institutions to increase the quality of classes. This solution is based on IoT technology and it is easy to install and use. Accuracy of attendance devices is also important, convenience and time-saving for users are also of interest to our group.

1.2.1 Research problem

There is no legitimate method to check the attendance of students daily. The present checking attendance technique is doing manually in the vast majority of the organizations in Vietnam. In certain organizations, there is an individual to mark attendance and different university pass a attendance sheet for understudies, and students need to sign on that attendance sheet. This is very tedious and inclined to mistakes. When signing the attendance sheet, a few students may neglect to sign the sheet. Some others may sign on behalf of their friends for mark presence for their students. In like manner, there are numerous issues may happen when checking participation physically.

Many researchers attempted to give various solutions for is an issue via mechanizing attendance checking process yet these arrangements additionally have numerous issues. A few researchers developed attendance management systems based on fingerprint scanners. This additionally time-consuming and it is not practical for an educational institute. It might appropriate answer for a work environment for mark attendance of employees. In a university it is required to mark the attendance for each and every lecture session and each and every lab session. So universities need to get attendance of the same student a few time for each day. There can be bust or water on student's fingers. In such cases, fingerprints may not work precisely. So fingerprint based attendance marking is not practical for universities. Some others have created The system uses facial recognition. It likewise hard to recognize students when there changes in appearance. Students many develop their hair, young men may grow a beard and shave it from time to time. At that point, the presence of students may change. If the student is not very closed to the camera, again it is difficult to identify that student.

There are systems created dependent on Radio Frequency Identification yet it is not constantly exact. It will distinguish the student precisely yet the issue happens if one student brings his/her RFID tag on behalf of that friend for a mark that friend's attendance. At that point, it is absolutely impossible to keep away from such face attendance. Along these lines, by thinking about every single such issue, this project is proposing a new solution that builds using both face recognition and RFID technologies.

1.2.2 Research gap

There are various specialists made to mechanize the attendance management process in various manners yet a considerable lot of them have a few issues at that point think of certifiable situations. Also, there are a few constraints and issues in numerous ventures recorded in the table underneath.

Existing literature	**Limitations**	**Solution**
Research uses fingerprints to uniquely identify a person	Employees have to wait on queues to mark attendance	Mark attendance automatically and quickly without waiting in long queues
Research was done using facial recognition as technology to identify students	Faces will change over time and it is difficult to recognize faces if the face is too far from the camera	Load student photos with relevant ID numbers on RFID tags and update student faces on the database every 2 weeks and maintain multiple photos of the same student
Research was done using RFID as technology to identify students	Can make fake attendance	Verify faces by checking faces with facial recognition after being identified by RFID
The student will hear beep sound after recognition but a student can not plainly comprehend who's attendance checked	Students do not receive any notices after marking attendance	Send email notifications to students after marking his/her attendance

Table 1.1: Research gap table

1.2.3 Research questions

1. How to identify the most accurate way a student uniquely for a computer?
2. How to automate the attendance management process based on face recognition?
3. How to avoid marking fake attendance by students?
4. How to know a student attend a class after checking attendance?

1.2.4 Research objectives

The main objective of this research is to design and implement an Internet of Things enabled automated attendance management system. The research is proposed to integrate both face recognition and radio frequency identification in to a one system in order to have very accurate attendance management system. Here are the specific objectives.

1. Identify students using Radio Frequency Identification

   The RFID reader needs to put close to the entrance of the lecture lobby or laboratory and when a student comes with his/her RFID tag, the reader should read the tag and identify the student as the primary step of recognizable proof

2. Identify students using face recognition

   When a student distinguished by Radio Frequency Identification, the image of the student ought to prepared to check the student and camera ought to turn on and got to observe the student. Then the system will check the student in front of the camera with the picture of the student that having previous Radio Frequency Identification ID. This will decrease the number of pictures that got to check by the system and optimize the workload to be done.

3. Register courses and subjects on the system

   The courses of the current students and subjects of each and every course have to be entered to the system to check whether a student comes to the correct lecture session or lab session.

4. Mark attendance on the database

   After verifying the student by checking the student with face recognition, the device would mark the student's present on the database with the student's identification number, date, time and status. So, it is easy to recognize both absent and late-comers students.

5. Check whether the student attended to correct session

   The start and end times of each session are stored on the database. So, student session can be reviewed using the subject / course data stored on the database and the time that the student arrived at the class.

6. Send notification emails to students

   The program proposed has the potential to give students email notifications after marking the student's attendance for the lecture session. Students can know his / her attendance has been marked after receiving notification. If the student has not received the notification email in any way, the student can inform it to an authorized person.

7. Generate attendance reports

   Authorized persons can produce students attendance reports by entering the student's email and subject.

1.3 Thesis structure

During the dissertation, the thesis has performed a number of related tasks, which will be presented in the report as follows:

- Chapter 1: Introduction
  Introduce the topic and the reason for choosing the topic, find out the problems that still exist in the checking student attendance by referring to previous projects of other researchers, asking research questions and identifying the research object, the layout of the thesis.

- Chapter 2: Literature review
  Overview of a number of studies related to the topic that the group explored through three parts: approach, model used and results achieved.

- Chapter 3: Theoretical basis
  The background knowledge for research works such as Internet of things, Radio Frequency Identification, face recognition technology and knowledge about creating a web application and associated database system.

- Chapter 4: Methodology
  Presenting the approach to the problem: this chapter will go into detail on how to implement the model, including the tools used.

- Chapter 5: Evaluation and Testing
  Evaluate the results and test the effectiveness of the system

- Chapter 6: Conclusion
  Summarize the work group has done, evaluate and plan the direction that the group continues to develop in the thesis.

Chapter 2

Literature review

For any school, university and institute the attendance management system is a very important one. Manually marking attendance is very difficult, and vulnerable to error. Many researchers who have taken an interest in this field are trying to provide a good solution to this issue. Most common assist management methods include face recognition and Radio Frequency Identification (RFID).

2.1 Monitor and alert students through the fingerprint control system over the internet

The topic of the dissertation [1] is about the student attendance system using fingerprint scanning technology. The research group of Ho Chi Minh City University of Technology and Education, Nguyen Van Phuc and Diep Minh Thien conducted research titled ”Monitor and alert students through the fingerprint control system over the internet”. This topic is widely known in Vietnam after the automatic fingerprinting device applied by this university. The fingerprint recognition process is divided into two major parts: the image processing process and the fingerprint comparison process. [1, page 6]
1) Image processing process:

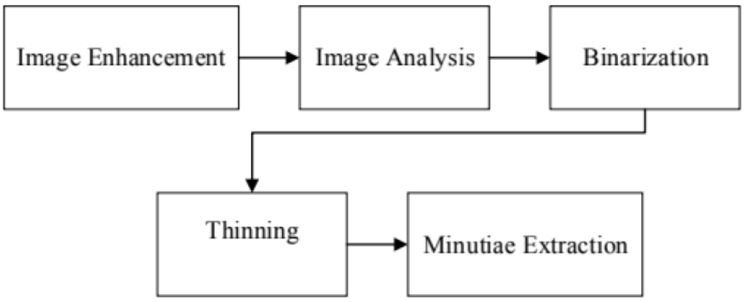

Figure 2.1: Image processing process

- Image Enhancement: Images taken from the fingerprint reader device will be clarified. Due to the fact that the fingerprint reader does not take good pictures or the fingerprint of the user during wear is worn, dirty, or the pressure of the finger while taking the fingerprint.

Therefore, this step is one of the most important steps of the process for clarifying the fingerprint image to extract the correct and complete features.

- Image Analysis: Through image analysis, the image will be removed from any interference or unnecessary information.
- Binarization: Binary of fingerprint image into black and white. This step serves for Thinning step. This step may or may not depend on the specific extraction algorithm.
- Thinning: Thin the protruding lines of the fingerprint image. This step is for extracting fingerprint characteristics. This step may or may not be due to the specific extraction algorithm.
- Minutiae Extraction: Extract features needed for fingerprint comparison. [1, page 8]

2) Fingerprint comparison process:

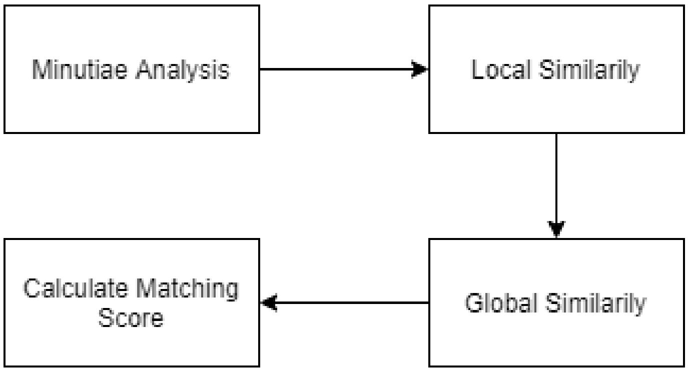

Figure 2.2: Fingerprint comparison process

- Minutiae Analysis: Analyze the necessary characteristics of the features for fingerprint comparison.
- Local Similarily: The fingerprint comparison algorithm will rely on the local information of the features (including: coordinates (x, y), the direction of the feature, the angles created by the tangent of the pattern line at the feature and the horizontal axis) of fingerprint to find the characteristic pairs of similarities between two fingerprints.
- Global Similarily: From similar regions on the local part, the algorithm will continue to expand comparisons globally.
- Calculate Matching Score: Calculate the ratio of similarities between characteristic pairs. This comparison will show how similar the two fingerprint images are. [1, page 9]

The system performs two functions: fingerprint attendance and adding new fingerprints to manage students in a classroom. Attendance to manage student access times is done by a fingerprint scan. After scanning the fingerprint, the ID data of the scanner will be sent to the internet to check, statistic of the number of times to go in, go out, the number of late visits, the number of days away, student information management, lecturers. The system also allows adding fingerprints, erasing fingerprints easily. [1, page 44]

2.2 Attendance System Using Face Recognition and Class Monitoring System

Prof. Arun Katara, Mr. Sudesh V. Kolhe, Mr. Amar P. Zilpe had a research named "Attendance System Using Face Recognition and Class Monitoring System" [2] This program automatically uses facial recognition to take the student attendance. Using continued observation, the authors suggested a method to obtain the exact attendance. Continuous monitoring enhances the efficiency of attendance estimates by easily observing expressions. This project used the visual class library of computers (OpenCV) to get face recognition. OpenCV provides freely available source codes, functions that are used by application developers to build solutions in the computer visual field. OpenCV is commonly used for the analysis and tracking of motion, face recognition, three-dimensional construction and target tracking. [2, page 273]

The system includes a face recognition system and a monitoring system in the classroom. Inside of the classroom is a camera. Then use the video frame to capture the photos of the students. The student is automatically checked and recognized by the face recognition program. The OpenCV is used to check the students' faces with faces on the student database. Face recognition process is also quicker with the correct and effective face recognition system. In face recognition, there are several algorithms that are used as facial geometry based recognition, including invariant methods and methods using machine learning. The researchers have proposed a new recognition system that gives faster detection rates and a process than others. The authors suggest this approach for applications in real time that function rapidly and robustly and said the approach would yield suitable results under various lightning conditions. After detecting faces the researchers did preprocessing. The preprocessing steps include the equalization of Histogram to extracted images. The technique of Histogram equalization makes the picture visible by enhancing the contrast of the picture, since it increases the range of intensities in an image. The researchers then involved in the development of the database. Enter details of each and every individual in this phase is compulsory because the system is a biometric based system. It is very important to store photos of every and every single person extracting biometric features from them. The researchers used a MySQL database to create database and store picture of 50 students in various angles, different expressions and different lighting conditions. [2]

The next phase of the project was the extraction and classification of features. This was a very critical step because the project's success depends heavily on this process of extraction and classification of features. The researchers have employed holistic techniques to achieve extraction of the function. The researchers used an algorithm called Principal Component Analysis (PCA) to economically represent the stored faces, and they further explained that the Principal Component Analysis algorithm used a technique called Eigen faces and some projections of that technique. The final stage was post-processing. The student details are updated to a MySQL database at this stage, after recognizing student faces. Then, the data from the present student send to and store the extreme device. Hence, such data can be seen by approved people. The authors developed a graphical user interface for easy use of the system, and after completion of recognition, an excel sheet with student attendance is generated. [2]

2.3 Attendance System Based on Face Recognition

The authors Hemavati Sabu, Madhumala Sonawane, Jayashri Patil, Pradnya Vikhar published a research paper entitled ”Attendance System Based on Face Recognition”[3]. The authors did not use a special micro-controller on this device. Authors used a webcam to capture the images, using face recognition techniques from Eigen to identify student faces. Viola and Jones are used for facial recognition algorithms. Every student's attendance is recorded to an attendance log and this record offers monthly-based attendance information and alerts the teachers if any of the students are consistently absent. [3]

Image acquisition was the initial portion of the program. The images obtained are transformed into a grayscale image and all further operations are conducted based on the grayscale images. Then the face detection step started. For face detection purposes the Viola and Jones Algorithm (also known as the Ada-Boost algorithm) was used. Researchers tested whether there is a presence of humans here. If the human faces are detected then the next step was to check by recognizing faces who they are. This step includes the extraction of features and the classification of features with another two sub steps. For extraction of the function a technique called Principal Component Analysis (PCA) is used. The identified faces are compared with the faces in the image database after performing extraction of the function and classification of the features. After the next step is recognizing faces, the process of documenting attendance begins. Following successful identification the presence of the student is listed in front of the student number in the database. Then the production process of the attendance report begins and the student information and the regular attendance details are taken from the database. Depending on the necessity this measurement occurs. Calculation of attendance day wise, student wise, and class wise can be made. The attendance reports produced are stored in a file to enable allowed individuals to use them in the future too. The program consists of two databases as data base for student information and a database for attendance. The student database is composed of a student number, the student name and the student class. Through day the attendance database contains the student's status. The authors say the system is very easy to deploy and it will take much less time to run. The system will reduce fake attendance so the system will be very effective. Around the same time, the researchers said the device will not be 100 effective because facial recognition might have error with 45 degrees of facial rotation rounding both vertical and horizontal axes. In dark and darkened conditions it's hard to achieve the best performance. Further authors said that “While detecting face parts, we might get multiple face detections in an image even though only one face exists. This face detection problem leads to false recognition and hence inaccurate attendance marking” [3]

2.4 Using Real Time Computer Vision Algorithms in Automatic Attendance

There was another research paper called “Using Real Time Computer Vision Algorithms in Automatic Attendance Management Systems” written by Visar Shehu and Agni Dika. [4] This paper presents a new way of handling the attendance using face recognition algorithms in real time. This consists of an additional tool to mark student attendance by instructors using machine learning and adaptive method to monitor students' facial changes by analyzing them over a long period of time. The program employed a procedure called continuous observation in order to observe students in a long time. This system integrates computer vision, and faces algorithms of recognition in the attendance management process. The program implements singing a digital camera in classroom according to how the camera can see the entire classroom at once. Then it can associate faces of students with faces that are stored in the database. [4]

The authors said the paper will address issues such as real-time face detection of multiple-object environments, face recognition algorithms as well as social and pedagogical issues with the applied techniques. This framework also used the Haar cascade classifier for the face detection method in OpenCV. It operates by training model that uses images of positive faces and negative faces. An image containing an object to be detected is called a positive image, and negative images are images with no objects to be detected. After a model is trained, face features which are stored in an XML file can be identified. Once a face is detected, a rectangle frame selects the area of the face, a crop the area, and then sends it to the module for image recognition. The authors said they were using a server-based module, programmed in Python that takes advantage of Eigen's faces to recognize a face. Even though the authors used this algorithm, some of the disadvantages of this algorithm are discussed. The researchers have identified that the algorithm depends on the scale, pose and color of the images being compared. [4]

To improve facial recognition the system uses another strategy. The faces will change from time to time. So when a student's face is identified with a different appearance, program can store a copy of that image in the image database to accurately recognize that student in the future. Authors store, along with the image, the time and date of that image. When comparing the student's images, comparison begins from the downward order of the date. Therefore, the program will compare the student's latest face picture. So that several images of the same person are stored in the device. The program is also capable of identifying students with different facial appearances. If faces are not identifiable in the classroom, the program will store those images so that authorized person can manually name those students and add them to the student image database. Then in next time the system will be able to identify those students. The program can only compare facets of students who enroll in the related course to speed up the process of face recognition. [4]

The authors then conducted an experiment to test the system. The system has been extended to three classes, with ten groups in all. Altogether there were 147 students enrolled for the classes. 70 percent of these students identified faces and 30 percent were recognised. When only the latest faces were tested, the success rate was around 56 percent. So it's obvious to say the face recognition rate wasn't in very good value. The authors discussed another system problem as a matter of student privacy. The students' images are stored in a server. Thus, third party people have the possibility to access those images. Authors said it would not provide a way to protect privacy because this application was done for face recognition. The authors have suggested a way to ensure privacy but this is not successful. Authors also suggested encrypting the students' images and comparing those encrypted images. But at the same time researchers said it won't be successful because more noise is contained in biometric images. [4]

2.5 Automated Attendance System using Machine Learning Approach

Some researchers tried by adding machine learning to face recognition to give a better solution. The research conducted under title “Automated Attendance System using Machine Learning Approach” by Hemantkumar Rathod, Yudhisthir Ware, Snehal Sane, Suresh Raulo, Vishal Pakhare and Imdad A. Rizvi is one example for such research [5]. This system was based entirely on MATLAB. As well as taking attendance using face recognition, the most steps of this research are the same. In this project, the researchers used eye recognition with the face recognition and also used a popular computer vision feature descriptor called histogram oriented gradients (HOG). It captures frames with students’ faces after inserting a camera inside the classroom. Theses frames further analyze to detect eyes on the faces to ensure these frames are of faces. If only eyes are detected, the frame will be accepted as a face and otherwise it will be rejected. The key method of object recognition using HOG is to feed the descriptors into a supervised learning-based reconnaissance framework. Researchers have used the Support Vector Machine classifier as a decision feature in this project. If it has trained to identify objects like humans, Support Vector Machine will make these decisions. The researchers used a ratio called signal-to-noise ratio as a comparison between the original image and the compressed image while doing model experiments. The researchers said the higher the signal-to-noise ratio means good quality of the compressed or reconstructed image. Another important thing about this model was that it has the ability to recognise many faces at one time. Thanks to this function the time taken to do the attendance marking is reduced. But the issue was that it requires high performance controller to do it the way it was designed. As a future scope, the researchers suggested implementing face recognition for Automated Teller Machines (ATM) with high-resolution cameras to avoid fraud, and further proposed attaching face recognition to voting sites to avoid duplicate votes. [5]

In this way many researchers tried to use face recognition to introduce an attendance management system. When carrying out the literature survey we can recognize that implementing an attendance management system using face recognition alone is not very effective. Meanwhile, some researchers use Radio Frequency Identification (RFID) to try to come up with a better solution. RFID provides some better results when comparing the face recognition. [5]

2.6 RFID Based Student Attendance and Monitoring system

The researchers Ashwin K, Aswin Perumal A, Krishnakumar S, Maheshwari M wrote a research paper titled “RFID Based Student Attendance and Monitoring System” [6]. This system also enabled a system via the Internet of Things because it uses RFID tags, RFID readers and the information gathered by reading RFID is connected to an existing network. This RFID system includes RFID reader, multiple RFID tags and an anti-collision protocol to prevent collisions between multiple RFID tags and to increase the identification performance. Using the adaptive binary tree and ALOHA techniques it was said the collision is normally avoided. Two major types of RFID tags exist as active RFID tags and passive RFID tags. Passive RFID tags only come from the RFID reader to deflect magnetic or electromagnetic waves. But they don’t have onboard radio frequency emitters so they can’t generate their own radio frequency signals. Active RFID tags have on-board radio frequency signal emitters and are able to generate their own radio frequency signals. But there should be a lot of energy required to produce a signal and it should have an internal power supply. There are also other types of tags from those two types of RFID tags as semi-RFID and RFID SAW tags. Semi-RFID tags contain a battery embedded to the internal supply circuitry with a tag. But the battery does not produce radio frequency signals in such a way that the tag is still passive. [6] RFID SAW (Surface Acoustic Wave) tags do not have internal or internal power circuits. They're not got internal transponders. RFID readers also classified according to whether or not they have an internal power supply. They have also listed themselves as participating or not. Active readers have their own power supply, so they can read active as well as passive tags with no power supply in their range so passive readers can read only active tags. Readers additionally graded according to the spectrum of radio frequency signals they can read. RFID readers which can read in the 120–150 kHz signal range are called low frequency readers. They are called high frequency readers if they operate at 13.56MHz as well as they can read with in the range between 10 cm to 1 m and readers operating at 433MHz are called Ultra High Frequency readers. They ranged from 1 m to 100 m. [6]

The goal of developing this system was to automatically record attendance when a student enters the classroom without showing the reader the student identification card. But to implement this proposed system, multiple readers of ultra-high radio frequency identification should be places in each classroom and therefore it is very expensive to implement. Authors therefore create a prototype system that uses a low frequency RFID reader and students need to use their ID card to mark their attendance. The main idea was to automatically record student attendance and keep attendance reporting. The system used passive tags and an active reader. This readers and tags operating frequency was 125 KHz, and their detection range was less than 10 cm. The passive tags have no power supply, and when they are within interrogator range, they emit signals. The chip of one tag becomes powered, and it starts to transmit signals. Then the reader detects changes in electromagnetic waves and according to that shift it determines the meaning of the tag. While researchers are implementing the prototype with a low-frequency reader, it was originally intended to do so with a high-frequency reader. So writers find the key issue that happens when they use UHF readers. It was prone to collisions from UHF readers when reading multiple tags at once. This can happen when more than one student is entering the classroom simultaneously. Researchers have therefore suggested a solution to this problem using a technique called the slotted adaptive binary tree and the RFID tags are singularised. According to this strategy, if a packet arrives while another packet is sending, the second packet must wait until the next slot starts and collisions will be prevented [6].

Using radio frequency detection, the device is able to detect the position of current positions of the students. It monitors the location by using the RFID reader's IP address which the tag values obtained from the user. The system sends that information to authorized parties following detection of the location. When considering how the RFID reader connected to the device, the authors stated that they were using the RS232 USB adapter cable connector to connect the reader via USB port to the device. This machine includes a database with the registration numbers for all students and their respective tag values. This database also has classroom time tables where the readers are housed. When a student enters the classroom, he/she reads his/her tag value and compares it with the registration number. At the same time it limits the involvement of the student in the subject or class. If the class and registration number of a student matched, his/her attendance marked in the database as being present. One speciality of this system was tracking the arrival of students and classroom leaves. If a student leaves the classroom before the end of the lecture session, the system marks the attendance of that student as being absent.

Using the ability to track the current location of the student, the system checks for the relevant lecture on the students who are not in the classroom. A technique called Geo Fencing is used to assess the student's geo position. Geo fencing is a technique that uses Google's positioning system or radio frequency identification to establish virtual perimeter or barrier around a physical geographic area. The tools set up on this technique will generate alerts when GPS-enabled device enters or exits from the Geo fencing area. The authors suggested at the end of this paper to expand their works to cloud as potential work. The attendance details of several departments of the same school can then be interconnected and student parents can see the attendance of their children to the classes [6].

2.7 Analysis of Student Activities Trajectory and Design of Attendance

A researcher Zhongyun Jiang did research named "Analysis of Student Activities Trajectory and Design of Attendance Management Based on Internet of Things" [7]. This work also utilized RFID to monitor student attendance. The aim was to monitor student attendance as the student moves through classroom entry. In addition to label student attendance, the device also has a feature of monitoring the activity trajectory of the student in real time. The machine consists primarily of the RFID scanner, electronic RFID tags, a wireless communication module, servers, and management system. This device used RFID tags that can be connected to the cell phones of the student and the students are expected to have the tag to access the premises of the university. The researcher splits the system as application layer, network layer, and sensing layer into three layers. Application layer contains applications for the Internet of Things, smart mobile phones and Personal Computer. Network layer consists of a database server and a network layer-to-sensing interface. This is primarily used for data exchange and communication between the internet, university wireless network and university wireless database server used by the university that stores all kinds of university data. The sensing layer has tags containing an RFID identifier, sensor network and identification cards for the user. This system's main control module is a personal computer and it also used a server that stores the system and database for student attendance management. This main control module uses USB ports, RJ45 port, and COM port to gather information from the wireless communication module and from all external devices connected to a personal computer. The main control module attendance management software was developed using Microsoft Visual C++, and the Microsoft access database was developed. The RFID-SIM card used as an RFID tag on that device. This card incorporated the existing structure of the network communication chip for RFID technology and radiofrequency identification. This SIM card uses a 2.4 G microwave band to do effective two-way communication that distances up to 500 cm. The researcher said this SIM card's communication standard conforms to the standard ISO7816 transmission protocol. The tag used for this method is therefore more improved as opposed to conventional RFID tags[6]. For this system, the researcher utilized SHRM203 as a reader module. This module is ideal for direct communications over long and middle distances. The module is a small RFID reader with low power which is used in the microwave band. The transmission of data between the reader module and the wireless communication network is achieved using radio frequency signals, and when the electronic tag reaches the reader range it reads the tag and transmits tag data. In wireless communication module, this machine utilized Zigbee coordinator for network construction. This coordinator is the center of the network node, and is responsible for the network being involved. This connects to the host via a COM port.

2.8 Design and Implementation of a smart attendance management system using multiple step authentication

There is another research done by Dhiman Kumar Sarker, Nafize Hossain and Insan Arafat Jamil on the title “Design and implementation of a smart attendance management system using multiple step authentication” [8]. Radio frequency recognition, biometric finger print reader, and base authentication for passwords are combined into one device. The language used for program creation has been C#. The system’s principal controller module was the Arduino super 2560 board. The device includes senor fingerprint, RFID reader and tags, 4x4 keypad, LCD monitor and a personal computer. This system’s method of identifying students involves several steps. As the attendance marking begins it reads the RFID tag first. If the tag value is correct, students must enter the password and show the fingerprint sensor to the finger print. Then, only the device marks present for the presence of appropriate students. The program has a windows application for teachers and other people who are allowed to use it. Using this method, they will produce attendance reports and keep an attendance log [8].

2.9 Design and Development of Portable Classroom Attendance System

There was another research done by Nur Izzati Zainal, Khairul Azami Sidek, Teddy Surya Gunawan, Hasmah Mansor, and Mira Kartiwi who are students of International Islamic University Malaysia. [9] The title of the research is “Design and Development of Portable Classroom Attendance System Based on Arduino and Fingerprint Biometric”. The device uses a monitor to indicate the date and time, and its attendance marking process starts when a figure is scanned by the figure print scanner. Before beginning the attendance marking process all the students’ fingerprints must be processed. When a figure print is identified, the program checks whether a figure print in the database matches the fingerprint. If it matches, that student’s number is displayed in the system. Simultaneously, when the finger print is checked, it encrypts the data and stores it in SD card. If finger print does not match, an error message is shown by stating that the ID does not match. A sub objective of this research is the development of a portable device. They used a memory card to store the attendance data to enforce the portability. They also consider the size of the machine to improve the portability. The researchers used Arduino microcontroller as the device and finger print scanner’s key controller, touch screen, memory card, battery, and other hardware components. This device uses a battery as its source of power. Therefore, availability is somewhat limited, because if the battery is small, the device can not work properly. [9] The microcontroller is Arduino mega which operates at 5V voltage based on ATmega1280. It includes four hardware UARTs for serial TTL (5V) communication and serial display, allowing for simple text data. This was useful for researchers to send textual data from Arduino board to the Arduino board and to obtain textual data. Researchers used USB communication to transfer attendance data to an SD card, as there is no space for attaching the SD card to Arduino. On a human finger, the finger print scanner produces pictures of ridges and valleys. This was used to identify a person, and each finger print receives a unique ID. A touch screen shield used to enhance the system's user-friendliness as the menu, date, time and user details on that screen is displayed. The user will use the touch-screen to pick the menu. Then an SD card module used to implement the interface between the memory card and the microcontroller. The external memory card used to store information about the students. In addition to these hardware modules, the fingerprint reader uses a Real Time Clock (RTC) module to get the current time, date and day. The exact time the student attends can be registered and the class leaves. Software components should be in place to code the system. So, the researchers used only an Arduino compiler as tools. Then the researchers also think about enhancing protection. So to come up with that, researchers used a technique of encryption based on Ceaser's cipher method. It includes replacing each alphabet letter with the letter which is a number of places further than that alphabet. The researchers suggested some improvements to this system as the introduction of an indicator of battery level to indicate the battery's existing capacity, security and device data backup. They suggested a wireless device that transmits the data wirelessly to a server as a backup and acts as a network for real-time attendance.[9]

2.10 IoT based Automatic Attendance Management System

A set of students named Sri Madhu B.M, Kavya Kanagotagi and Devansh who are students of Siddaganga Institute of Technology, India was done research titled "IoT based Automatic Attendance Management System". [10] These researchers have used the Radio Frequency Identification technology to identify students as the key technology. This device has three components generated in RFID part as an RFID reader, RFID tag, and a computer with a particular database. RFID allows card receiver data to be transferred around 10 meters, depending on the tag / card type used. So the researchers agreed to use RFID to identify a student as their technology. The program has been designed in a way that makes it cost efficient and user-friendly. The researchers used their main controller, the Raspberry Pi 3. They then used an official Raspberry Pi adapter, HDMI cable, LCD display, 10 GB memory memory card, RFID card and reader, Buzzer and PIR sensor. The system's entire operation was split into two phases as the process of registration and acknowledgement. The registration process gathers information of the student such as name, number of students, subjects registered to exist semester, time table for that student and so on. In this stage the relevant student details are embedded in RFID tag. The reconnaissance phase consists of recognizing students and storing details of students on the database. All students are required to bring their RFID card to mark the attendance, and tap it on the RFID reader. The reader will then recognize the card, and record the attendance in the database of the faculty. This was done using Raspberry Pi synchronized by IoT and required information appears on the LCD display. Researchers further identified the reason why they selected Raspberry Pi 3 as their principal controller in this research paper. It consists of an extended GPIO with 40 pins, 4x2 USB ports, 10/100 LAN, micro SD card slot, micro USB power. It also has a powerful CPU, on board Wi-Fi and Bluetooth, HDMI port, DSI display port, and CSI camera port, in addition to these ports and slots. But there are many things that can be achieved with these new features and technologies. That was the primary reason researchers chose Raspberry Pi 3 as their main controller. The researchers used a PIR senor to detect the motion indicating a human is entering the region of the sensor. Once a person is detected by the PIR sensor, the RFID reader will be triggered and it will only accept one card at a time before a person is detected by another PIR sensor. Hence, the database will not update until the second PIR sensor detects a person. The developers used XAMPP software to build PHP database [10]

Further the research paper explained the system's workflow steps. When a PIR sensor detects motion, the machine starts having attendance. It begins the RFID reader, and the LCD screen is initialized. Universal Asynchronous Receiver/ Transmitter begins reading RFID tags after the device initializes. Then, it sends Raspberry Pi Model scanned student IDs of the RFID card data. The next step of the research process was to scan and match the student IDs and collect relevant information from the students. Finally, the device marks student participation after matching student information with date, time and class. To see their attendance, the researchers developed an Android application for the students. To solve the problem of attendance proxy, the researchers used two passive infrared sensors. The first PIR sensor detects a person by detecting a person's body heat and give an output of 1. If the PIR output is 1 the RFID reader must read the attendance system RFID tag before the student passes the second PIR. The student must then tap his/ her RFID card at that time, and not who is not present. The student then enters the classroom, so that the second PIR sensor reads big. Once the second PIR sensor reads high, the student's attendance records according to the subjects into the database. Researchers have therefore assumed in this research that the students do not carry the RFID tag for their mate. The researchers here did not create any specific interface to produce attendance reports or interface to see the teachers attendance. The researchers simply give the teachers direct access to viewing the database. The research paper eventually discussed the possible changes that can be made to the program. There the researchers suggested attaching a web camera and detecting students using face recognition or using fingerprint scanner or Retina scanner to enhance student identification process accuracy. [10]

Likewise, many researchers have tried in various ways to provide approaches to the attendance marking. It is understandable, in considering this literature review, that systems developed using face recognition and radio frequency communication (RFID) are succeeded to some extent. There have been some problems but some changes can be made to solve them. Therefore, the new system has been built to solve most of the problems in the attendance marking process, so that this system can do the attendance marking very accurately and with good results.

Chapter 3

Theoretical basis

3.1 Internet of Things (IoT)

3.1.1 What is Internet of Things?

"The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction." [11]

Nearly all of the data available on the Internet today was first collected and man-made. The problem is that people have limited time, attention and precision, which means that they are not very good at collecting data about things in the real world, and if we had machines that knew all about things, using data that they gathered without any support from us, [12] we would be able to track and count everything and greatly reduce waste, loss and cost and this is what gave birth to "Internet of Things".

3.1.2 IoT Architecture

IoT is not just Internet-connected consumer devices. Indeed, IoT is the technology that creates systems capable of autonomously sensing and responding to real-world stimuli without human intervention. Therefore we need to develop a process flow for a definite framework on which to build an IoT solution. [12] Typically IoT Architecture consists of these 4 stages:

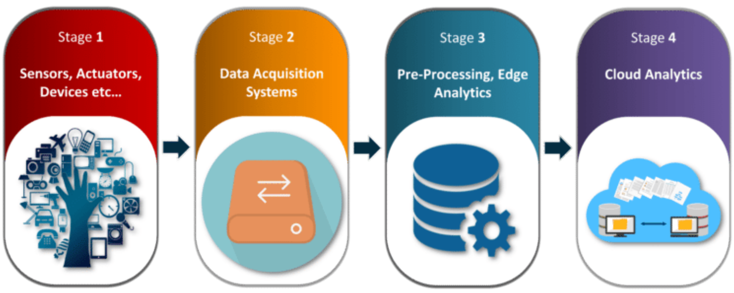

Figure 3.1: IoT Architecture

Stage 1 (Sensors/Actuators): A thing in the context of "Internet of Things", should be equipped with sensors and actuators thus giving the ability to emit, accept and process signals.

Stage 2 (Data Acquisition Systems): The data from the sensors starts in analogue form which needs to be aggregated and converted into digital streams for further processing. Data acquisition systems perform these data aggregation and conversion functions.

Stage 3 (Edge Analytics): Once IoT data has been digitized and aggregated, it may require further processing before it enters the data center, this is where Edge Analytics comes in.

Stage 4 (Cloud Analytics): Data that needs more in-depth processing gets forwarded to physical data centers or cloud-based systems.

3.1.3 Applications of IoT

The Internet of Things platform can be used to accumulate data related to a specific geographic location using remote monitoring tools and perform analytic so that early warnings of a disaster can be found out.

There are many applications of Internet of Things:

- Natural Disaster Management with IoT
- Better Health-Care with IoT
- Cloud Computing Training
- Smart Farming with IoT
- Smart Energy Management with IoT
- Pollution Control with IoT

3.1.4 Security issue of IoT

Anyone wishing to join the Internet of Things needs to establish a comprehension of what IoT security is and why it is so important. There are many reasons why security risks to the Internet of Things are so challenging. [13]

- The number of devices connected to the Internet is increasing and is expected to reach 50 billion by 2020. [14] The growth of the number of devices is too large compared to the speed of development of security solutions.
- While servers and personal computers are protected by the presence of the owner, IoT devices, especially ubiquitous sensors, are more likely to be stolen or intrusive. An attacker can find a hole and attack the IoT system.
- While servers and personal computers are protected by the presence of the owner, IoT devices, especially ubiquitous sensors, are more likely to be stolen or intrusive. An attacker can find a hole and attack the IoT system.
- IoT devices often have low battery capacity, memory and processing power. Therefore, they cannot perform security or anonymous tasks that require large memory resources.
- The number of devices increases rapidly leading to the huge amount of data collected. If this data is stolen, it can have major consequences. For example, hackers can infiltrate the data of the smart home to know the owner's living habits and conduct illegal activities such as theft.

3.1.5 Challenges to IoT

According to the study [15], the challenges for IoT include the following: security, privacy, interactivity, few standards, legal, copyright, economic and other issues. other development issues. These issues are studied [16] summarized as Table 3.1.

Problem	Challenge	Description
Security	Practice designing	Lack of resources in training future generations about security design for IoT.
	Trade off between security and cost	Lack of sound decisions about the cost benefit analysis of IoT.
	Standards and metrics	Lack of standards and criteria to determine the security of IoT devices
	Authentication and control	Lack of optimal control role in the IoT device communication model to prevent the risk of hijacking and cyber attacks.
	Maintenance and upgrade	There is not enough information on maintenance and upgrades. This is based on the expected lifespan of IoT devices in the network.
	Shared responsibility	Security in IoT can be achieved when network components are shared by multiple parties.
	Law	IoT devices and software are developed without legal supervision.
	Obsolete equipment	Restrictions on replacing old or unwanted equipment.
Privacy	Fairness in data collection and usage	Lack of strict rules against data collection and usage.
	Transparency, expression and enforcement	The lack of multi-stakeholder models allows for transparency, expression and enforcement.
	Expectations for wide security	Lack of privacy protection models for IoT and inability to realize users' privacy expectations.
	Freely design freedom	Limited resources to develop IoT devices that incorporate preset privacy principles.
	Identification	Lack of protection from data collected by IoT devices.
Interaction	Exclusive ecosystems and customer wishes	Lack of the concept of a closed ecosystem in data collection and reuse format at the user's choice. Security keys and personal protocols can be implemented.
	Financial and technical constraints	Limited technical resources and investment resources.
	Schedule risks	Possibility of outdated interactive standards.
	Financial and technical constraints	Limited technical resources and investment resources.
	Schedule risks	Possibility of outdated interactive standards.
	Financial and technical constraints	Limited technical resources and investment resources.

	Financial and technical constraints	Limited technical resources and investment resources.
	Schedule risks	Possibility of outdated interactive standards.
	Technical risks	Lack of awareness of engineering design risk protocols.
	The device is in poor condition	Lack of material standards for best design practices.
	Legal system	Standard legal systems to maintain IoT device compatibility.
	Configuration	Lack of a standard configuration for connecting large numbers of IoT devices.
IoT standard	Disseminate standard efforts	Lack of effort in developing standards and protocols.
Regulations, laws and rights	Data protection and cross-border data flow	Limited development of data sharing and reliable policies, laws and regulations.
	Distinguish data	Lack of laws on the use of IoT data in different ways.
	Legal enforcement support and community protection	Lack of laws on IoT data to use against criminals.
	Responsibility of the device	Laws against the issues of liability of IoT devices
	Disseminate equipment under legal action	Complex responsibilities during the operation of IoT equipment.
Emerging economic problems	Invest	Limited investment in IoT research and development in both developed and developing countries.
Development issue	The infrastructure	Add a burden or pressure on the internet and media infrastructure around the globe. Limited activities to strengthen internet and communication infrastructure.
	Technical and industrial development	Limited research to assess the economic and technical benefits of IoT in emerging economic countries.
	Coordinate policies and regulations	Little awareness of policy plans with the continued growth of IoT.

Table 3.1: Challenges to IoT

3.2 Radio Frequency Identification

3.2.1 Introduction

Radio Frequency Identification (RFID) has a long history and is a part of both present and past technological development. RFID makes toll payments quickly and object recognition quickly. [see 17, page 15] Furthermore, RFID offers advantages, such as tracking properties, monitoring safety conditions, and helping deter counterfeiting. RFID is an important part of techno-logical revolution. RFID stands for Radio Frequency Identification, a term describing any identification method in which an object is connected to an electronic device that uses radio frequency or magnetic field variations to communicate. An RFID system's two most talked-about components are the tag, which is the identifying device attached to the item we want to track, and the reader, which is a device that can recognize the presence of RFID tags and read the information stored on it. The reader will then notify about the existence of the tagged objects to another device. The machine the reader interacts with normally runs software that stands between applications and readers. That software is called middleware RFID. [see 18, page 21] [Figure 3.2] shows how the pieces fit together.

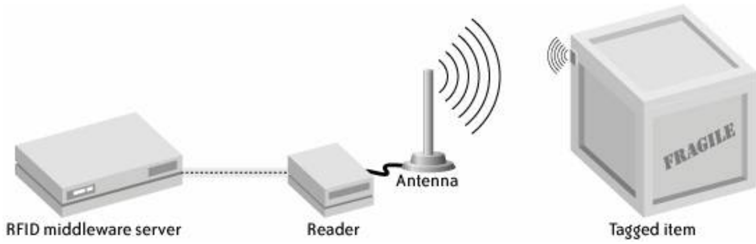

Figure 3.2: Spreadsheet of CDDiscs and days

3.2.2 RFID system components

Many manufacturers develop RFID devices, and they exist in countless variants. However, an RFID system mainly consists of three components; the middleware transponder/ tag, reader, and RFID.[19]

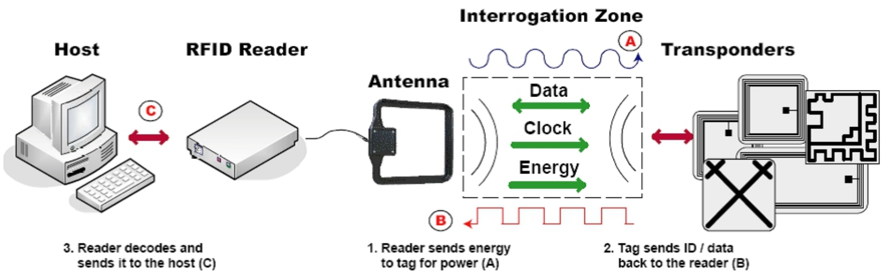

Figure 3.3: RFID system components

1) RFID transponder/tag

[19]

	Passive Tags	Semi-Passive Tags	Active Tags
On board power supply	No (From Reader)	Yes (Internal Battery)	Yes (Internal Battery)
Transmission range	Short (up to 6.096m)	Medium (up to 30.48m)	Long (up to 228.6m)
Communication pattern	Passive	Passive	Proactive
Cost	Cheap	Medium	Expensive
Type of memory	Mostly Read-Only	Read-Write	Read-Write
Life of tag	Up to 20 years	2 to 7 years	5 to 10 years

Table 3.2: Characteristics of passive, semi passive and active RFID tags

2) RFID reader

A RFID Reader is a scanning device which reads the tags reliably and communicates the results to the middleware. A reader uses its own antenna to communicate with the tag by broadcasting radio waves that will be replied to by all tags within range. Readers can process many items at once, allowing longer read processing times. They can be either mobile or stationary, and are differentiated by their storage capacity, processing capability and the frequency they can read. [19]

RFID reader consists of the following functional blocks:

a) HF interface

The master part of the reader which has these functions:

- Supplying RFID transponders with power by generating high frequency power
- Modulation of the signal to the transponder
- Reception and demodulation of signals from the transponders

b) Control unit

The slave part of the reader that performs the following functionalities:

- Communication and execution of the application software‘s commands
- Signal coding and decoding
- Communication control with a transponder

3) RFID middleware

The middleware broadly refers to software or devices that connect RFID readers to enterprise information systems, and the data they collect. RFID middleware helps to make sense of RFID tag reads, applies filtering, formatting, and logic to tag data collected by a reader, and provides back-end applications with this processed content. RFID middleware serves to manage the data flow between tag readers and enterprise applications, and is responsible for the consistency and thus usability of the content. It offers connectivity for writers, context-based filtering and routing, and integration of enterprise / B2B. The next sections will further address RFID middleware architecture and components. [19]

The following issues need to be considered when designing a solution to RFID middleware:

Multiple hardware support: The middleware must have a common interface to access various hardware types providing different features. Synchronization and scheduling: All middleware processes should have intelligent scheduling and synchronization. This minimizes latency, and enhances middleware efficiency. RFID readers handle incoming data in real time: The middleware is supposed to handle the enormous amount of data captured in real time by the connected readers without reading misses. Interfacing with multiple applications: the middleware should be capable of communicating concurrently with multiple applications by fulfilling all applications specifications with minimal latency. The application developer can only use the default set of interfaces that the middleware offers regardless of the type of hardware that is connected to it. Scalability: The middleware architecture must allow fast integration of new hardware features and data processing. [19]

3.2.3 RFID middleware components

A RFID middleware is the interface that sits between the RFID hardware and RFID applications. It provides the following advantages:

- It hides the RFID hardware details from the applications
- It handles and processes the raw RFID data before passing it as aggregated events to the applications
- It provides an application level interface for managing RFID readers and querying the RFID data

A RFID middleware layer integrates all system drivers of different hardware and exposes standard interfaces to this hardware for access to the program. If the application has been supplied with all system drivers of all connected readers, handling and interfacing each of the devices would be a hard job. The application developer would then have to consider all of the internals and operations unique to the hardware. Also, the application would find it very difficult to process the data in real time if it comes with the large amount of raw tag data recorded by the readers. A RFID middleware provides a standardized way to handle this flood of information, which processes the raw data and provides clean and filtered data to the application.
[19]

As shown in Figure 3.4 a RFID middleware is generally composed of four major layers:

- Reader Interface
- Data Processor and Storage
- Application Interface
- Middleware Management

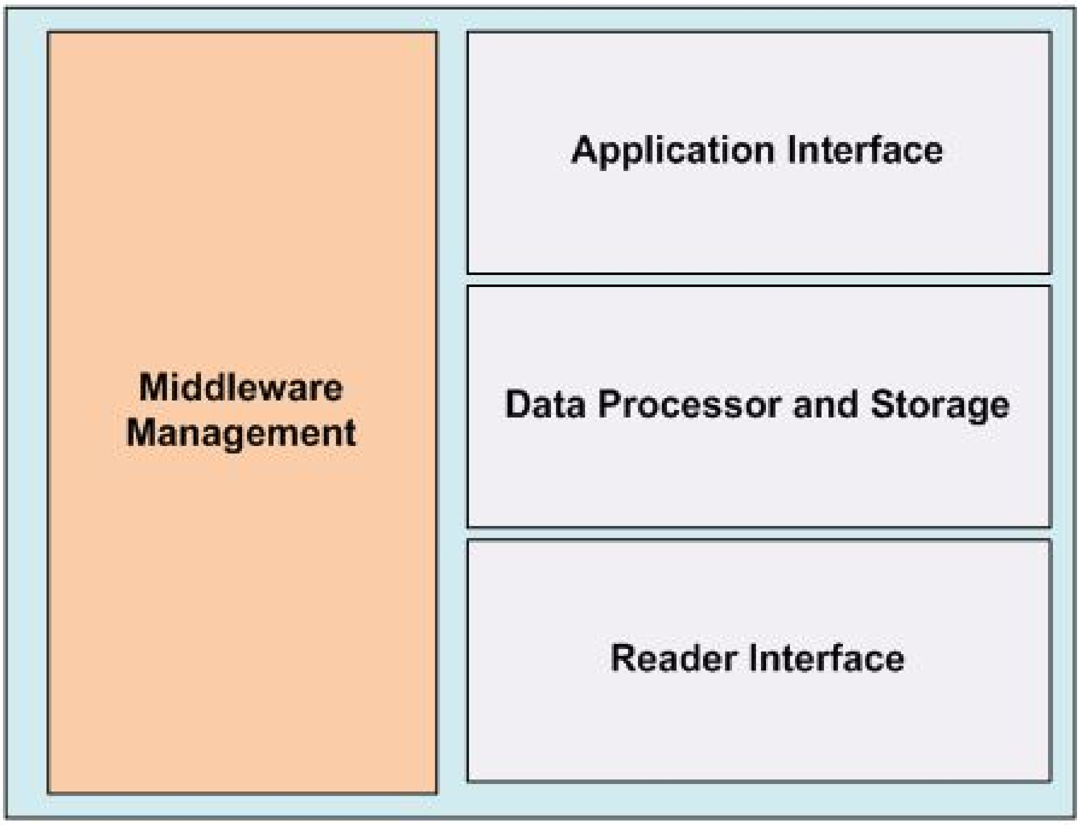

Figure 3.4: RFID middleware components

1) Reader interface

The reader interface is the lowest RFID middleware layer which handles the RFID hardware interaction. It maintains device drivers of all system-supported devices, and manages all parameters relevant to hardware such as reader protocol, air interface, and host-side communication.

2) Data processor and storage

It is the responsibility of the data processor and storage layer to process and store raw data coming from the readers. The data filtering, aggregation, and transformation are examples of processing logic carried by this layer. This layer also processes the events associated with a particular application on the data level.

3) Application interface

The application interface provides an API to access, communicate, and configure the middleware for the RFID. It integrates the enterprise applications with the RFID middleware by translating requests of the applications into middleware commands of low level.

4) Middleware management

The middleware management layer helps managing the configuration of the RFID middleware, and provides the following capabilities:

- Add, configure, and modify connected RFID readers
- Modify application level parameters such as filters, and duplicate removal timing window
- Add and remove services supported by the RFID middleware

Typically, RFID readers are abstracted as a logical reader that is either a collection of several readers, or a part of the reader. This grouping method is used when a collection of readers is required to capture data from a specific area, such as a warehouse with several loading docks. The benefit of this is that the program will question a limited number of logical readers, instead of combining events from each of the readers.

3.3 Face Recognition and OpenCV

3.3.1 Face Detection

Face detection applications use machine learning and formulas known as algorithms to detecting human faces within larger images. These larger images might contain numerous objects that aren't faces such as landscapes, buildings and other parts of humans (e.g. legs, shoulders and arms).

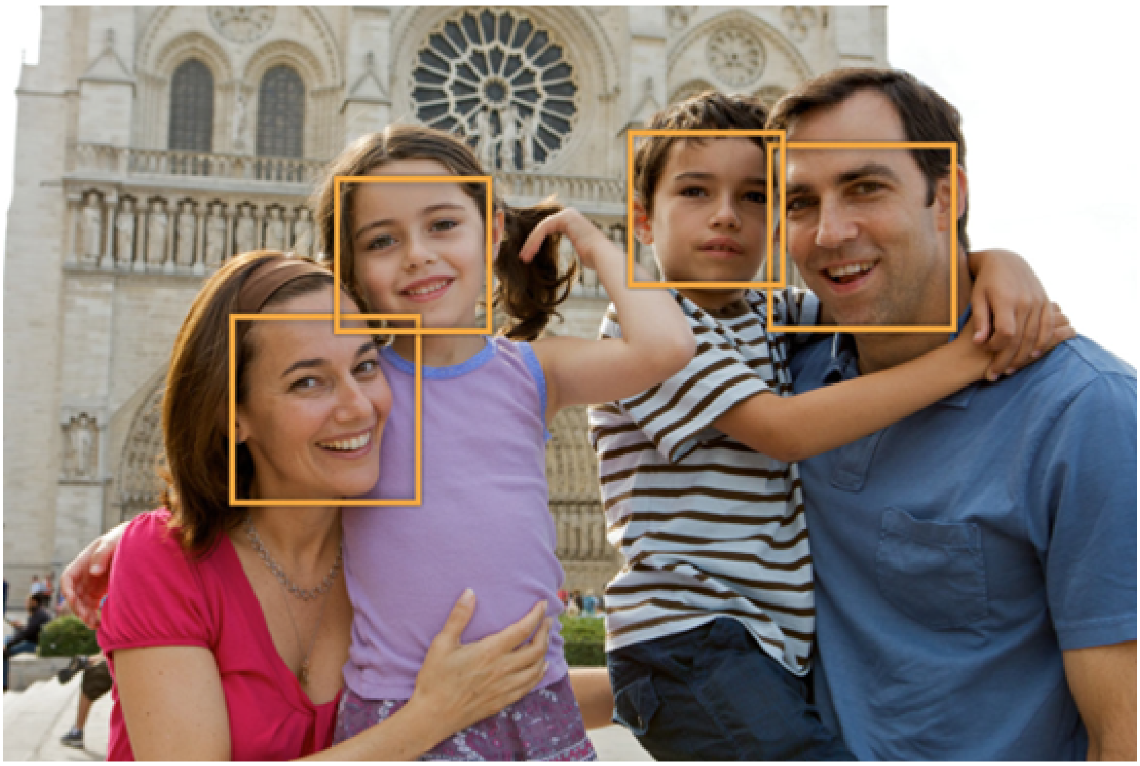

Figure 3.5: Face detection

Face detection is a broader term than face recognition. Face detection just means that a system is able to identify that there is a human face present in an image or video. Face detection has several applications, only one of which is facial recognition. Face detection can also be used to auto focus cameras. And it can be used to count how many people have entered a particular area. It can even be used for marketing purposes. For example, advertisements can be displayed the moment a face is recognized.

3.3.2 Face Recognition

Face recognition is a technology that can recognize or verify a subject through a face picture, video or any audio-visual item. It's a biometric identification system that uses body measurements to verify a person's identity, in this case face and head. To identify, verify and/or authenticate a individual, the technology collects a collection of specific biometric data from and individual correlated with their face and face expression[20]. Face recognition systems record an incoming image in a two-dimensional or three-dimensional way from a camera device, depending on the device's characteristics.

Figure 3.6: Face recognition

These compare the relevant details of the incoming image signal in picture or video in a database in real-time, being much more accurate and stable than the information obtained in a static image; This method involves an Internet connection because it is not possible to locate the database on the capture computer, as it is hosted on servers. In this face comparison, it analyzes the incoming picture mathematically without any margin of error and verifies that the biometric data fits the person who is expected to use the service or who is requesting access to an application, device or even building.

Face recognition systems can operate with the highest safety and reliability standards, thanks to the use of artificial intelligence (AI) and machine learning technologies. Similarly, the process can be carried out in real-time, thanks to the integration of these algorithms and computing techniques.

3.3.3 OpenCV

What is OpenCV?

OpenCV is the most popular library for computer vision. Originally written in C/C++, it now provides bindings for Python. It was started at Intel in 1999 by Gary Bradsky, and the first release came out in 2000. Vadim Pisarevsky joined Gary Bradsky to manage Intel's Russian software OpenCV team. In 2005, OpenCV was used on Stanley, the vehicle that won the 2005 DARPA Grand Challenge. Later, its active development continued under the support of Willow Garage with Gary Bradsky and Vadim Pisarevsky leading the project. OpenCV now supports a multitude of algorithms related to Computer Vision and Machine Learning and is expanding day by day.

OpenCV supports a wide variety of programming languages such as C++, Python, Java, etc., and is available on different platforms including Windows, Linux, OS X, Android, and iOS. Interfaces for high-speed GPU operations based on CUDA and OpenCL are also under active development.

OpenCV-Python is the Python API for OpenCV, combining the best qualities of the OpenCV C++ API and the Python language.[21]

Figure 3.7: OpenCV

OpenCV uses machine learning algorithms to search for faces within a picture. Because faces are so complicated, there isn't one simple test that will tell you if it found a face or not. Instead, there are thousands of small patterns and features that must be matched. The algorithms break the task of identifying the face into thousands of smaller, bite-sized tasks, each of which is easy to solve. These tasks are also called classifiers. For something like a face, you might have 6,000 or more classifiers, all of which must match for a face to be detected (within error limits, of course). But therein lies the problem: for face detection, the algorithm starts at the top left of a picture and moves down across small blocks of data, looking at each block, constantly asking, "Is this a face? ... Is this a face? ... Is this a face?" Since there are 6,000 or more tests per block, you might have millions of calculations to do, which will grind your computer to a halt.

To get around this, OpenCV uses cascades. What's a cascade? The best answer can be found in the dictionary: "a waterfall or series of waterfalls." Like a series of waterfalls, the OpenCV cascade breaks the problem of detecting faces into multiple stages. For each block, it does a very rough and quick test. If that passes, it does a slightly more detailed test, and so on. The algorithm may have 30 to 50 of these stages or cascades, and it will only detect a face if all stages pass.

The advantage is that the majority of the picture will return a negative during the first few stages, which means the algorithm won't waste time testing all 6,000 features on it. Instead of taking hours, face detection can now be done in real time.

OpenCV-Python

OpenCV-Python is a library of Python bindings designed to solve computer vision problems

Python is a general purpose programming language started by Guido van Rossum that became very popular very quickly, mainly because of its simplicity and code readability. It enables the programmer to express ideas in fewer lines of code without reducing readability.

Compared to languages like C/C++, Python is slower. That said, Python can be easily extended with C/C++, which allows us to write computationally intensive code in C/C++ and create Python wrappers that can be used as Python modules. This gives us two advantages: first, the code is as fast as the original C/C++ code (since it is the actual C++ code working in background) and second, it easier to code in Python than C/C++. OpenCV-Python is a Python wrapper for the original OpenCV C++ implementation.

OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.[21]

3.4 Real-Time Eye Blink Detection using Facial Landmarks

3.4.1 Facial landmarks

Facial landmarks are used to localize and represent salient regions of the face, such as: (eyes, eyebrows, nose, mouth, jawline)

Facial landmarks have been successfully applied to face alignment, head pose estimation, face swapping, blink detection and much more.

What are facial landmarks?

Figure 3.8: Facial landmarks are used to label and identify key facial attributes in an image

Detecting facial landmarks is a subset of the shape prediction problem. Given an input image (and normally an ROI that specifies the object of interest), a shape predictor attempts to localize key points of interest along the shape. In the context of facial landmarks, our goal is detect important facial structures on the face using shape prediction methods. Detecting facial landmarks is therefore a two step process:

- Step 1: Localize the face in the image.
- Step 2: Detect the key facial structures on the face ROI.

Face detection (Step 1) can be achieved in a number of ways. We could use OpenCV's built-in Haar cascades. We might apply a pre-trained HOG + Linear SVM object detector specifically for the task of face detection. Or we might even use deep learning-based algorithms for face localization. In either case, the actual algorithm used to detect the face in the image doesn't matter. Instead, what's important is that through some method we obtain the face bounding box (i.e., the (x, y)-coordinates of the face in the image). Given the face region we can then apply Step 2: detecting key facial structures in the face region. There are a variety of facial landmark detectors, but all methods essentially try to localize and label the following facial regions: (eyes, eyebrows, nose, mouth, jawline)

Understanding dlib's facial landmark detector

The pre-trained facial landmark detector inside the dlib library is used to estimate the location of 68 (x, y)-coordinates that map to facial structures on the face. The indexes of the 68 coordinates can be visualized on the image below:

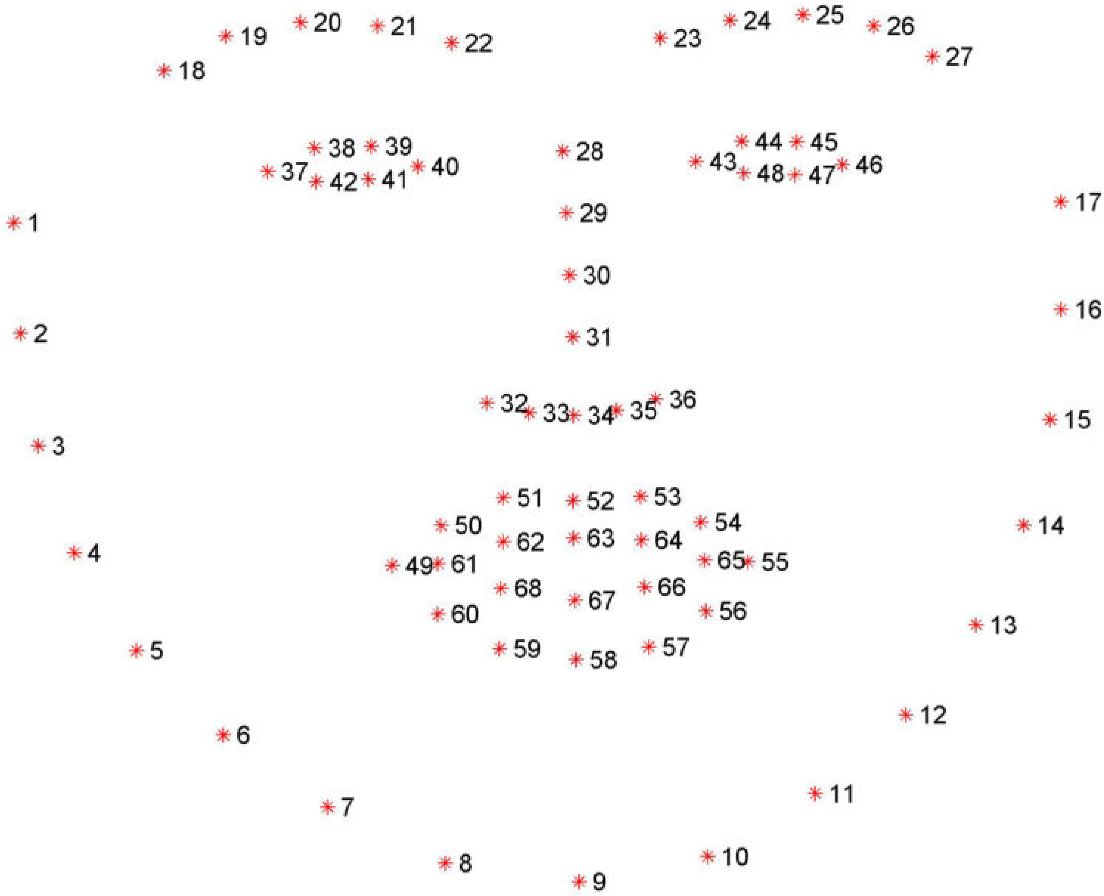

Figure 3.9: Visualizing the 68 facial landmark coordinates from the iBUG 300-W

3.4.2 Eye Aspect Ratio (EAR)

In terms of blink detection, we are only interested in two sets of facial structures: the eyes.

Each eye is represented by 6 (x, y)-coordinates, starting at the left-corner of the eye (as if you were looking at the person), and then working clockwise around the remainder of the region:[22]

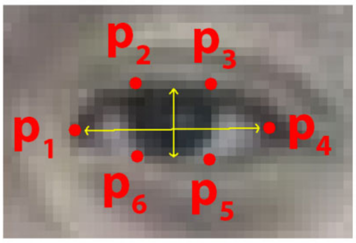

Figure 3.10: The 6 facial landmarks associated with the eye.

Based on this image, we should take away on key point:

There is a relation between the width and the height of these coordinates.

Based on the work by Soukupová and Čech in their 2016 paper, Real-Time Eye Blink Detection using Facial Landmarks, we can then derive an equation that reflects this relation called the eye aspect ratio (EAR):

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Figure 3.11: The eye aspect ratio equation.

Where p1, . . . , p6 are 2D facial landmark locations.

The numerator of this equation computes the distance between the vertical eye landmarks while the denominator computes the distance between horizontal eye landmarks, weighting the denominator appropriately since there is only one set of horizontal points but two sets of vertical points.

Why is this equation so interesting?

Well, as we'll find out, the eye aspect ratio is approximately constant while the eye is open, but will rapidly fall to zero when a blink is taking place.

Using this simple equation, we can avoid image processing techniques and simply rely on the ratio of eye landmark distances to determine if a person is blinking.

To make this more clear, consider the following figure from Soukupová and Čech: [22]

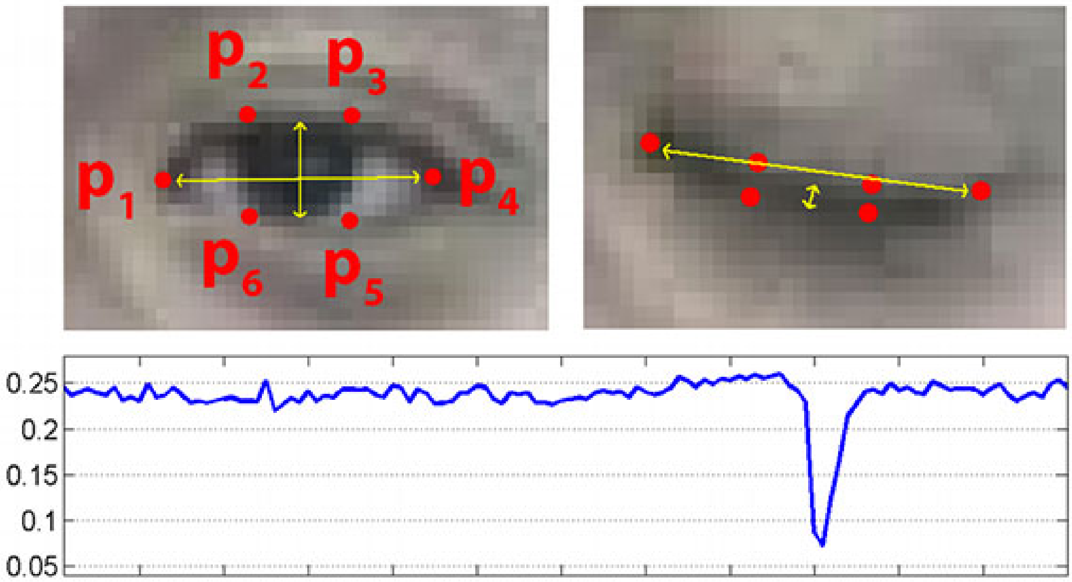

Figure 3.12: Top-left: A visualization of eye landmarks when then the eye is open. Top-right: Eye landmarks when the eye is closed. Bottom: Plotting the eye aspect ratio over time. The dip in the eye aspect ratio indicates a blink (Figure 1 of Soukupová and Čech).

On the top-left we have an eye that is fully open — the eye aspect ratio here would be large(r) and relatively constant over time.

However, once the person blinks (top-right) the eye aspect ratio decreases dramatically, approaching zero.

The bottom figure plots a graph of the eye aspect ratio over time for a video clip. As we can see, the eye aspect ratio is constant, then rapidly drops close to zero, then increases again, indicating a single blink has taken place.

In our next section, we'll learn how to implement the eye aspect ratio for blink detection using facial landmarks, OpenCV, Python, and dlib.[22]

Chapter 4

Methodology

This section will identify the methods used to attain some of the project's features. The project's key functionality is to mark student attendance after verifying his / her RFID card value and recognizing the student's face. Raspberry Pi board is used as the project 's principal controller and python is used to program the device. The system built as a portable device is one speciality of this project. All existing attendance management systems must be attached to a particular location and I can not switch from one location to another. The newly developed program has access to information on the capacity and attendance through the internet.

The development process includes the following four major steps:

1. Planning
2. Analysis
3. Design
4. Implementation

4.1 Planning

This was the project starting-point. The key reason this project came up with was problems faced by the students and lecturers due to attendance. The standard practice for getting student attendance in Vietnam is for students to parse an attendance sheet so students need to register for attendance. There are several problems to this method. Resident, many often fail to sign the attendance. Often one student can sign in his or her friend's name. If the speaker forgets to carry the attendance sheet, then the entire attendance for that session will be lost. There are a lot of issues, as well. After identifying issues, the next step was to find existing literature. Most kinds of attendance management systems are developed using a different kind of technology. Fingerprint-based attendance management system is the most common. Although this method is ideally suited to places like companies, it is not very appropriate for university or educational institution. There are several lecture and lab sessions while considering educational institute. There are a large number of students who may take part in one session. Hence it's not practical to wait for a big time in a queue to mark attendance. Student fingers may have dirt, oil, water. In these cases it is difficult to determine the right fingerprint.

It has developed several attendance control systems based on RFID technology. Especially in this situation, student often carry on behalf of a friend the RFID tag of his or her self. So, that's going to mark fake attendance. Some researchers developed attendance management systems based on face recognition. This also has some problems, since students' appearance can change from time to time. Thus, the new system designed and developed by integrating RFID technology and face recognition together, by considering all of these cases.

4.2 Analysis

This phase focuses mainly on the study of the requirements gathered. Several attendance management systems were developed using different technologies. Some of them are based on fingerprints, some on RFID and some others on face recognition. The key emphasis in this study is the production of the most efficient and easy to operate device. I recognized, when considering usability, that designing a portable device is easier. Then the next thing we took into account was low cost.

The analysis shows that a developing system that uses Fingerprint is not practical for an educational institute because a large number of students and fingers may have particles of dust. It is also not realistic to just use face recognition, because if a student's appearance is changed, the device would be unable to recognise the student. It can also have some problems when using RFID technology, including fake attendance. If we are able to use both face recognition and RFID technologies together, this reduces many problems that can occur when they are used individually. In addition to considering this technology, it is very useful if we can store details of the attendance in a cloud environment. Such data can then be accessed online by approved parties, and data would be safer than stored in a local environment. The hardware components used were less costly in comparison to other current on the market systems. This process of research was very useful to arrive at a successful design for the new system.

4.3 Design

Following completion of the analytical process, system design was performed with the goal of translating the collected requirement specifications into a type appropriate for their implementation using a programming language. Architectural Analysis and Architectural Design were carried out in this phase. The device structure is analyzed using the information obtained during the previous level. System planning is focused on the system's performance.

The solution based on the technology for face recognition and Radio Frequency Identification. The system is planned to be built according to how it can be easily implemented on any Vietnamese university or institute. Here is an overview of the system that it proposed to construct.

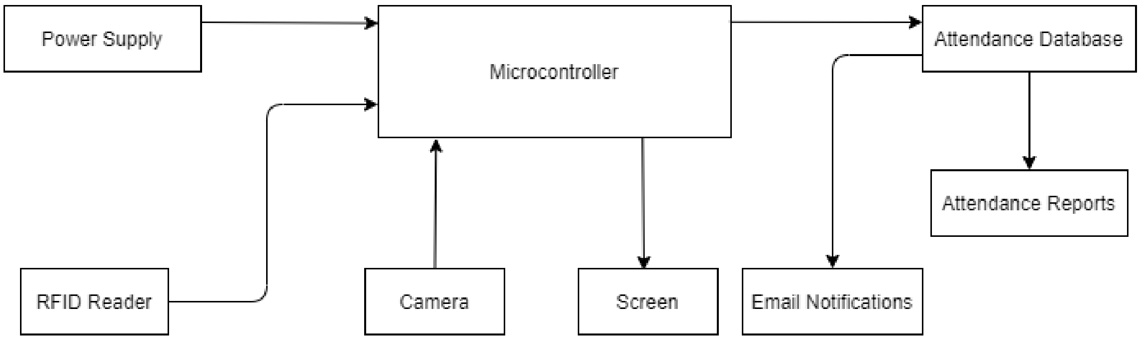

Figure 4.1: System Overview

The key concept is to locate the device near the entrance to the lecture hall or laboratory, and read RFID tags for identity first when students tap their RDID tag on the RFID reader. Until students start to enter the lecture hall, the set of students to participate in the lecture or lab practice will be allocated. The subjects and time slots that the student will be taking part will be stored in the database. When a student taps his / her student Id with an RFID tag, the program checks that the student is on the correct subject and the right slot. If those details match, a student's face is checked by the machine. Instead, by comparing the student's picture with the student's saved image, it ensures the correct student is presence. If all of these specifics are right, this marks the student's existence on the database. At the same time, it gives the student an email update. The flowchart for the designed system is shown here.

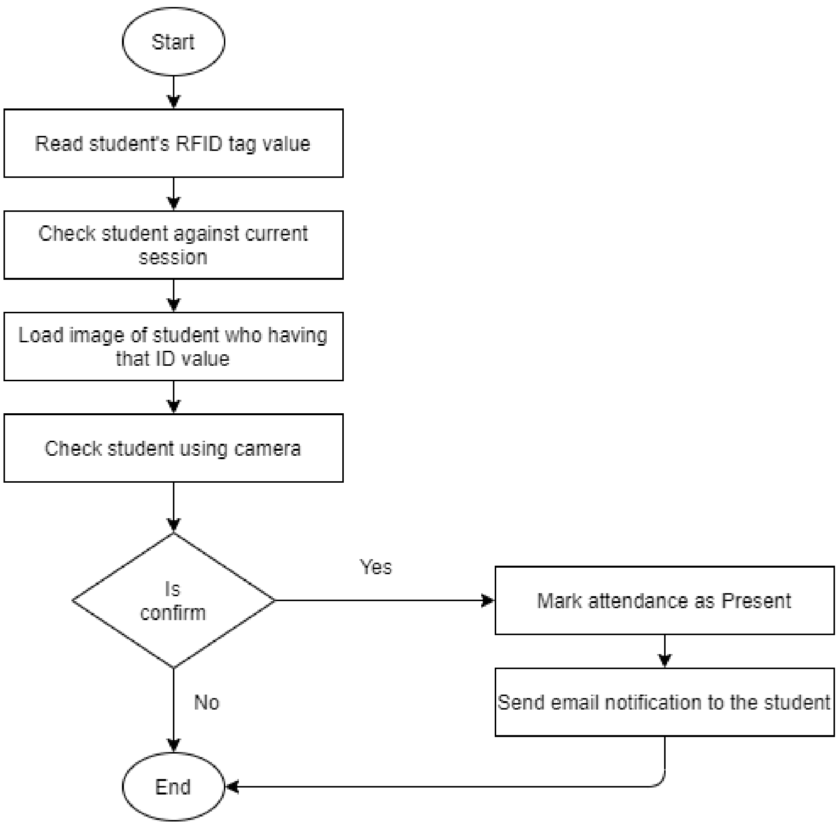

Figure 4.2: Flowchart of system

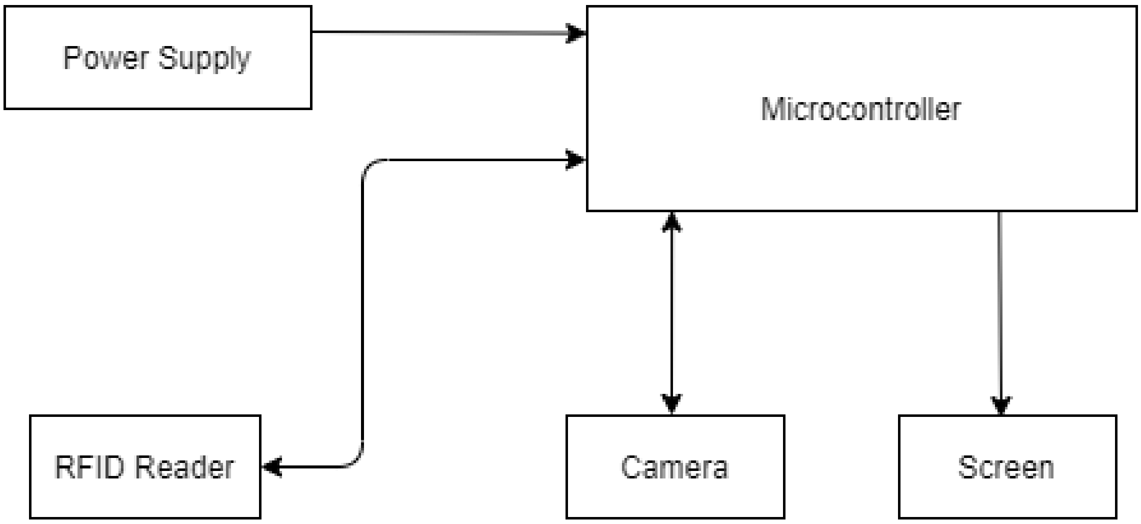

Figure 4.3: System Design

Function of each block:

- **Power Supply:** Supply 5V power to the Microcontroller. Powered from Raspberry Pi issued to the RFID Reader block.
- **RFID Reader:** To read and store fingerprint data, RC522 module sensor is used to control this block.
- **Microcontroller:** Collect data from components and handle Screen block.
- **Screen:** Display information when the user interacts, such as scanning the card, recognizing faces and checking schedules.
- **Camera:** Capture pictures of users when checking attendance.

Microcontroller

The Raspberry Pi 3 Model B

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

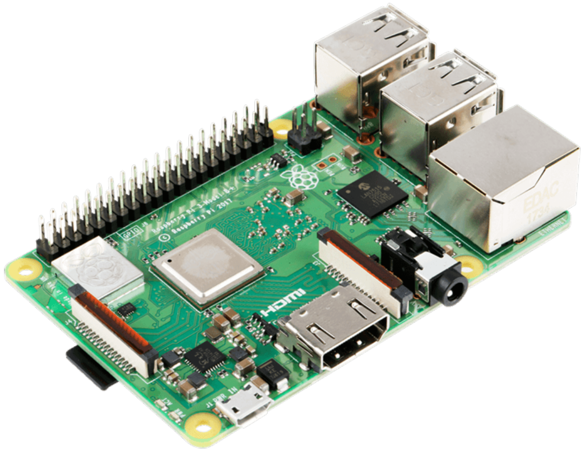

Figure 4.4: Raspberry Pi Model 3 B

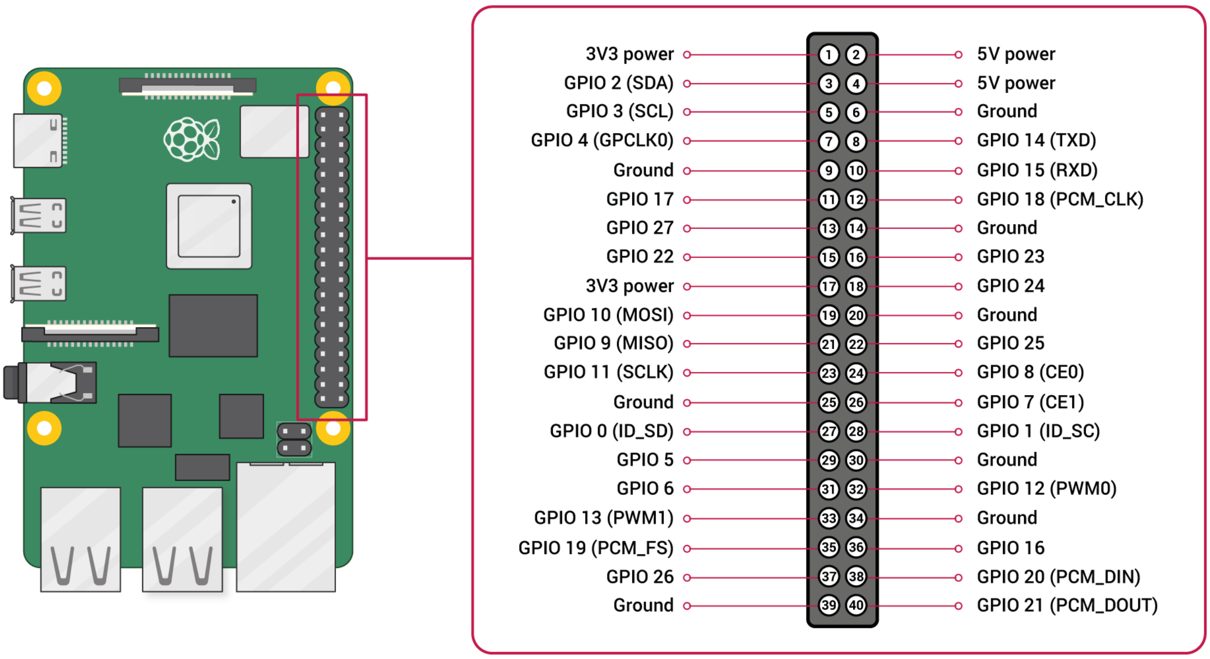

Figure 4.5: GPIO Pinout Diagram

RFID Reader

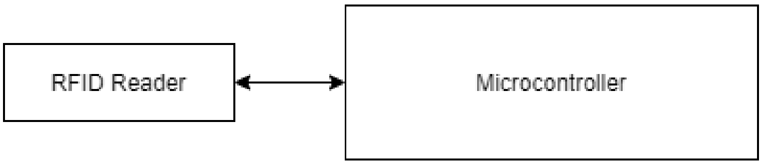

Figure 4.6: Relationship between Micro controller and RFID Reader

RFID Reader sends signals to Microcontroller, Microcontroller receive signals and then send to other blocks. RFID Reader can only active when the Microcontroller block requires.

RC522 Features:

- 13.56MHz RFID module
- Operating voltage: 2.5V to 3.3V
- Communication : SPI, I2C protocol, UART
- Maximum Data Rate: 10Mbps
- Read Range: 5cm
- Current Consumption: 13-26mA
- Power down mode consumption: 10uA (min)

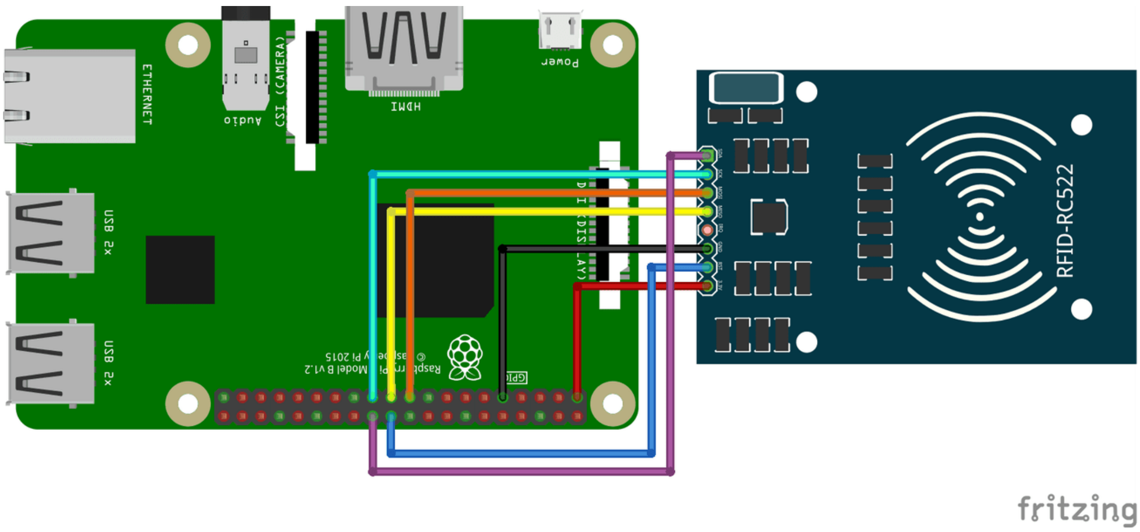

Figure 4.7: Raspberry and RC522 connection

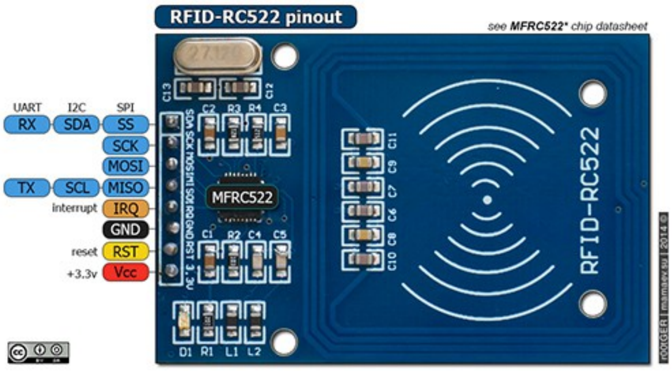

Figure 4.8: RC522 Pinout

RFID Reader	Raspberry Pi
3.3V	3.3V
RST	Pin 22 / GPIO 25 (can use any GPIO pins)
GND	GND
IRQ	Not connected
MISO	Pin 21 / GPIO 9 (MISO)
MOSI	Pin 19 / GPIO 10 (MOSI)
SCK	Pin 19 / Pin 23 / GPIO 11 (SCK)
SDA	Pin 19 / Pin 24 / GPIO 8 (CE0)

Table 4.1: RFID pins

Screen

The system uses computer screen connects to Raspberry Pi via HDMI plug. The screen displays user interface when users interact with the system. User interface of 3 main phases: detect RFID card, recognize students' faces and display the timetable.

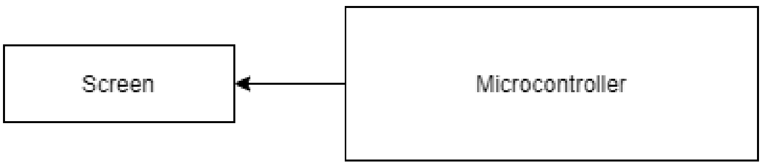

Figure 4.9: Relationship between Microcontroller and Screen

Camera

Use Raspberry Pi Camera Module to connect with Raspberry Pi via CSI camera port.

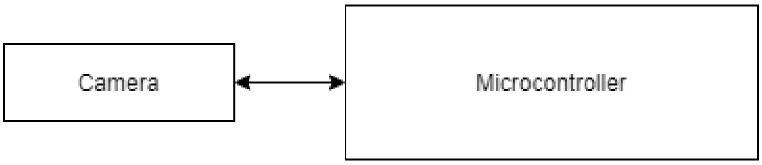

Figure 4.10: Relationship between Microcontroller and Camera

Figure 4.11: Camera connection

The database of the system is developed according to the following entity relationship diagram. There are four main entities as Administrator, Student, Course and Attendance.

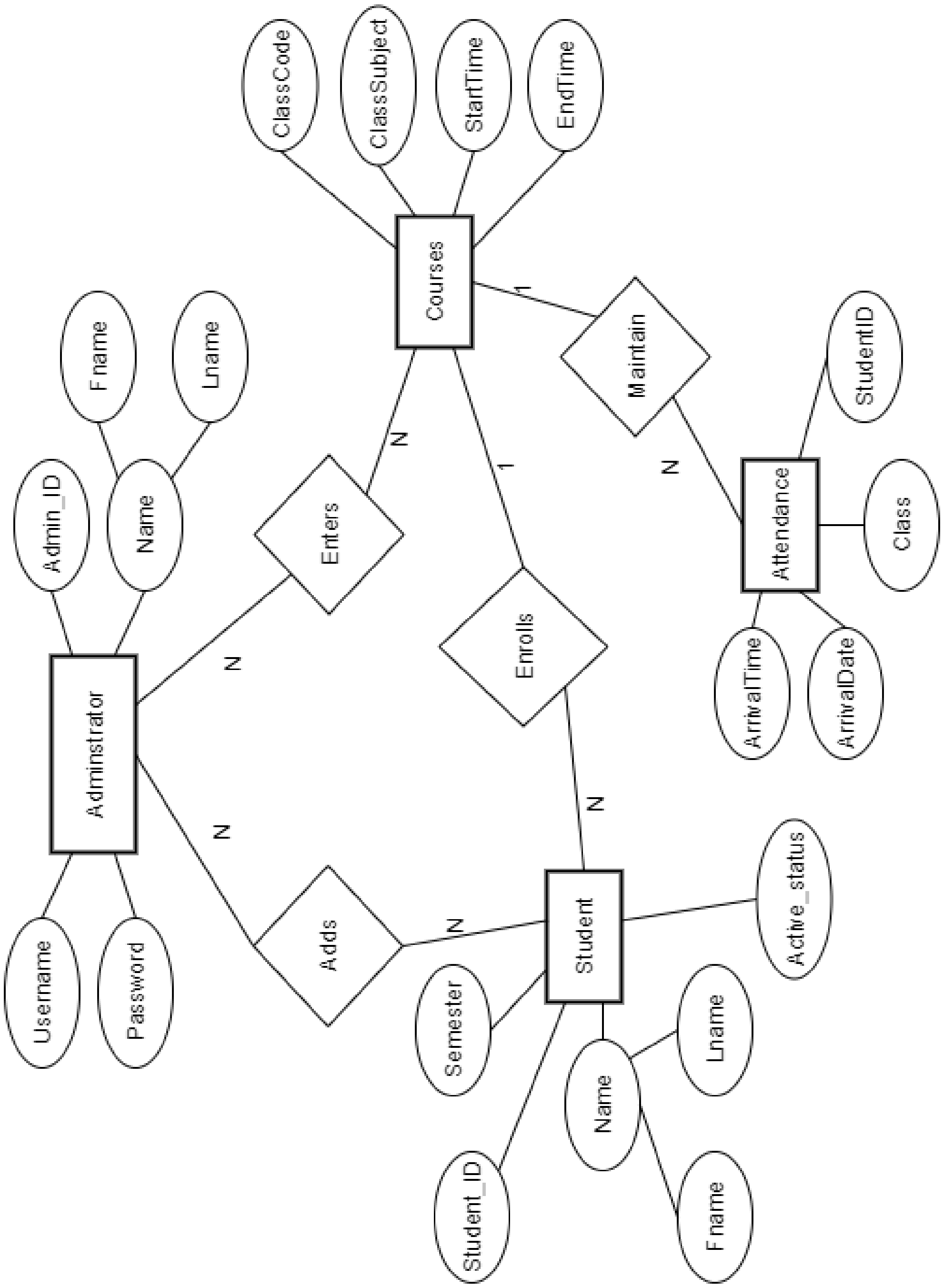

Figure 4.12: ER Diagram of database

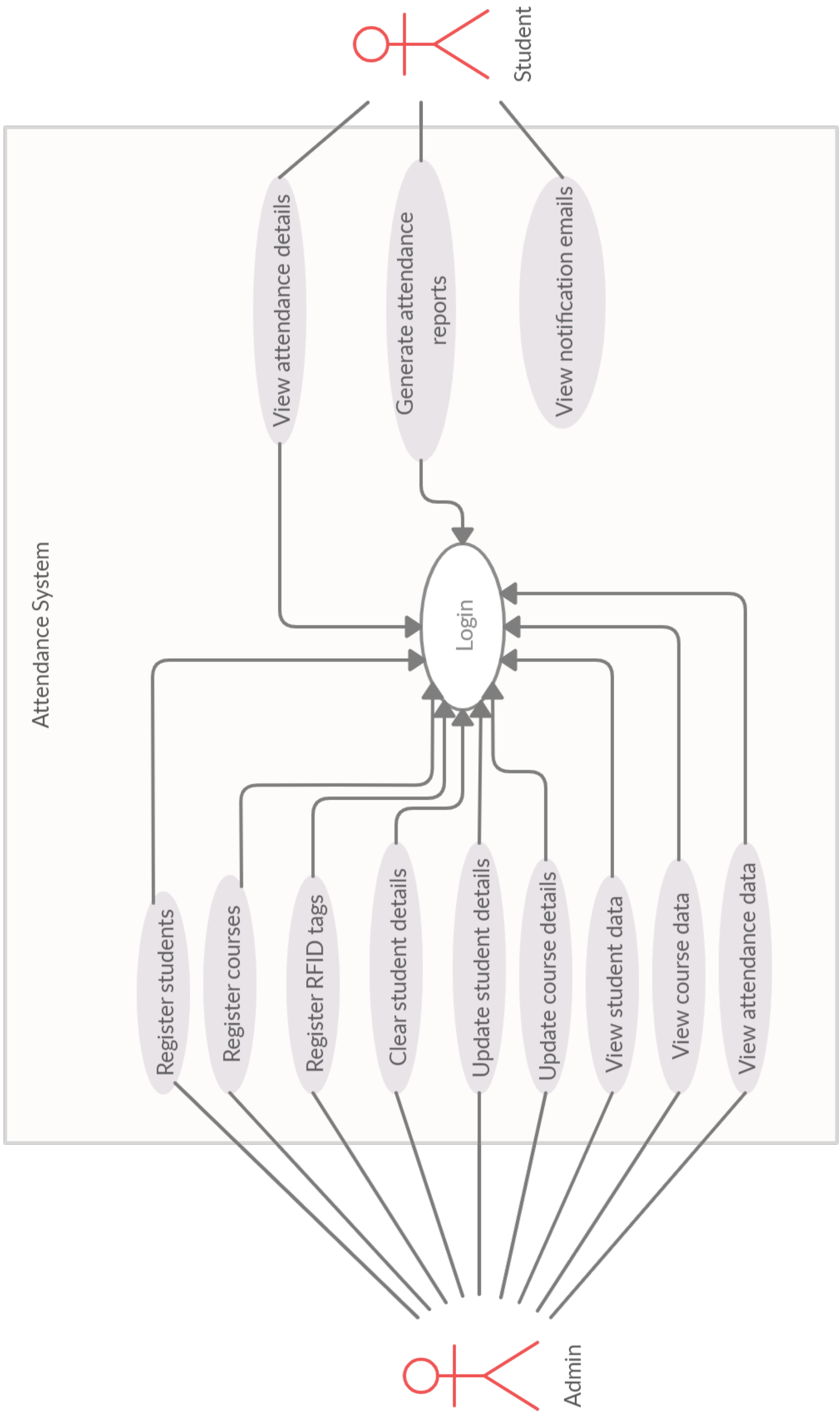

Figure 4.13: Use case diagram

4.4 Implementation

The main objective of this research was to develop and implement an automated attendance management system automated by the Internet of Things. In order to provide a more reliable attendance management system, the study was proposed to incorporate both facial recognition and radio frequency detection into one system. To achieving the goal, it was needed to come up with some sub-objectives.

4.4.1 Identify Students Using RFID Tags

The RFID reader attached to the device should read the tag and identify the student as the first step to identify if a student has his/her RFID tag. RFID reader was connected to a breadboard and reader and Raspberry pi connected using breadboard wires. RC522 RFID reader was used to reading RFID tags and for register RFID tags for students.

RC522 RFID reader contains eight pins and first, it was needed to get a proper understanding on the reader. This is operating at 3.3V current and 13.56MHz. The eight pins of the reader are SDA (Serial Data Signal), SCK (Serial Clock), MOSI (Master Out Slave in), MISO (Master in Slave Out), IRQ (Interrupt Request), GND (Ground Power), RST (Reset Circuit) and (PWR) 3.3V power in pin. It is possible to connect RFID reader directly to Raspberry pi but for safe, the reader is connected to a breadboard and wired.

Table 4.1 shows how RFID reader wired with Raspberry Pi.

After connecting hardware components, the next step was to install software components that are needed to work with an RC522 RFID reader. First, it was needed to set up python in order to work with an RFID reader. So, a python library called "SPI-Py" was cloned and installed to Raspberry Pi. This library contains key components for handle interactions with an RFID reader. Following command used to do the cloning.

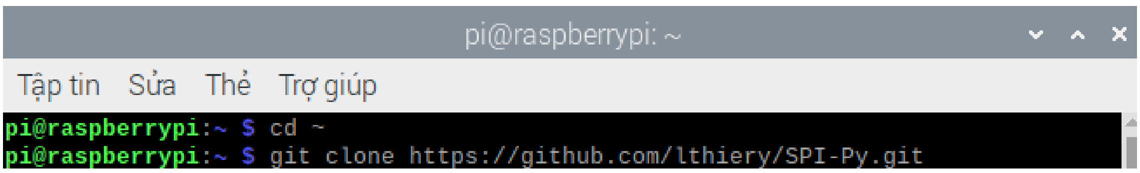

Figure 4.14: Command to clone

Then it was possible to do the installation as follows:

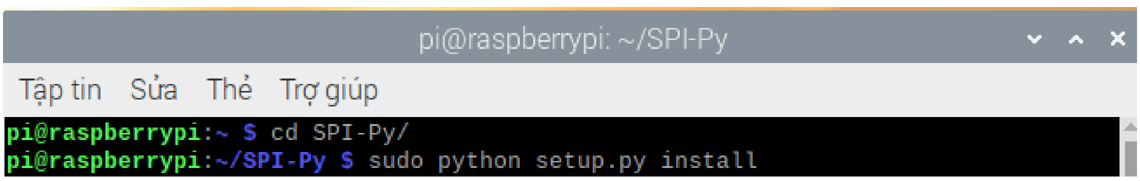

Figure 4.15: Installation

After that MRFC522 library installed and set the environment to work with an RFID reader. There were two things done with RFID in this research. One was registering RFID tags for students. Another one was reading RFID tags and mark attendance. RC522 component used for both of these purposes.

Reading RFID Values

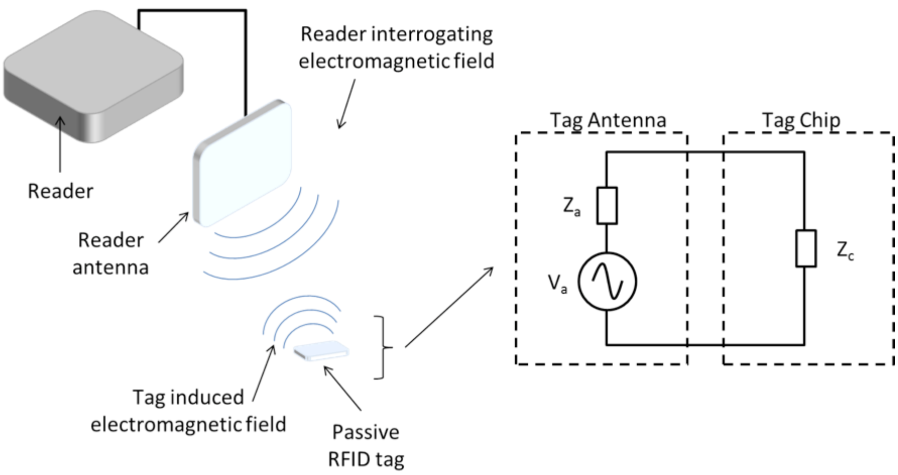

Figure 4.16: Reading RFID values

To read values, the user needs to tap on RFID reader so that reader needs to read the value of that tag. The current research used an active RFID reader and a passive RFID tag. RFID tag has a silicon microchip inside which is attached to a small antenna and entire module encapsulated to a glass or plastic container.

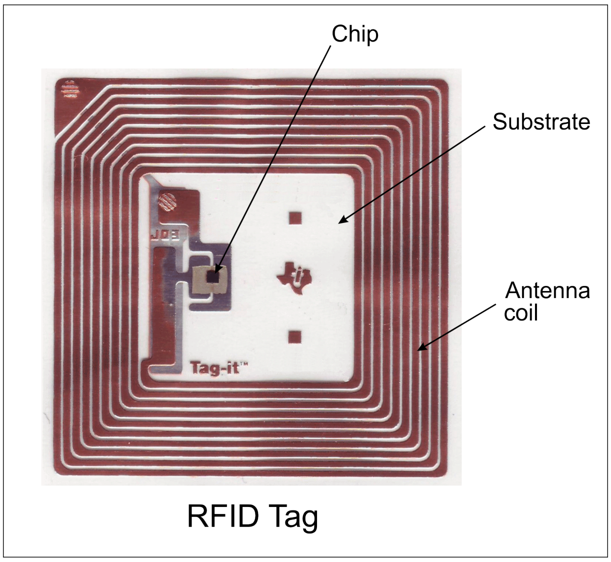

Figure 4.17: RFID tag inside

RFID reader consists of a radio signal transmitting and receiving scanner and an antenna.

This is the part that interacts with the RFID tag and receives RFID data.

There are two types of RFID systems as Active RFID systems and passive RFID systems.

1) **Active RFID systems:** This system contains RFID tags with a battery-like power supply. The battery life or power source is the only restriction for these tags. In corporate applications, usually active RFID systems are used that must be capable of reading RFID tags that are accessible at some distance from passive RFID systems.

2) **Passive RFID systems:** The RFID tags get power prom RFID reader in this kind of systems. The method called "Inductive coupling method" or "Electro Magnetic method" used to get power to the tags. In the inductive coupling method, the reader contains a coil which is connected to the AC power supply so that a magnetic field generated around it. Then tag placed on reader coil and an electromotive force is inducted on the tag. So, a current generates on the tag and it transmits the signals. In Electro Magnetic method, the reader transmits electromagnetic waves and the antenna in tag receives those waves. Then a voltage is generated and used to get the Direct current power. The tag transmits a signal using that power and reader receive that signal and monitor.

The following research also used a passive RFID system and program for reading RFID tags was written using python programming language. First, it is needed to import necessary python library files to use in built functions of python and for communicating with the Raspberry Pi module. GPIO (General Purpose Input Output) pins of Raspberry Pi module used to connect RC522 reader with Raspberry Pi. There are 40 GPIO pins on a current Raspberry Pi module so it enables users to connect inputs and outputs with raspberry pi. To communicate via these pins, it is needed to import the library file within a python script. After that, another importing done to access RFID reading functions. That was SimpleMFRC522. When a reading success a LED bulb will blink. So, to program that task, it was needed to access function called sleep. That function comes with a python library called time. Therefore, at the beginning of the python script following three imports were done on behalf of the reading value of RFID tag.

```
import RPi.GPIO as GPIO
import SimpleMFRC522
import time
```

Figure 4.18: Importing

Then the mode for GPIO pin numbering is set in to BCM (Broadcom) and skip warnings. Another important thing needs to do is setting Raspberry pi pin for output. So, these three tasks performed as follows.

```
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(18,GPIO.OUT)
```

Figure 4.19: Set pins

Then two functions are written to read data and blink LED. Following is the function to blink the LED.

```
def blink_led():
    GPIO.output(18,GPIO.HIGH)
    time.sleep(1)
    GPIO.output(18,GPIO.LOW)
```

Figure 4.20: Blink LED

That function set 18^{th} GPIO pin high to turn on the LED and keep it on for 1 second. After that, it turns off the LED.

Following is the function to read the value from RFID tag. That will read the id of tag and text stored in the tag. The blink_led function is called within the get_data function.

```
def get_data():
        id, text = reader.read()
        print(id)
        print(text)
    blink_led()
```

Figure 4.21: Get data

After implementing these two functions, reading values from RFID reader performed using SimpleMFRC522 method.

Writing to RFID Tag

When RFID tags registering for students it is required to write student identification number to the RFID tag. This should do with the process of student registration to the institute. Only thing needs to do for writing is executes the script and tap the tag on RC522 module. Then the system will prompt to enter student name or student identification number. So, the user needs to enter those details. Likewise, any number of student can be register at once. The system administrator of the institute can perform this task very easily by connecting the device to a monitor using an HDMI cable. After setting those initial configurations, the system can use portably in the same Wi-Fi network range. Following is the script to write to RFID tag.

```
#!/usr/bin/env python

import RPi.GPIO as GPIO
import SimpleMFRC522

reader = SimpleMFRC522.SimpleMFRC522()

try:
        text = raw_input('New data:')
        print("Now place your tag to write")
        reader.write(text)
        print("Written")
finally:
        GPIO.cleanup()
```

Figure 4.22: Writing

4.4.2 Creating A MySQL Database

To store attendance details, it is necessary to have a database. First we created a local database on Raspberry Pi and after completing database locally, it was implemented on a cloud environment.

Database part starts by installing MySQL server and Python bindings for MySQL on Raspberry Pi. Following command used to install them.

```
pi@raspberrypi: ~
Tập tin   Sửa   Thẻ   Trợ giúp
pi@raspberrypi:~ $ sudo apt-get install mysql-server python-mysqldb
```

Figure 4.23: Install MySQL

While the process of installation, it prompts and asked a password for MySQL root. After completing the installation, the shell is displayed and it was used to do all the configurations with the MySQL server. First, it is needed to login to MYSQL server root and create a database on the server. To create a database, the following code can be used.

```
thanhnguyen@thanhnguyen-FX503VD:~$ sudo mysql -u root -p
[sudo] password for thanhnguyen:
Enter password:
mysql> CREATE DATABASE attendance
mysql> USE attendance
```

Figure 4.24: Login to the root of MySQ

After creating a database, three tables created to store student RFID tags registration details, store course details and store student's attendance details. Student registration details table consists with student first name, last name, e mail address and course. Data is inserting to this table at the time of RFID tags are registering for students. In other words, when data writing to the RFID tags. Attendance table consists with student name, tag id, date and time. Course details table consists, of course, codes and course names.

At the time of working with a database, it is required to have a connection with the database. To connect to the database, it is needed to import the library to connect with the database at the beginning of the script. Following code segment will import MySQL python libraries.

```python
import mysql.connector
```

Figure 4.25: MySQL connector

Then it is necessary to start connection by specifying a host name, user, password and database. When local database used, the following code segment creates the connection to the MySQL database.

```python
mydb=mysql.connector.connect(
    host="localhost",
    user="admin",
    passwd="password",
    database="attendance"
)
```

Figure 4.26: Connecting to the database

After implementing database locally and checked, the database was hosted on a web server in order to get more expanded capabilities. For hosting the database, a free hosting service web site used that is called "db4free.net". This web site provides facility to host MySQL databases for free. A MySQL database created there with three tables called class detail, student data and attendance data.

Class detail table consists with id, class code, class subject, start time and end time. There, id is the primary key of the table and it is an auto incrementing field. "PhpMyAdmin" webserver is used to work with MySQL database. Here is the structure of the class detail table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	class_code	varchar(250)	utf8mb4_0900_ai_ci		No	*None*		
3	class_subject	varchar(250)	utf8mb4_0900_ai_ci		No	*None*		
4	start_time	varchar(250)	utf8mb4_0900_ai_ci		No	*None*		
5	end_time	varchar(250)	utf8mb4_0900_ai_ci		No	*None*		

Figure 4.27: Structure of class details table

Student data table consists with id, first name, last name, email address, joined date, joined time, class and active status. The ID is the primary key of this table and it is also an auto increment field. The class is the subject that student registered and active status gives whether student available for the course or not.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	first_name	varchar(255)	utf8mb4_0900_ai_ci		Yes	*NULL*		
3	last_name	varchar(255)	utf8mb4_0900_ai_ci		Yes	*NULL*		
4	email_address	varchar(255)	utf8mb4_0900_ai_ci		Yes	*NULL*		
5	join_date	varchar(20)	utf8mb4_0900_ai_ci		Yes	*NULL*		
6	join_time	varchar(10)	utf8mb4_0900_ai_ci		Yes	*NULL*		
7	class	varchar(250)	utf8mb4_0900_ai_ci		Yes	*NULL*		
8	active	varchar(10)	utf8mb4_0900_ai_ci		Yes	*NULL*		

Figure 4.28: Structure of students details table

Attendance data table has student id, class, arrived date and arrived time. Following is the structure of attendance data table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	student_id	varchar(255)	utf8mb4_0900_ai_ci		Yes	*NULL*		
3	class	varchar(255)	utf8mb4_0900_ai_ci		Yes	*NULL*		
4	arrived_date	varchar(20)	utf8mb4_0900_ai_ci		Yes	*NULL*		
5	arrived_time	varchar(10)	utf8mb4_0900_ai_ci		Yes	*NULL*		

Figure 4.29: Structure of attendance details table

4.4.3 Install OpenCV On Raspberry Pi

After implementing RFID reading and writing, the next main task was detecting students using face recognition. This face recognition was done using Open CV. Open CV is an open source library that contains many functions that are very useful in real time computer vision.

So it can be used to detect different objects and humans using video streams and images. The programming language used to work with Open CV was Python.

Installing Open CV on Raspberry pi was one hardest part of the project. The installation gives many errors because of incompatibilities between Raspberry Pi OS versions and Open CV versions. However, as a result of continuous trying, the installation became a success. The first step of the installation was expanding the file system to access all available space of SD card. Open CV used a big amount of space because it is working with images, videos and big library files. So before installing Open CV, that necessary configuration was done on Raspberry Pi. Before all the configurations, pi was updated and upgraded. Following command used to update and upgrade the Raspberry Pi.

Update Raspbian OS:

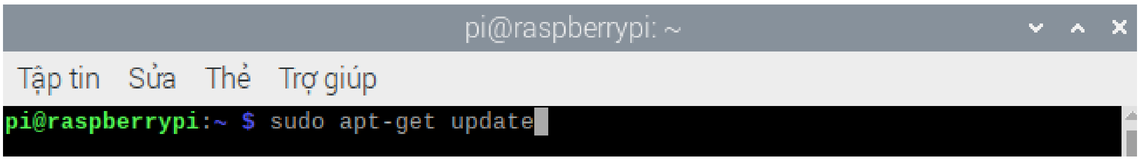

Figure 4.30: Update pi

Upgrade Raspbian OS:

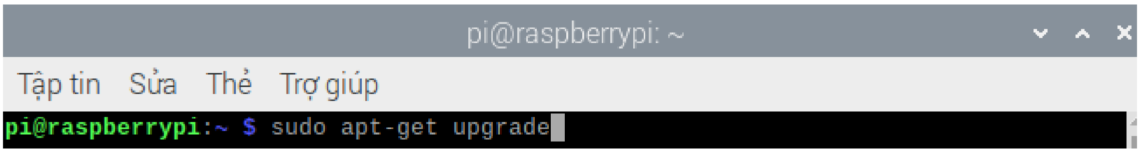

Figure 4.31: Upgrade pi

After doing that configuration, it was required to install dependencies with some developer tools. The developer tool used is called “Cmake” and it installed using the following command.

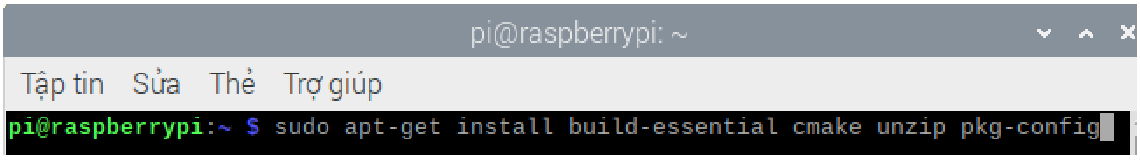

Figure 4.32: Install Cmake

After that, a collection of image and video libraries installed because they are required to work with image and video files. So, those dependencies are installed using following commands.

```
sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
sudo apt-get install libxvidcore-dev libx264-dev
sudo apt-get install libgtk2.0-dev libgtk-3-dev
sudo apt-get install libatlas-base-dev gfortran
```

Figure 4.33: Install dependencies

Then, the development tools for Python 3 is installed using the following command.

```
pi@raspberrypi: ~
Tập tin   Sửa   Thẻ   Trợ giúp
pi@raspberrypi:~ $ sudo apt-get install python3-setuptools python3-dev
```

Figure 4.34: Install Python

After installing python development tools, the pip tool installed as follows.

```
wget https://bootstrap.pypa.io/get-pip.py
sudo python3 get-pip.py
```

Figure 4.35: Install Python Tools

Install package python-opencv with following command in terminal (as root user).

```
thanhnguyen@thanhnguyen-FX503VD:~$ sudo apt-get install python-opencv
```

Figure 4.36: Get Open CV

Because of the programming language used is Python, there should be some configurations on a Python virtual environment. So, numpy is installed to the Raspberry pi using the following command.

```
sudo pip3 install numpy
```

Figure 4.37: Install numpy

Now, installation of Open CV was completed and it was tested using following commands.

```
import cv2
cv2.__version__
```

Figure 4.38: Test Open CV

This command gives the Open CV version as output. So it ensures that open cv installation was successful.

4.4.4 Face Recognition Overview

Recognize faces of students is the next main task of this research. This process is the biggest sub task of the project because face recognition is a result of a series of sub processes. It is needed to detect faces of humans at the beginning. After that, need to pre-process those faces. Then feature extraction should be performing before going to face recognition. When feature extraction is completed, the module must be trained to recognize faces and finally it can be evaluated and used to recognize students. The overview of this process can be shown as follows.

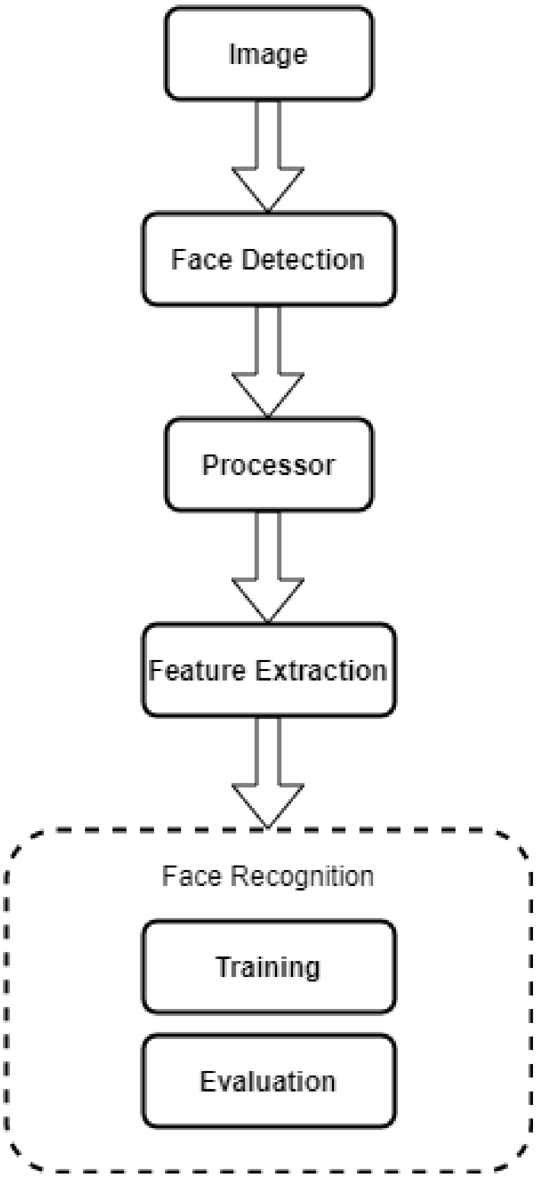

Figure 4.39: Face Recognition Overview

4.4.5 Face Detection

This is the beginning of the cycle of face recognition. In front of the screen, the appearance of a face and the position of the face in a picture should be identified. The Open CV library has a pre-trained facial detector based on deep - learning principles. Open CV contains Haar cascades designed for the detection of objects. The cascades have been trained with both positive and negative pictures. Positive pictures are present normal images, with negative pictures the inverse of positive pictures. In dark and dark areas, negative images display bright areas in light color. Positive images are images with image faces and negative images that are images without faces in face detection. Therefore, both positive and negative images are required for training the classifier in the face detection algorithm. The classifier is a computer program or an algorithm which decides whether an image is an image that is positive or negative. Features of these images should subsequently be extracted.

A trainer and a detector are given for Open CV. It has established pre-trained classifiers for various objects such as faces, eyes, smiles, etc. The open CV provides two types of pre-trained classifiers ready for face recognition. These classifications are Haar Classification and LBP Classification. These two classifiers are using grayscale images to detect the presence of faces as the color is not required to decide if faces are present in the image. All of them are XML files which contain with in opencv/data/ folder. Under that folder, Haar classifiers are available at opencv/data/ haarcascades folder and LBP classifiers are in another folder.

1) Haar cascade classifier:

This is a machine learning based classifier which is trained using images with faces and without faces. For extracting features, following three haar features used.

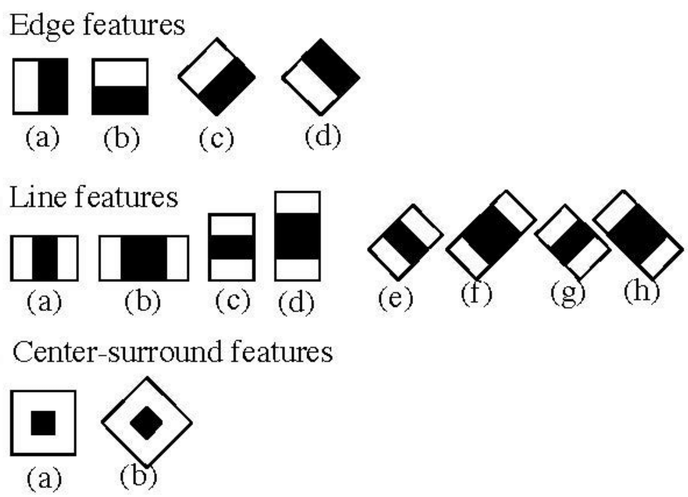

Figure 4.40: Haar Cascade Features

There is a window in the picture to calculate a single picture. This function is the single digit of the set of pixels collected from the black portion of the window, below the white section.

All windows of potential sizes are positioned at all possible image locations and a number of features are determined. Most of the characteristics are irrelevant in this calculation. Classifier then discards these irrelevant characteristics using the ”Adaboost” technique. Haar classification has high accuracy of face detection and a low false positive rate. The inconveniences of the Haar Classifier are complex, slow and take longer training periods.

2) LBP (Local Binary Patterns) cascade classifier:

This also required to train with a big number of images. LBP classification is simple and quick, but less precise. It needs a shorter training time robust to osculation. This is not suitable for systems such as attendance systems due to lower accuracy. The Haar Cascade Classifier is used in the study by considering a high face detection rate among these two classifiers.

The first step is to import the necessary library files when coding face detection. Cv2 is the first import. This imports the Open CV library so that we can use Open CV's built-in functions. This is a library for a python programming language which enables to work with large multi-dimensional arrays and metrics. Then python imagine library (PIL), that supports opening, manipulating and saving various image file formats, is required.

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

Figure 4.41: Imports

After that, the cascade classifier should be specified. It is possible to give cascade classifier as:

```
cascadePath = "/root/opencv-3.4.1/data/haarcascades/haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);
```

Figure 4.42: Set class path

Because it is needed to detect faces, haarcascade_frontalface_default.xml is used. Then, real-time video capturing can be started. To start the camera and videos, an object created from Video Capture that comes under open CV. The argument of that can be an index of device or video file name. The frame size of video can be set using the set method of Video Capture object.

```
cam = cv2.VideoCapture(0)
cam.set(3, 640) # set video widht
cam.set(4, 480) # set video height
```

Figure 4.43: Capturing objects

Then the minimum window size can be set to recognize an object as a face.

```
minW = 0.1*cam.get(3)
minH = 0.1*cam.get(4)
```

Figure 4.44: Set minimum Window Size

After that, the camera can be started by cam.read() method.

The next important task is, calling the classifier function and passing some important parameters to it.

```
faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (int(minW), int(minH)),
       )
```

Figure 4.45: Important parameters

- Here “gray” is to say input grayscale image.
- scaleFactor specifies how much of image size should reduce from each image scale.
- minNeighbirs gives a number of neighbors that should in each rectangle.
- minSize is minimum rectangle size to indicate a face.

The detected faces can be marked by a rectangle using the following code segment.

```
for(x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
```

Figure 4.46: Mark detected faces

The goal of this part is to recognize faces, not just detect faces. So it is necessary to have a proper dataset to train the recognizer. So, data gathering was done by getting many numbers of images from one person. At the student registration phase, it is required to get at least 30 images of a student’s face. This can be performing by running student registration script. When student registration is running, 30 images of one student captured and stored on face dataset. When the camera is on the student can stay freely and at that time, the system will get images from student’s video frame. So, it is possible to have images with different angles of the same face.

4.4.6 Training

This is the phase that trains face recognizer. This used all the user data on the dataset and train the recognizer using a specific Open CV function. The process ends with a result of the yml file and saved in the trainer directory.

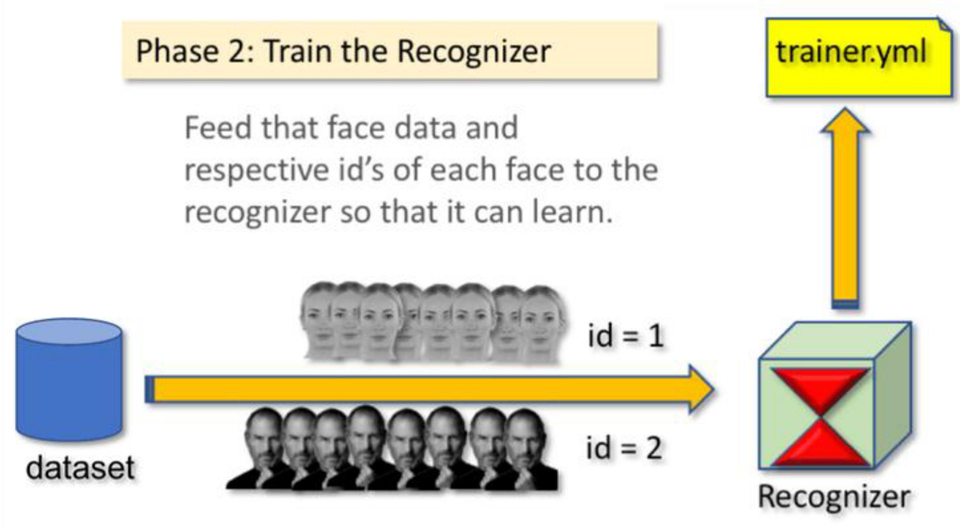

Figure 4.47: Face recognition training

The trainer script starts with importing required libraries. In that case, Open CV, numpy, PIL (Python Imagine Library) and OS (Operating System) libraries for python are imported.

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

Figure 4.48: Necessary imports

Then the path of an image dataset is defined as follows.

```
path = '/root/face_db/dataset'
```

Figure 4.49: Path of images

After that, two objects are created with the recognizer and detector. There are three types of built in recognizers in Open CV. Namely, they are Eigen Faces, Fisher faces and Local Binary Patterns Histograms (LBPH).

Eigen faces: This recognizer uses some distinct features of faces like eyes, mouth and nose to recognize person rather than considering all parts to recognize a person by face.

Fisher faces: This is an improved version of Eigen faces algorithm and it looks at all the faces to be trained at once and find principal components from all the faces.

Local Binary Patterns Histograms isn't looked at the whole image but it Tries to find the local structure by comparing each pixel with neighbor pixels. This research used LBPH recognizer in order to have a fast recognition.

```
recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("/root/opencv-3.4.1/data/haarcascades/haarcascade_frontalface_default.xml");
```

Figure 4.50: Recognizer

Then the following function was written to get the images and label data.

```
def getImagesAndLabels(path):

    imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
    faceSamples=[]
    ids = []

    for imagePath in imagePaths:

        PIL_img = Image.open(imagePath).convert('L') # convert it to grayscale
        img_numpy = np.array(PIL_img,'uint8')

        id = int(os.path.split(imagePath)[-1].split(".")[1])
        faces = detector.detectMultiScale(img_numpy)

        for (x,y,w,h) in faces:
            faceSamples.append(img_numpy[y:y+h,x:x+w])
            ids.append(id)

    return faceSamples,ids
```

Figure 4.51: Label data

After that, training can be performing as follows.

```
print ("\n [INFO] Training faces. It will take a few seconds. Wait ...")
faces,ids = getImagesAndLabels(path)
recognizer.train(faces, np.array(ids))
```

Finally, saved the trained model and printed the number of faces trained.

```
recognizer.write('/root/face_db/trainer/trainer.yml')
print("\n [INFO] {0} faces trained. Exiting Program".format(len(np.unique(ids))))
```

Figure 4.52: Train recognizer

4.4.7 Recognizing

This is the final stage of the face recognition process. Face recognition procedure is started from face detection. After that images are pre-processed and features are extracted. Then the model is trained to recognize faces. Finally, it is time to recognize faces. This recognizer returns the id or name of the student and how much confidence there to match the student's face with existing faces on the database.

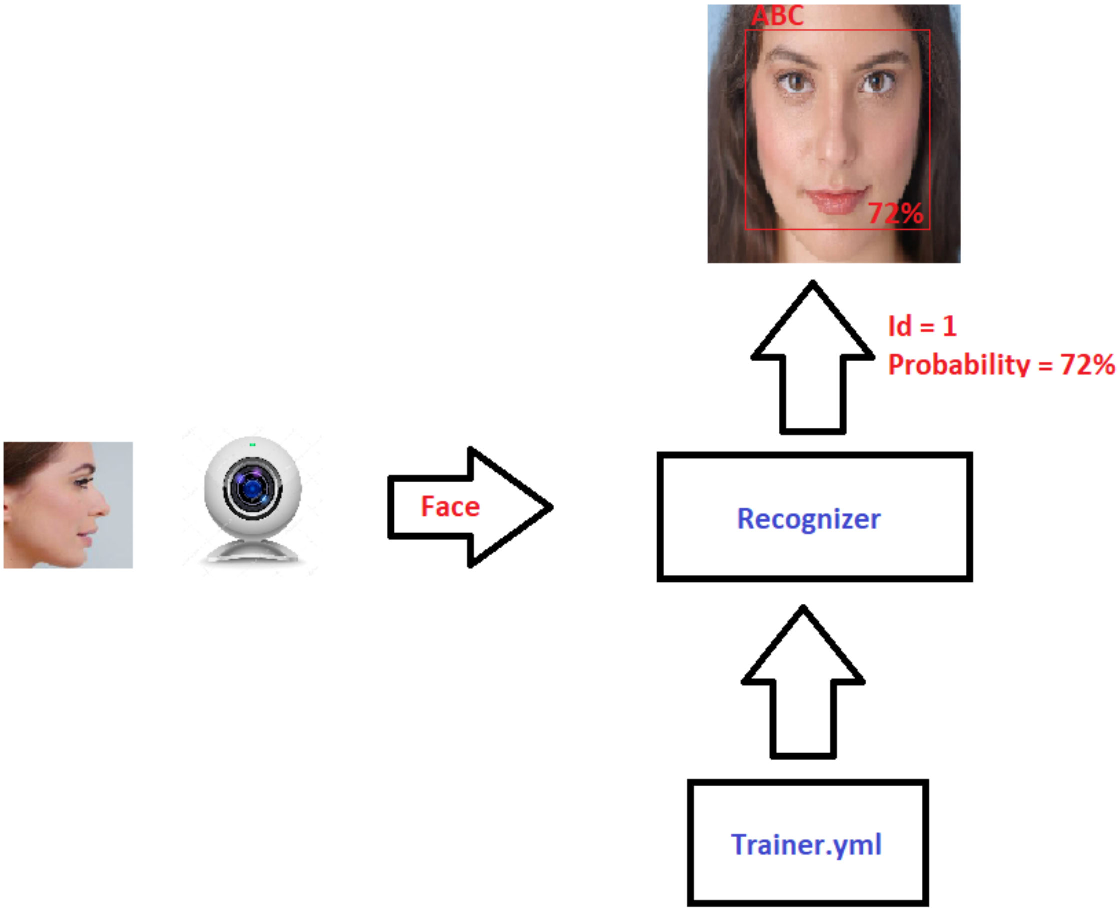

Figure 4.53: Face recognition

The python code for Recognition also begins with necessary imports, cascade and recognizer objects from Open CV. Then an array is created to store names of students which will display with an image of the student. After that turn on the camera and start detecting faces. When faces are detected, start recognizing those faces. At the time of recognition, the system finds the name of the student and percentage of confidence that match the captured image with images of the student which are saved on the database. If the confidence more than 70%, system consider that is the correct student and mark the attendance.

4.4.8 Eye blink detection

We use eye blink detection to find if user is standing in front of the camera or not. After detect the blink of eyes 1 time, the process moves to the face recognize stage. For 15 seconds after camera opened, if no blinking eyes found the system will send the error notification that the user use fake image to check attendance. The image can be photo from smart phone screen, or the picture or somebody in paper.

This code is used Facial Landmarks and Eye Aspect Ratio (EAR) to detect eye blink.

```
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

def midpoint(p1,p2):
    return int((p1.x + p2.x)/2),int((p1.y + p2.y)/2)

font = cv2.FONT_HERSHEY_SIMPLEX

def get_blinking_ratio(eye_points, facial_landmarks):
    left_point = (facial_landmarks.part(eye_points[0]).x, facial_landmarks.part(eye_points[0]).y)
    right_point = (facial_landmarks.part(eye_points[3]).x, facial_landmarks.part(eye_points[3]).y)
    hor_line = cv2.line(img, left_point, right_point,(0,255,0), 1)

    center_top = midpoint(facial_landmarks.part(eye_points[1]), facial_landmarks.part(eye_points[2]))
    center_bottom = midpoint(facial_landmarks.part(eye_points[5]), facial_landmarks.part(eye_points[4]))
    ver_line = cv2.line(img, center_top, center_bottom,(0,255,0), 1)

    #length of the line
    hor_line_length = hypot((left_point[0] - right_point[0]), (left_point[1] - right_point[1]))
    ver_line_length = hypot((center_top[0] - center_bottom[0]), (center_top[1] - center_bottom[1]))
    ratio = hor_line_length/ ver_line_length, ver_line_length
    return ratio
```

```
while True:
    ret,img = video_capture.read()


    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = detector(gray)

    for face in faces:
        #x, y = face.left(), face.top()
        #x1, y1 = face.right(), face.bottom()
        #cv2.rectangle(frame, (x,y), (x1,y1), (0,255,0), 3 )# green box, thickness of box
        landmarks = predictor(gray, face)

        left_eye_ratio,_ = get_blinking_ratio([36,37,38,39,40,41], landmarks)

        right_eye_ratio, myVerti = get_blinking_ratio([42,43,44,45,46,47], landmarks)

        blinking_ratio = (left_eye_ratio+right_eye_ratio)/2

        if(blinking_ratio >= 6):
            cv2.putText(img, "blinking", (50,50), cv2.FONT_HERSHEY_SIMPLEX, 2, (255,0,0))
            print("blinking")

            img = recognize(img,clf,faceCascade)

        elif((blinking_ratio < 6) and (delta>30)):
            print("Fake image----")
            messagebox.showerror('Error','-------Fake image-------')
            call_var()
```

Figure 4.54: Detect eye blink

4.4.9 Sending Notifications E-mails To Students

When an attendance marked for a student, there should be a way to inform it to the student. So, to ensure the attendance for a specific session is marked, an e mail notification sends to the student's mail. Therefore, if the notification not received, the student can know that the attendance is not marked. Then he/ she can get relevant action to mark the attendance.

For sending mails, smtplib python library is used. So, at the beginning smtplib is imported. After that, a server object is created using smtplib.SMTP. Then it is needed to start the connection with the server. Starttls(). After starting the connection, it is possible to login to the sender's mail and sends mails to student's mail address.

```
to = email
gmail_user = 'attendancesystembku@gmail.com'
gmail_pwd = '!attendancesystem'
smtpserver = smtplib.SMTP("smtp.gmail.com",587)
smtpserver.ehlo()
smtpserver.starttls()
smtpserver.ehlo
smtpserver.login(gmail_user, gmail_pwd)
date = datetime.datetime.now().strftime( "%d/%m/%Y %H:%M" )
header = 'To: ' + to + '\n' + 'From: ' + gmail_user + '\n' + 'Subject: Check attendance completed\n'
body = '\nWelcome ' + first_name + ',\n\nYour attendance is marked at: ' + date + '\n\nSubject: ' + subject_name +
footer = '\n\n' + day_in_week + ' From: ' +start_time + ' To: ' + end_time + ' Room: ' + room
msg = header  + body + footer
print(msg)
smtpserver.sendmail(gmail_user, to, msg)
print('sent mail\n____________________________________')
smtpserver.close()
```

Figure 4.55: Mail of completed check attendance

```
# send mail at here
to = t4.get()
gmail_user = 'attendancesystembku@gmail.com'
gmail_pwd = '!attendancesystem'
smtpserver = smtplib.SMTP("smtp.gmail.com",587)
smtpserver.ehlo()
smtpserver.starttls()
smtpserver.ehlo
smtpserver.login(gmail_user, gmail_pwd)
date = datetime.datetime.now().strftime( "%d/%m/%Y %H:%M" )
header = 'To: ' + to + '\n' + 'From: ' + gmail_user + '\n' + 'Subject: Register completed \n'
print(header)
msg = header +  '\nDear ' + t1.get() + ',\n\n' + 'Thank you for registering for attendance-sy
smtpserver.sendmail(gmail_user, to, msg)
print('sent mail')
smtpserver.close()
```

Figure 4.56: Mail of completed registration

4.4.10 Registering Students For Taking Attendance

This is the initial process that needs to do in order to take attendance of students. The student registering process is an integrated process of saving student information on the database, getting images of the student to train them and registering RFID tag for the student. The process starts when the script for registering students is running. At the beginning of the student registration process, the system asks for entering student first name, last name, e mail address and course code. The person who registering students can perform this task. After entering those details that information stored in the online database. It is required to have internet access at the registering process and it will get a small time to upload those data to the database.

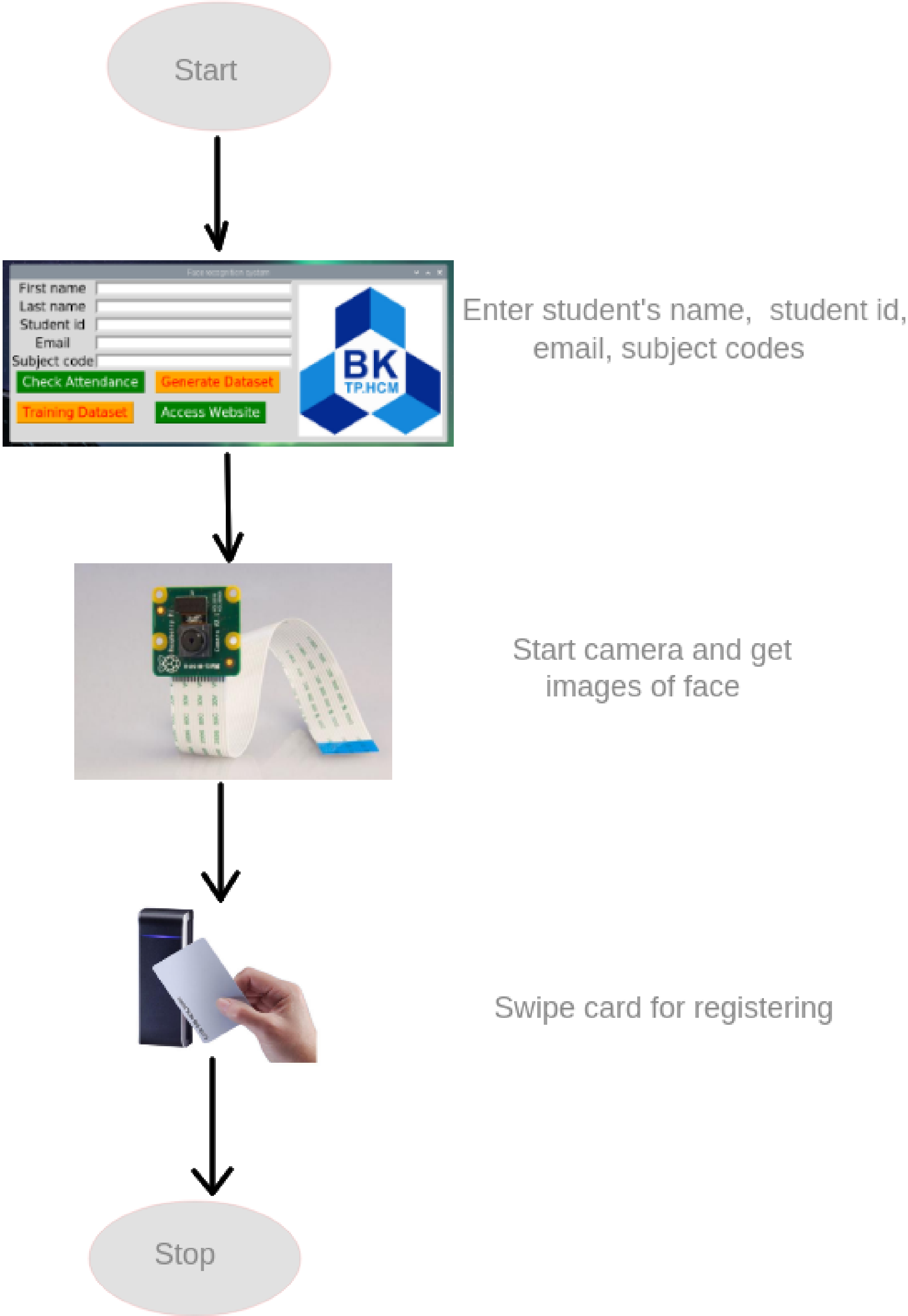

Figure 4.57: Registering students for taking attendance

When the record is created on the database, the auto generated student id is returned to the system. Then That ID used to identify student uniquely and then the system starts capturing images of the student. The camera turns on and system get 200 images of the student while student moving face. Then store those images and those are used as data to train to recognize that student. After getting 200 images, the system asks to tap RFID tag to register tag for that student. When a tag is placed, system register that tags for the student and finishes registering that student. When one student registered, the system prompt to register another student. Likewise, it is possible to register multiple students continuously. After registering all students, the system should be trained to recognize students using face recognition. This task can be done by just running one python script. The bellow diagram will further explain the student registering process step by step.

4.4.11 Marking Attendance

This is the main task of the system. This process is an integration of reading RFID tags, recognizing students using faces, mark attendance on the database and send notification e mails to students. When the system is ready to mark attendance, it is waiting for RFID tags. If a tag is tapped on the RFID reader, system fetches that student's details including student name, course, class start time and end time. At the same time, it fetches images of faces of that student and starts camera for the get video frame of the student. Then the system checks the student's face of the video frame with saved images of the student's faces. It is needed to match the student face with 50% of saved images and 15 images out of 30 saved images should be matched. This limitation is there to recognized student accurately with face recognition. If the face of the student is matched, then system check for the time of attendance for that student. This is checked with saved course details, student's presence time and student's registered course.

If all of them are matched, the system marks attendance on the database and send an e mail notification to the student. After marking attendance for one student, system waiting for mark attendance of next student. Normally the system is continuously running and waiting for student RFID tags to mark attendance. Here is the flow of marking attendance process of one student.

Marking attendance process is validated for several cases. If the RFID tag is not a registered one, the system will give an error message. If the face of the student is not matched, again it will give an error message. Although the student is with a correct RFID tag, face, if the student is in the incorrect time slot for course, again the system will give an error message because the student is there at in correct time. Following diagram will further describe how marking attendance is performing.

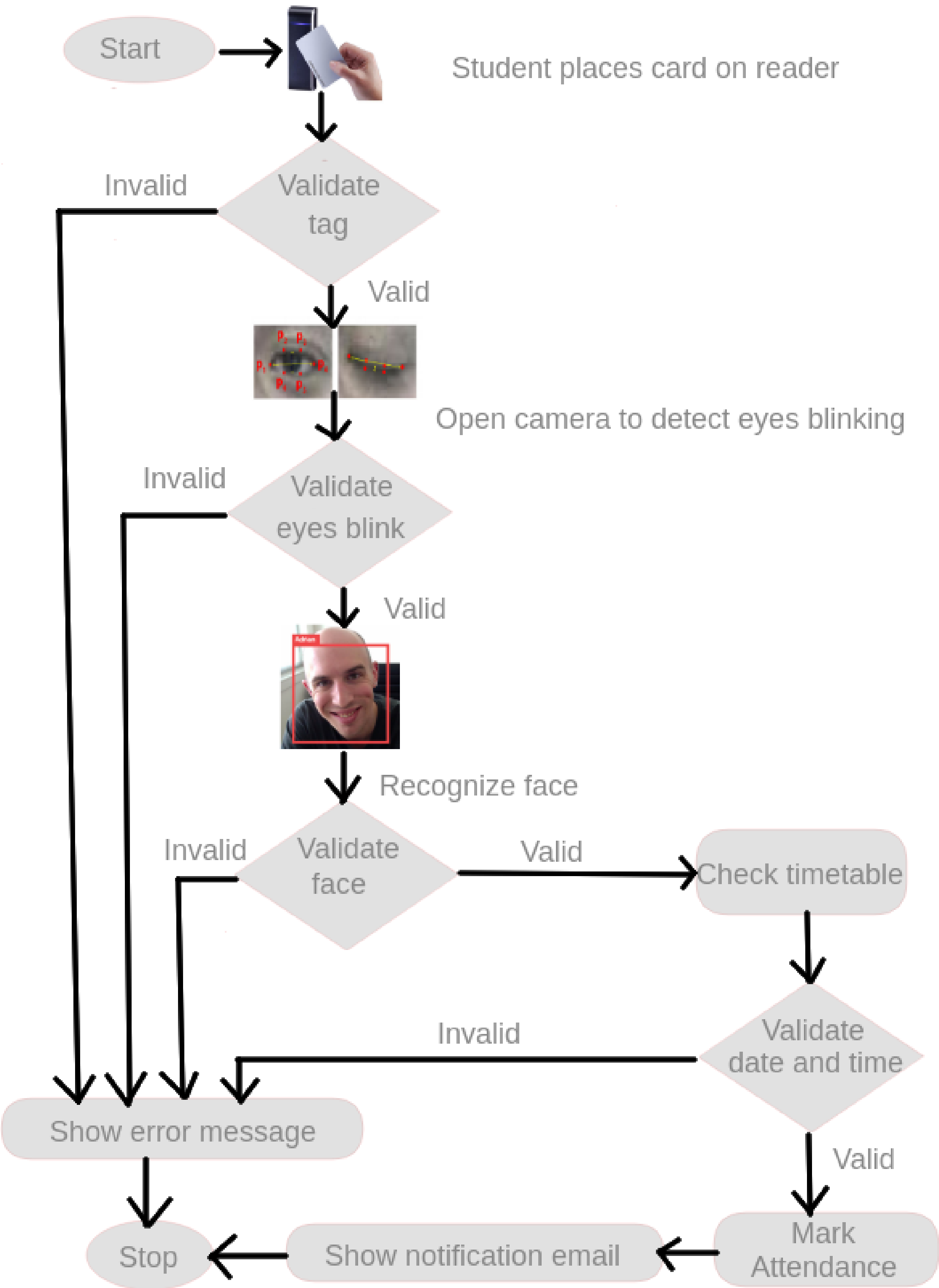

Figure 4.58: Marking attendance

4.4.12 Register Courses / Subjects

Registering subjects is performed using a web site developed for the system. Database of the system was hosted and using that database, it was possible to explore the service globally. The system is an Internet of Things based system so that by hosting the database, many advantages and opportunities can be gain. So, the system was designed and developed in the way that courses can be registered using web site and reports also can generate and view using that web site.

The web site was hosted on a free web hosting website called "000webhost". This web site gives facility to host PHP web sites that are using MySQL as a database. The course data entering to this web site is saved on hosted database and system works on that hosted database.

Following is the home page of the web site developed to the system.

The home page contains two buttons for insert data/ insert course details and views course information, student information, and student attendance details. Following is the form to register new courses on the system.

Here courses code can be a character code like IT3010 or just numeric value. Course subject is the name of the subject. Start time and end time of the subject should be entered using 24h time format. Hours and minutes can be separated using ":". That web application was developed using HTML, PHP, CSS and bootstrap. The application is developed in a way that it automatically adjusts its layout and view according to the device's screen size.

4.4.13 View Attendance Details, Students Details And Courses Details

These details can be viewed using the web application created for the system. When going to the home page of a web application, it is possible to navigate to the view information section. There at the beginning, users can access student information by providing the student's email address. Then student attendance details can be accessed by giving a first name and the email address of the student. Subject time tables also can be viewed using view class time table section.

After completing the implementation, it is necessary to test the system to ensure that the system works properly. Each and every unit and module of the system is tested at the time of the implementation. That is called "Unit testing". After implementing different units then they are integrated. After integrating each unit, integration testing is performed to ensuring that the integrated system works correctly. Finally, the entire system was tested as one unit.

Chapter 5

Evaluation and Testing

5.1 Evaluation

Application interface

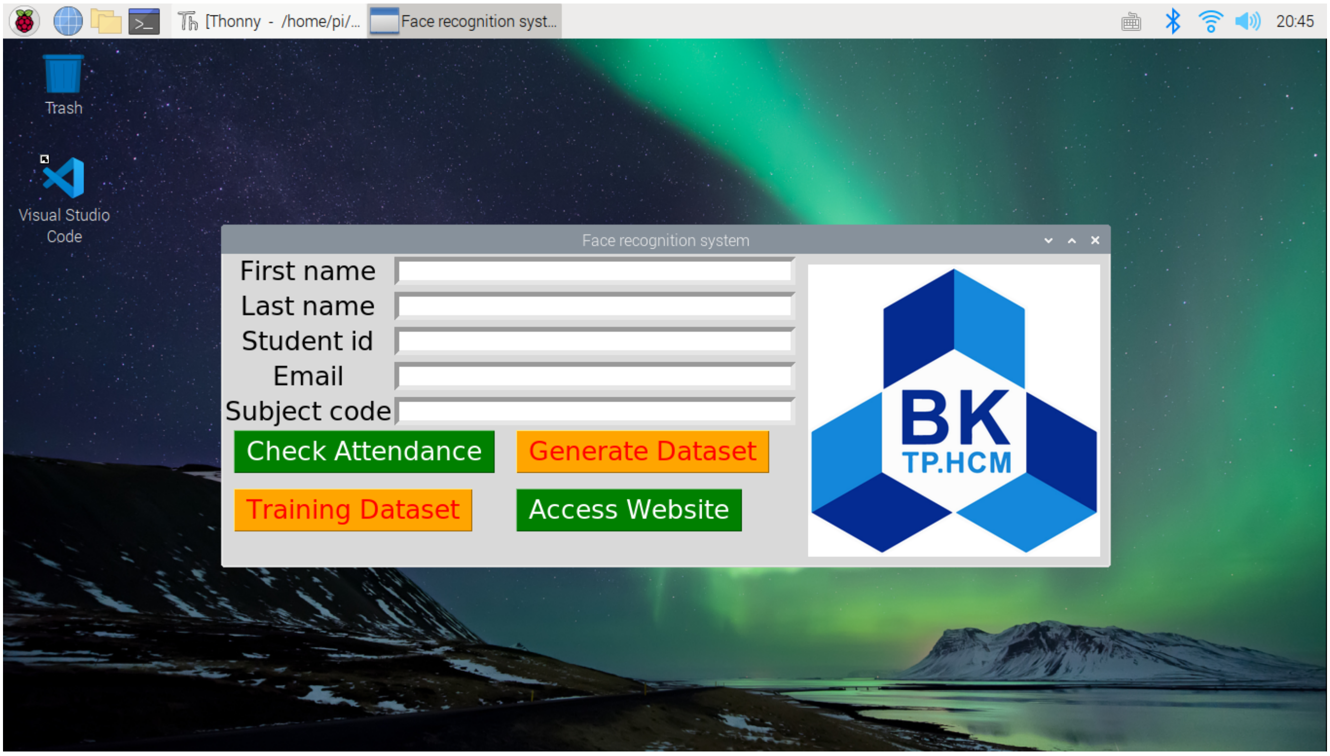

Figure 5.1: Main interface

———Danh gia hieu nang he thong, tung giai doan hoat dong–
———mat bao nhieu thoi gian. Danh gia tong quan cac chuc nang cua he thong————

5.2 Testing

5.2.1 Test-case for registration

- Dang ky luc dien day du thong tin
- Dang ky luc thieu thong tin
+ Dang ky luc thieu ma mon hoc subject number, class number

5.2.2 Test-case for checking attendance

Case 1: RFID card with correct face and correct timetable

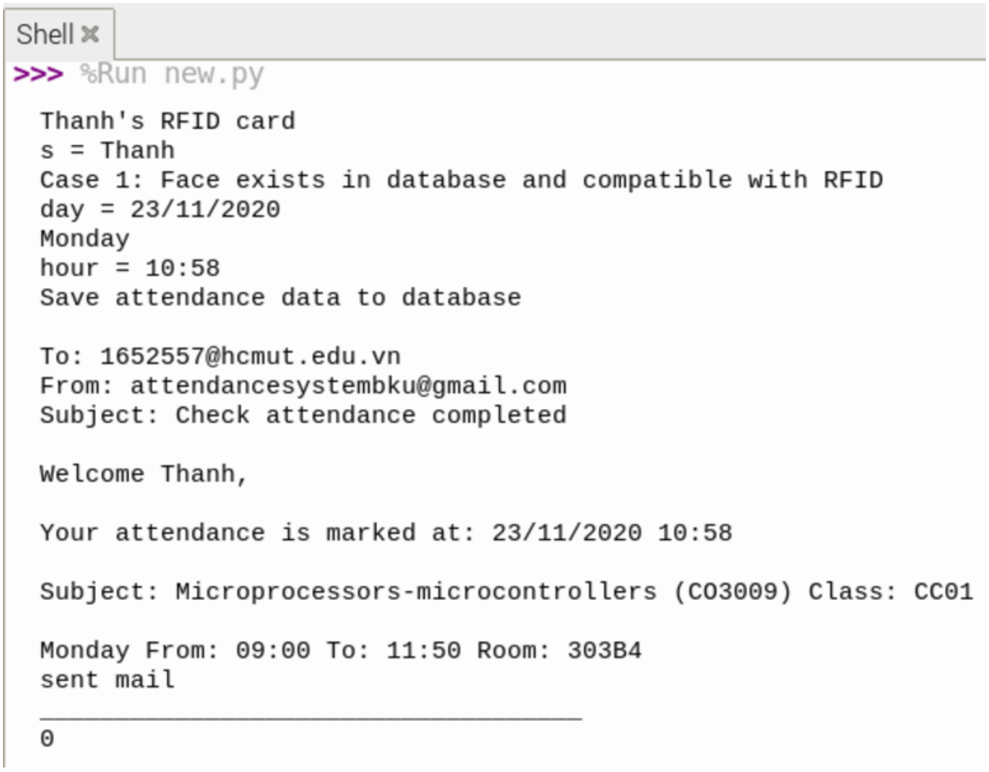

Figure 5.2: Case 1

Send email:

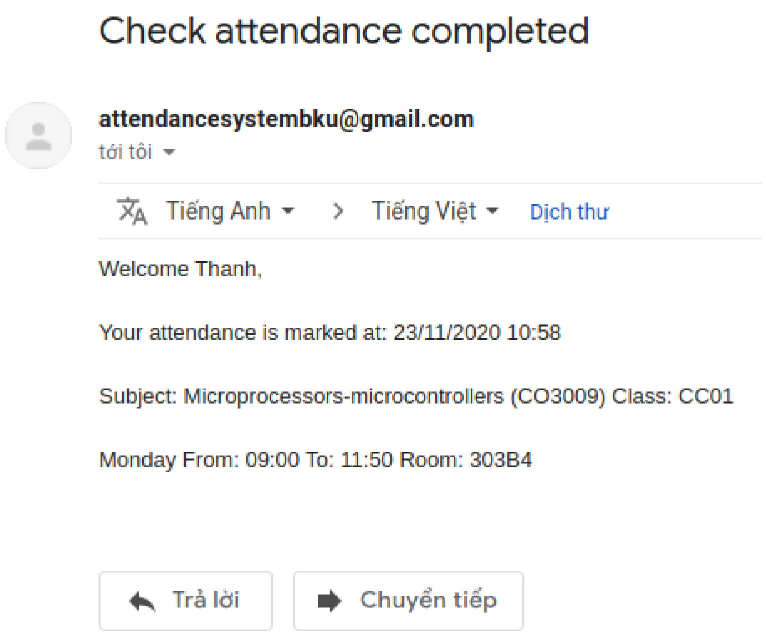

Figure 5.3: send email

Case 1.1: RFID card with correct face and no class today

Figure 5.4: Case 1.1

Case 1.2: RFID card with correct face and wrong lesson time

```
Python 3.7.3 (/usr/bin/python3)
>>> %Run new.py
  Thanh's RFID card
  s = Thanh
  Case 1: Face exists in database and compatible with RFID
  day = 23/11/2020
  Monday
  hour = 13:27
  No class available!!!
  ____________________________________
```

Figure 5.5: Case 1.2

Case 2: RFID card with other's face in database

```
>>> %Run new.py

  Thanh's RFID card
  blinking
  Case 2: Face exists in database and doesnt compatible with RFID card
  ____________________________________
```

Figure 5.6: Case 2

Case 3: RFID card with unknown face

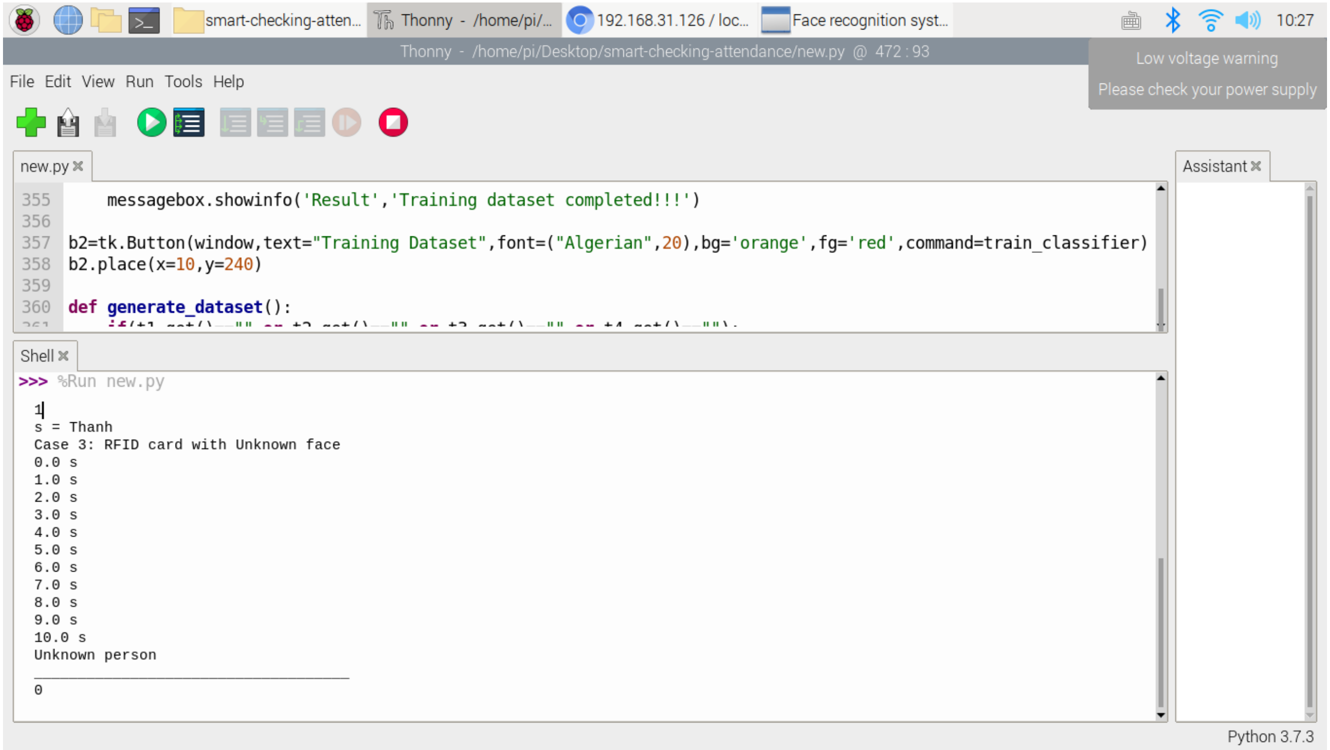

Figure 5.7: Case 3

Exception case: Fake image

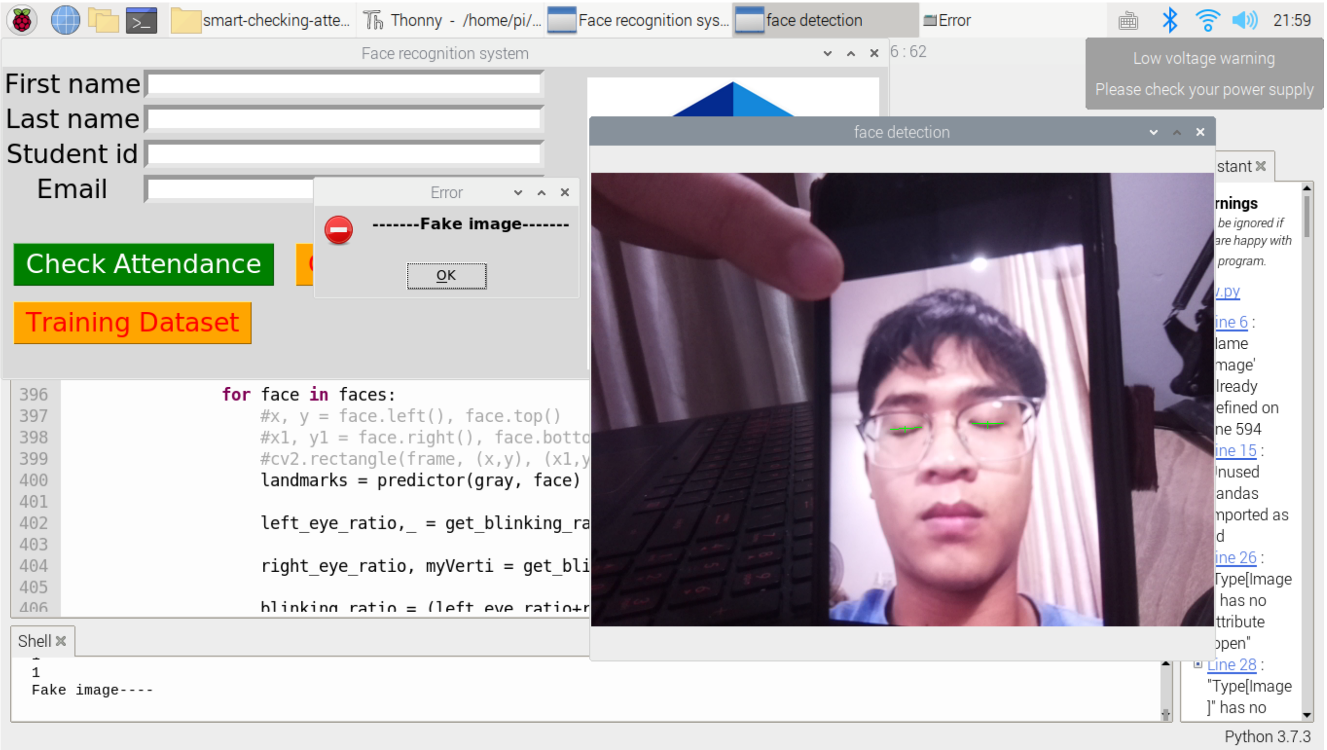

Figure 5.8: Fake image

Chapter 6

Conclusion

6.1 Conclusion

At the research stage, we choose a solution for checking attendance based on IoT is the combination between scanning student ID card (RFID) and face recognition. We use Raspberry Pi 3 with Raspbian OS to install RFID module library and OpenCV library.

6.2 Future work

In the next stage, we will complete the based python code about scanning RFID and face recognition. Improve the quality and accuracy of the face recognition system. We will solve some problems related to the detection stage when the quality of the taken pictures were not good.

We may change to other components (camera, sensors, screens...) if needed to improve our system.

We will complete the web-app and database for the system. The security of the website and the database is very important.

Bibliography

[1] Nguyen Van Phuc and Diep Minh Thien. “Giam sat va quan ly sinh vien thong qua he thong quet van tay dieu khien qua mang internet”. dissertation. Dai hoc Su pham Ky thuat, 2018.

[2] P. A. Katara, S. V. Kolhe, and A. P. Zilpe. “Attendance System Using Face Recognition and Class Monitoring System”. In: *International Journal on Recent and Innovation Trends in Computing and Communication* 5.2 (2017), pp. 273–276.

[3] Hemavati Sabu et al. “Attendance System Based on Face Recognition”. In: *International Journal of Innovative Research in Computer* 5.3 (2017), pp. 2320–9801.

[4] V. Shehu and A. Dika. “Using Real Time Computer Vision Algorithms in Automatic Attendance”. In: *IEEE Explore* 2.2 (2010), pp. 397–402.

[5] H. Rathod, Y. Ware, and S. Sane. “Automated Attendance System using Machine Learning Approach”. In: *IEEE Explore* (2017).

[6] A. K., K. S. A. P. A, and M. M. “RFID Based Student Attendance and Monitoring system”. In: *International Journal of Innovative Research in Computer and Communication Engineering* 3.Special issue 1 (2015), pp. 2320–9801.

[7] Z. Jiang. “Analysis of Student Activities Trajectory and Design of Attendance”. In: *IEEE Explore* (2016), pp. 600–603.

[8] D. K. Sarker, N. I. Hossain, and I. A. Jamil. “Design and Implementation of a smart attendance management system using multiple step authentication”. In: *IEEE Explore* 3.Special issue 1 (2016), pp. 91–95.

[9] Z. I. Nur, K. A. Sidek, and T. S. Gunawan. “Design and Development of Portable Classroom Attendance System”. In: *IEEE Explore* (2015).

[10] K. K. D. B.M Sri Madhu. “IoT based Automatic Attendance Management System”. In: (2017).

[11] *What is IoT? – A Simple Explanation of the Internet of Things*. iotforall. 2020. URL: `https://www.iotforall.com/what-is-iot-simple-explanation/` (visited on 05/05/2020).

[12] *What is IoT (Internet of Things)? IoT Architecture Explained*. edureka. 2019. URL: `https://www.edureka.co/blog/what-is-iot/` (visited on 05/05/2020).

[13] *Internet of Things Security Risks*. Microsoft. Dec. 9, 2019. URL: `https://www.avast.com/c-iot-security-risks` (visited on 05/09/2020).

[14] *The Internet of Things How the Next Evolution of the Internet Is Changing Everything*. Cisco. 2011. URL: `https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf` (visited on 05/09/2020).

[15] Lyman Chapin Karen Rose Scott Eldridge. *The Internet of Things: An Overview Understanding the Issues and Challenges of a More Connected World.* The Internet Society (ISOC). 2015. URL: https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf (visited on 05/09/2020).

[16] Pradeep Kumar Mallickb Nallapaneni Manoj Kumara. *Blockchain technol-ogy for security issues and challenges in IoT*. nternational Conference onComputational Intelligence and Data Science (ICCIDS 2018).

[17] Lu Yan et al. *The Internet of Things From RFID to The Next Pervasive Networked Systems*. Auerbach Publications, 2008.

[18] Bill Glover and Himanshu Bhatt. *RFID Essentials*. O'Reilly, 2006.

[19] Mehdia Ajana El Khaddar, Mohammed Boulmalf, and Hamid Harroud. *RFID Middleware Design and Architecture*. Intechopen. 2011. URL: https://www.intechopen.com/books/designing-and-deploying-rfid-applications/rfid-middleware-design-and-architecture (visited on 05/17/2020).

[20] Ph.D. Mubarak Shah. "Face Detection and Gesture Recognition for Human-Computer Interaction". In: *University of Central Florida Orlando, USA* (2001).

[21] *Introduction to OpenCV-Python Tutorials*. OpenCV. 2020. URL: Introduction%20to%20OpenCV-Python%20Tutorials (visited on 05/16/2020).

[22] Tereza Soukupova and Jan Cech. "Real-Time Eye Blink Detection using Facial Landmarks". In: *Faculty of Electrical Engineering, Czech Technical University in Prague* (2016). URL: http://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf.