**HANOI UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**

****

**Instructor: MSc. Bui Quoc Khanh**

Faculty of Information Technology, Hanoi University

**FACE DETECTION FOR**

**CHECKING ATTENDANCE REPORT**

**Students: Nguyễn Duy Hải**

**Hoàng Thanh Nga**

**Phạm Hồng Ngọc**

**Nguyễn Thu Trang**

***Abstract***

***Currently, in most schools in general and universities in particular, the traditional attendance is still an inevitable and regular. When it comes to taking attendance at university, we just think that attendance is simply the lecturer calling out the student's name with a printed list on a computer or on paper ... without regard to applying new technology to attendance. The fact has shown that the traditional attendance has caused many shortcomings and difficulties such as time consuming, students ask someone to take attendance or go to school for them. Not only that, the missing attendance list is also a major difficulty for traditional attendance. So in this subject, we develop the application of facial recognition to student attendance, data set description, libraries used, methods, algorithms used to develop projects. The biggest purpose of our project is to develop an application for attendance thanks to facial recognition, which helps to control class sizes faster and more modern, in addition to having a secondary function of emotion detection.***

1. **INTRODUCTION**

## **Overview**

Image processing is one of the increasingly popular areas in social life. With the rapid development of modern machines such as digital cameras, digital camcorders, computers, smart phones, the amount of information that humans can obtain in the form of images is quite large. Not only dealing with smudges, recycling and restoring old photos, today image processing techniques have brought about great advances such as face detection and facial recognition, the type of object when it is combined with the field of artificial intelligence. One of the most interesting human recognition problems today is facial recognition.

One of the most interesting problems in the field of image processing today is Face Recognition. Although this is a very interesting topic, the problem of face recognition is always a difficult one, posing many challenges and difficulties for researchers. As we all know, the face plays an important role in the communication process between people, it carries a rich amount of information, such as from the face we can determine gender, age, race, emotional state, especially determining the relationship with the object (known or not). Therefore, the problem of facial recognition plays an important role in many areas of people's daily lives such as monitoring systems, access management, information search for a celebrity, etc. security, confidentiality. There are many face recognition methods to improve performance, but more or less these methods are facing challenges in terms of brightness, tilt orientation, image size, or the influence of environmental parameters. Face recognition has been studied since 1980, is a research area in Computer Vision, and is also considered a field of research in Biometrics (Fingerprint recognition, or Iris recognition). While fingerprint and iris recognition can be widely practical in practice, facial recognition remains challenging. Compared with fingerprint and iris recognition, facial recognition has a richer data source and requires less controlled interaction. The problem of recognizing human faces still has many challenges, so every year, at home and abroad, there are still many studies on human facial recognition methods. There are two popular recognition methods today, which are feature-based recognition of face elements such as Gabor Wavelet transform and Neural network, SVM, etc., and recognition based on the whole face as a method: PCA, LDA, LFA. With the Principle Component Analysis algorithm (PCA), although it can give relatively good results, this techniques has the disadvantage of being computationally expensive and complex with the increase database size, since all the pixels in the image are necessary to obtain the representation used to match the input image with all others in the database [1]. While with Gabor Wavelet transform, despite being a good method with relatively high accuracy, the face images rendered by it are convolved with a bank of Gabor filters and hence the dimensionality of the Gabor feature space is overwhelmingly large, yielding high complexity [2]. Many down sampling techniques have been proposed in order to reduce the dimensions by utilizing select feature points[3]; their output still consists of a large number of high dimensions of feature matrix leading to the possibility of partial loss of feature discriminative information and marked reduction of accuracy in the classification stage.

In this project, we explored how to detect facial features using machine learning through using ML Kit's facial recognition algorithm. Firebase ML Kit provides the ability to test and use machine learning without knowing much about it and without spending time and effort in building ML models. Using the Face Recognition API, you can easily build an Android Things app that detects facial features.

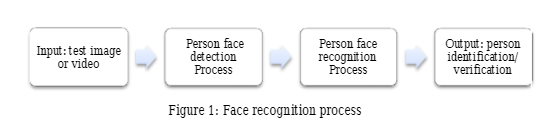
## **Purpose**

This report, the result of our project, aims to gain more understanding of image processing technology through the development of an Android application with functions: face detection, face recognition to check attendance and emotion recognition. Not only that, the application is also expected to have a nice, friendly and easy-to-use interface. Through the development of this application, we hope that student attendance checking and class management will become faster and easier.

1. **METHODOLOGY**

The Face Recognition problem includes different problems such as face detection, facial landmarking, feature extraction, labeling, and classification. To build this face recognition application, we use the machine learning algorithm of the Google Firebase ML Kit libraries combined with database SQLite. Problems to be solved:

* Firstly, detect all faces in image or in video
* Secondly, extract the features of the face that you use to distinguish it from other faces: how big is the face, how long is the face, how high is the nose, how long is the eye, mouth, ear....
* Third, enter name and student id for each feature extracted face
* Then, compare those features of the newly added face with those you already know and identify that person's name
* Finally, after matching and identifying the right object, automatically save the information and update the list into the system with name, id and date.



## **Dataset**

Raw data in our application includes the faces of the students in the picture or the video stream. We conduct image processing to obtain a dataset that is the features for each detected face. The data includes landmarks, length, height of the whole face and each part of the face: left eye, right eye, nose, left mouth, right mouth, left ear, right ear, name and student id assigned to each face. We will use the dataset to build a recognition application with the machine learning library of Firebase ML Kit and the mobile Vision API of ML Kit. Our goal is training a model of ML Kit to detect all faces in the video stream of pictures and identify whose face this is.

## **2. Technology**

### ***2.1.* Firebase ML Kit for Android**

Our application is deployed in the Java language. To make face recognition easy and fast, we use the machine learning algorithm of the Google Firebase. Firebase ML Kit is a mobile SDK that helps us experiment with machine learning technologies. Firebase ML Kit is an attempt by Google to make machine learning easier to use and accessible to people who don't know much about machine learning technology, providing pre-trained models that can be used for development of Android and Android Things apps. In this Android project, we use the camera to scan all faces appearing in the video stream. Furthermore, once a face is detected, we can detect facial features such as face rotation, size, etc. Furthermore, by using the facial recognition API, our application can go deeper in the process of retrieving this facial analysis:

* Identify and locate facial features: Get the coordinates of the eyes, ears, cheeks, nose and mouth of every detected face.
* Get the contours of facial features: Get the contours of detected faces and their eyes, eyebrows, lips, and nose.
* Recognize facial expressions: Determine whether a person is smiling or has their eyes closed.
* Track faces across video frames: Get an identifier for each individual person's face that is detected. This identifier is consistent across invocations, so you can, for example, perform image manipulation on a particular person in a video stream.
* Process video frames in real time: Face detection is performed on the device, and is fast enough to be used in real-time applications, such as video manipulation.

### ***2.2* . Database Browser SQLite:**

******

In our project, all face data including faces, student name, student id will be stored in SQLite.

SQLite is a compact, complete, relational database system that can be installed inside other applications. SQLite is a popular choice because of embedded database software for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as today it is used by a number of browsers, operating systems, and widely embedded systems (such as mobile phones). SQLite has bindings to many programming languages.

SQLite stores the entire database (definitions, tables, indexes, and the data itself) as a single cross-platform file on a single server. It implements this simple design by locking the entire database file while writing. SQLite read operations can be multitasked, although writes can only be performed sequentially.

1. **IMPLEMENTATION**

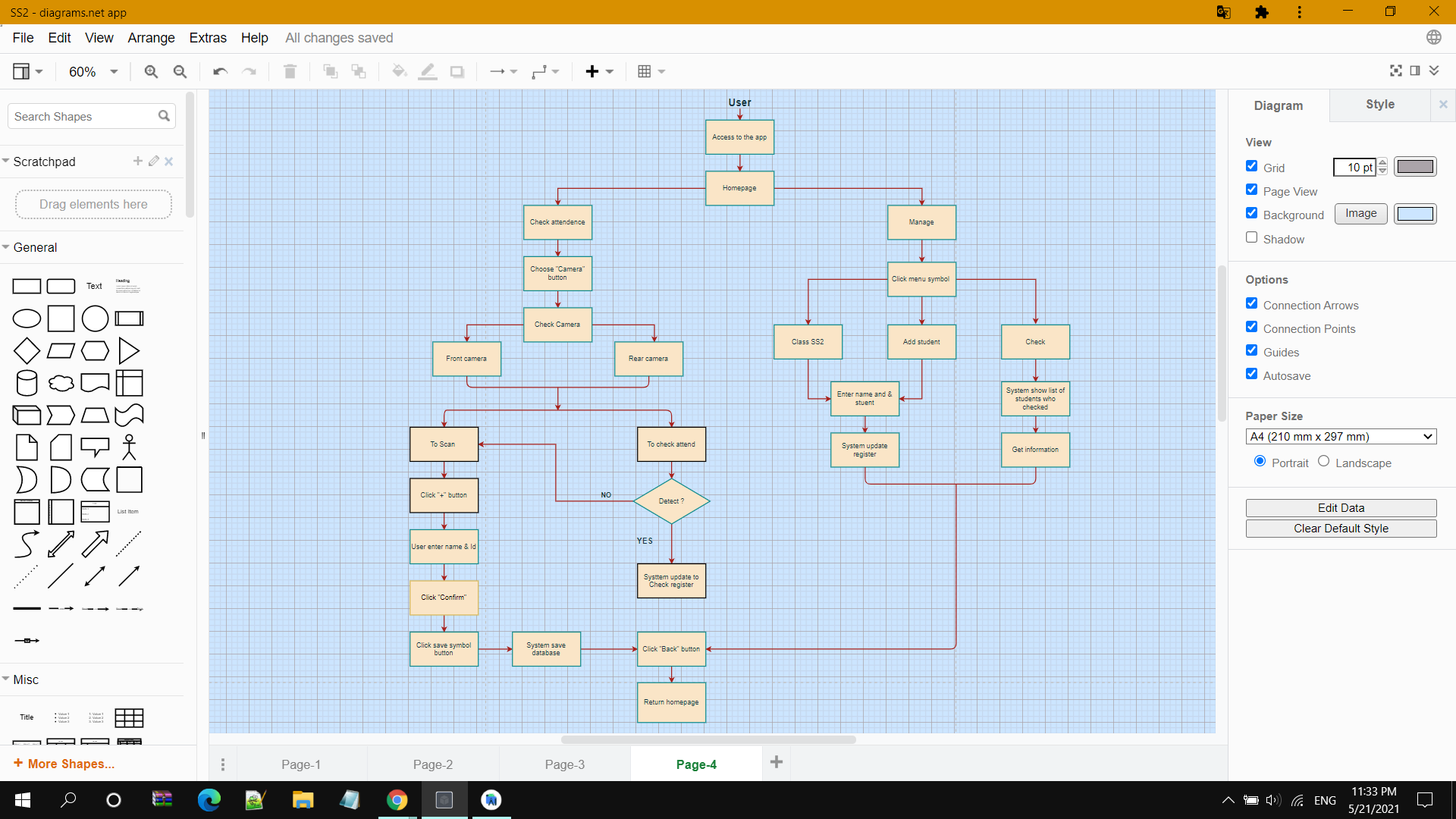
**1. Work Breakdown**

My group consists of four members and the work was divided equally for each member to ensure both progress of the project and chances for every member to contribute to the project. The work breakdown can be seen in the following table:

|  |  |  |
| --- | --- | --- |
| **#** | **Member** | **Task** |
| **1** | Hoang Thanh Nga | database, code, report |
| **2** | Nguyen Thu Trang | Find research paper, code, test, report |
| **3** | Nguyen Duy Hai | Find research paper, code, test, report |
| **4** | Pham Hong Ngoc | database, code, report |

All members took part in the implementation of the application. The detailed contribution of members during the implementation phase is given in the Implementation section of this report document.

## **2. Deployment**



With the first screen we use a splash screen when the app is open with a delay of 2,5 second. Then, the user selects the “Manage” button, in the menu navigation, choose “ Add students”, enter the name and id of the student we want, confirm this information. The system will automatically update to Class SS2.

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
|  |  |

Back to the homepage, we select the “Check attendance” button, wait for camera permission, and the application will detect if the frame has a human face by drawing a border around it. Users click the “+” button and enter name with id, then click “Confirm” for scan to save data of input face..

After entering information of students, we will start to recognize, by choosing the “Camera” button, point the camera at the face we need to recognize, the application willconduct facial recognition processing. If it recognize objects that have information in database, it will save attendance and write name student in “Check” with the specific date

|  |  |  |
| --- | --- | --- |
|  |  |  |

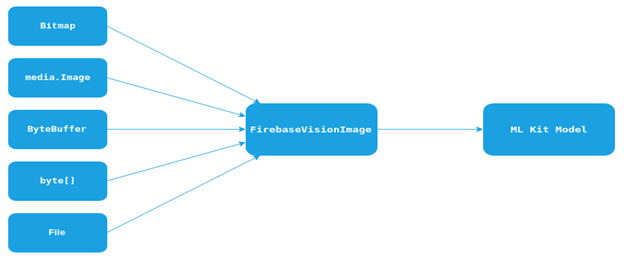
This application will return more accurate results if the user scans students from multiple angles.

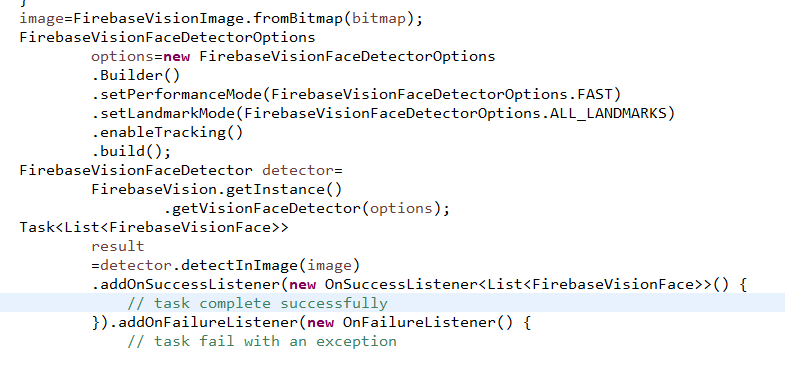
## **Data processing**

The data for a face recognition system is divided into two sets: the training set and the recognition set (probe set or test set). The training set consists of faces with known identities. In order for accurately detecting and identifying faces, for input pictures, we require the input face to match the rectangular contour around the face to get the 200x200 pixel image.

In terms of data preparation for image dataset, preprocessing steps firstly include converting images in the video stream to bitmap format. Then, we create an object named FirebaseVisionFace object. Each FirebaseVisionFace represents a detected face and contains all the information associated with it. One FirebaseVisionFace includes the basic structures of the face: landmarks and position, length, height of the overall face in general, each part of the face in particular of each subject is detected: left eye, right eye, left ear, right ear, nose, left mouth, right mouth…

We use the ML Vision Face Model of ML Kit to train data. The processing will be handled by the FireBase Library and will pass its output to the listener function automatically under the hood. When another face is included for recognition, we get the detection result from the embedded AI Model of MLKit, a Listener will be triggered with an Object of FirebaseVisionFace, where it will consist of detection results for each Faces that has been detected.



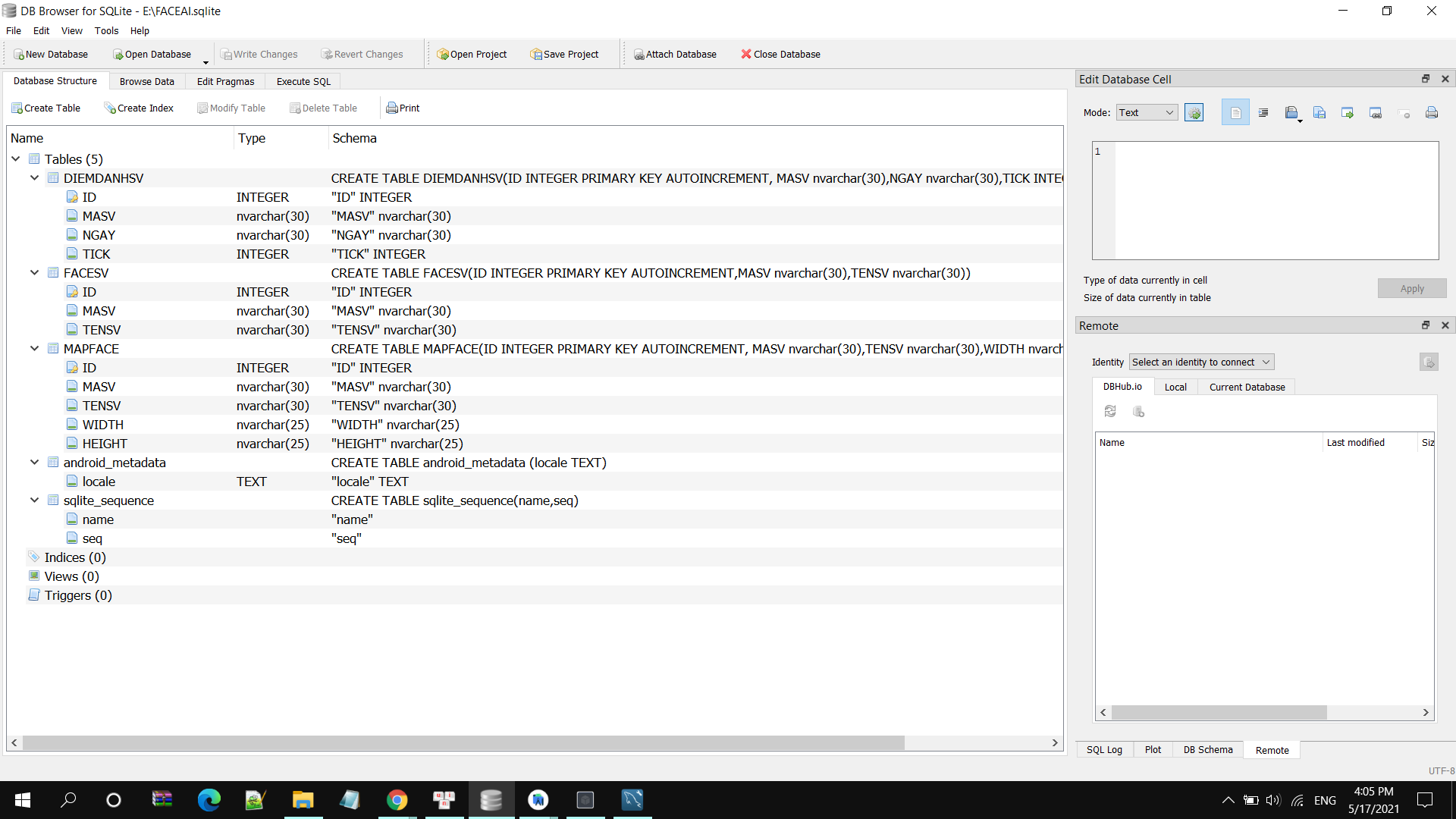


## **4. Database design**

Database design : Layered Architecture Style. In this setup, you have the presentation or GUI tier, the data layer, and the application logic tier.

* The application logic tier. The application logic tier is where all of the "thinking" takes place, and it knows what the application allows and what it can do, as well as making other decisions. This logic tier also writes and reads data to and from the data tier.
* The data tier. All of the data used in your application is stored in the data tier. This tier allows you to safely store data, perform transactions, and even browse through massive amounts of data in a matter of seconds.
* The presentation tier. The user interface is the presentation tier. This is the interface through which the app user communicates. This is where they input the necessary data. This tier also serves as a conduit between the data and the user, relaying the user's various behaviors to the logic tier.
* The presentation tier is your PC, Tablet, Mobile, etc. All of the actions in the presentation tier will be passed on to the logic tier after your action . The logic tier would be run on a Web server.In data tier, the database is SQLite database.

Here is tables database that created for my application :

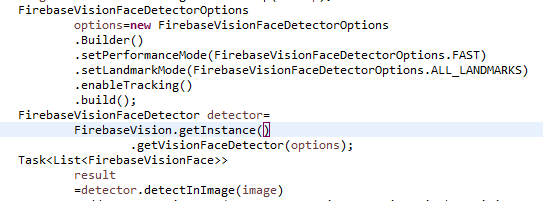


## **5. Functions**

### ***5.1. Face detection and face tracking***

Our application can detect multiple faces and track faces in video sequences.

To perform the face detection, first we have to create a camera and configure the face detector. We create an instance of FirebaseVisionFaceDetector, it helps you to detect the faces in a provided FirebaseVisionImage. Using this we can configure a few different properties for the recognition process. So in our scanning process, the preview image or from the selected image from the gallery is converted as FirebaseVisionImage, it creates a FirebaseVisionImage from the bitmap, the FirebaseVisionFaceDetector instance to detect the faces in our image. This can be done by calling the detectInImage() function, passing in our FirebaseVisionImage instance. If the recognition operation succeeds, it returns a list of FirebaseVisionFace objects in the success listener.



To decode the face detection results, we would iterate through the list, and process each detected face independently. Each FirebaseVisionFace object represents a face that was detected in the image. From this object, we can get the bounding rectangle coordinates in the source image and also the landmarks and the contours from the face.

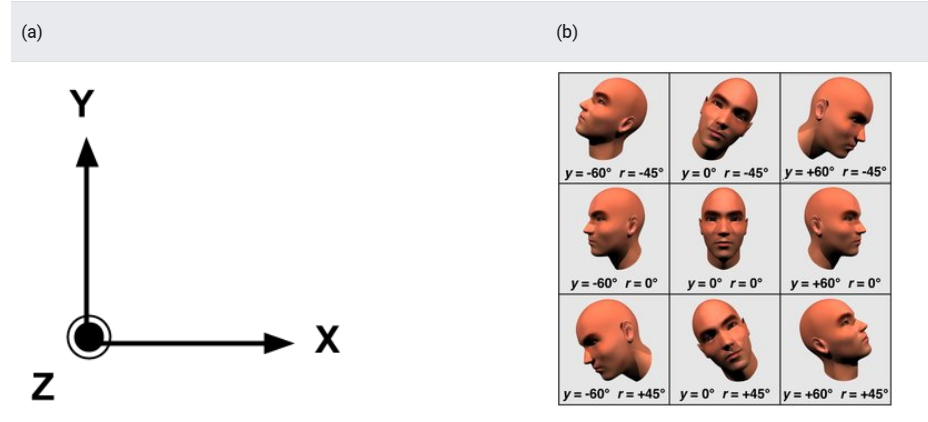
|  |  |
| --- | --- |
|  |  |

5.1.a 5.1.b

### ***5.2. Face recognition***

Face recognition automatically determines if two faces are likely to correspond to the same person. For each detected face, the user will enter information for the corresponding face including full name and student id.

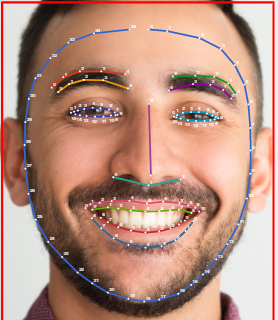
First, for face recognition, each subject will have to scan its face 10 times, including different angles to extract features.



(a) The coordinate system with the image in the XY plane and the Z axis coming out of the figure. (b) Pose angle examples where yEuler Y, rEuler Z

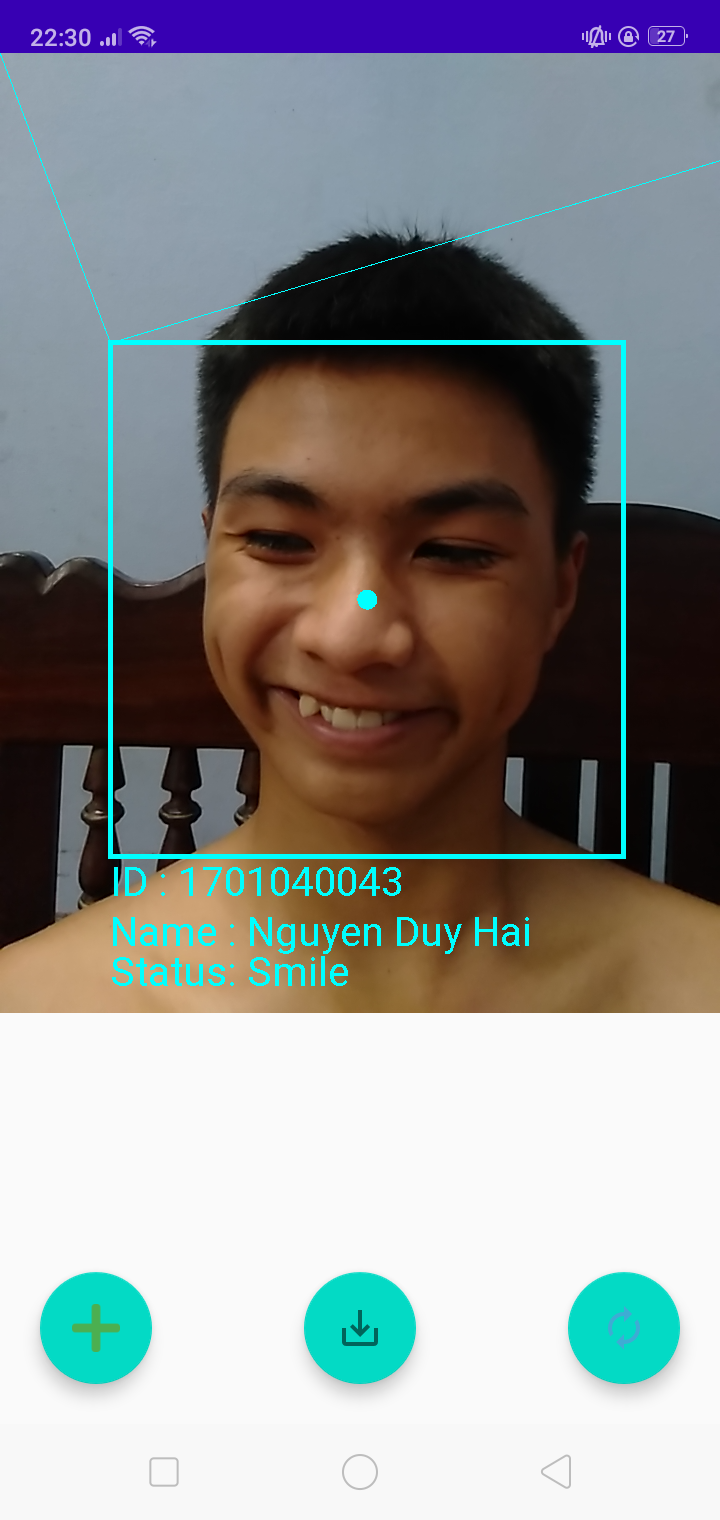
Specific feature extraction is marking contours around the face, locating, marking and calculating the length and height of the overall face in general and each part of the face in particular for each object detected: left eye, right eye, left ear, right ear, nose, left mouth, right mouth...

Each encrypted face will be assigned a student name and id through the information input step. All this information will be saved in the database. This extraction is a combination of using the Firebase ML Kit API, drawing contours with canvas, and combining operations to calculate the length. Then we put all the collected data into the model of ML Kit for training. Results returned:





The next step is the most difficult step, which is recognition. The simplest approach to face recognition is to directly compare the unknown face with all the images we have of people that have been tagged. When we find a previously tagged face that looks very similar to our unknown face, it must be the same person. When a new object is detected, the ML Kit API will query the database to find the most suitable face and display the name with the student id of that face on the screen.



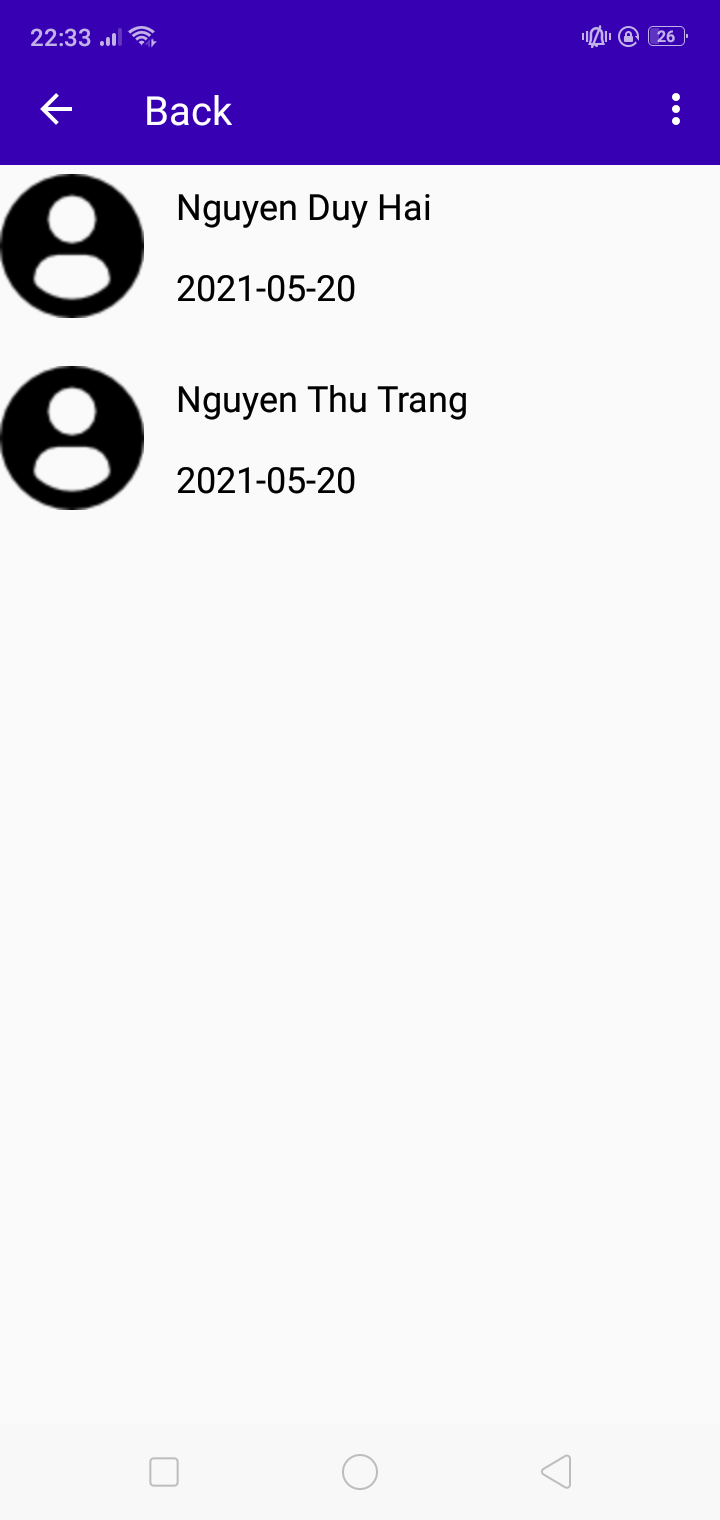
5.2.a 5.2.b

|  |  |
| --- | --- |
|  |  |

### ***5.3. Checking attendance***

After matching and identifying the correct object, the application will automatically save the information and update the list in the system with the name, id and date. To get information about list checked students, the teacher clicks the menu symbol in the top right corner and clicks the “Check” button. In this part, to display a list of students, we use recyclerview with 3 main parts: model ( Helps in positioning the objects), activities

( helps with animating the items for common operations such as Addition or Removal of item) and adapter(to handle the data collection and bind it to view) . We implemented Linear layout manager layouts to show student in a vertical

**

### ***5.4. Emotion recognition***

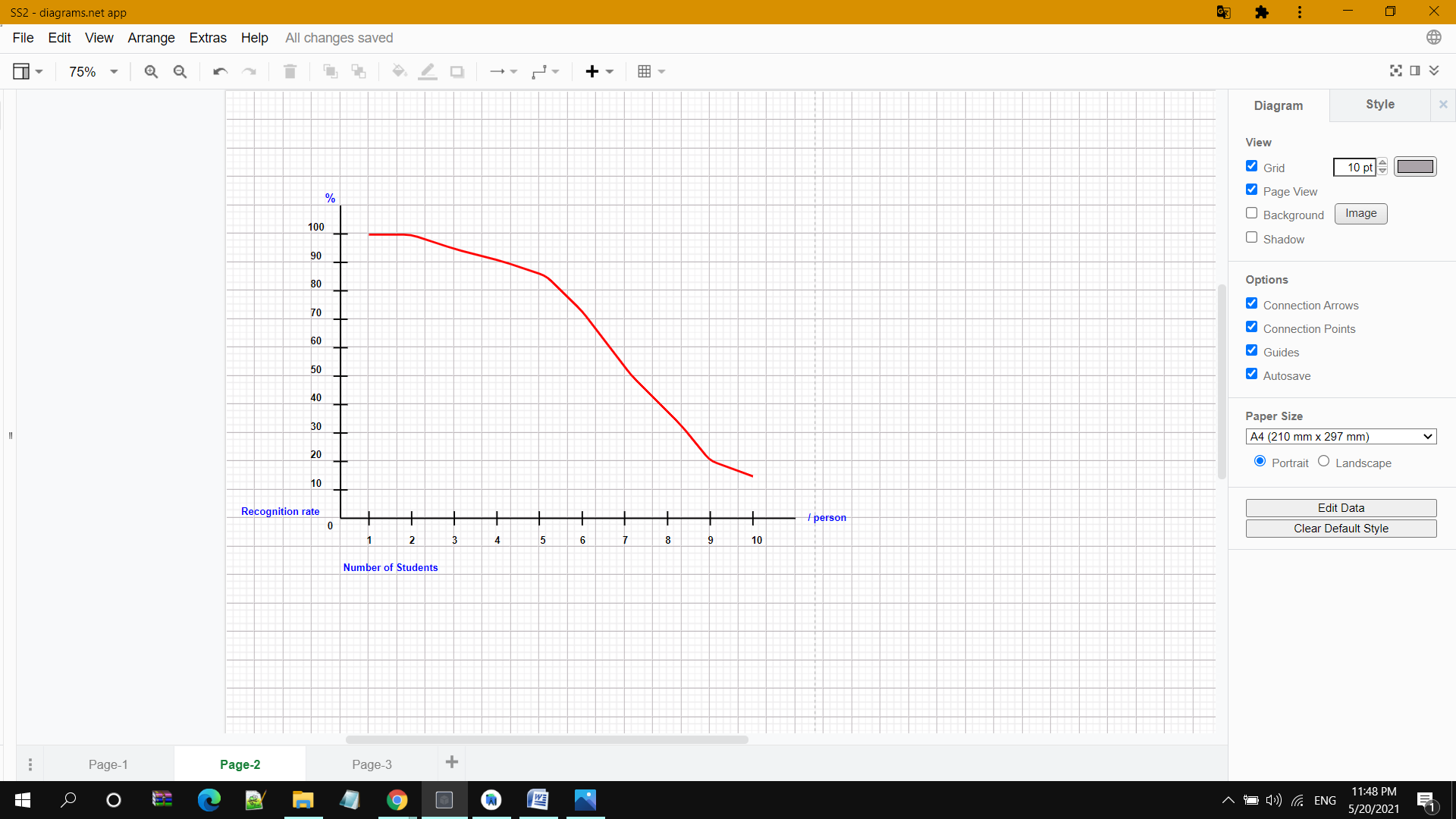
Our team has developed a small function that detects emotions when objects smile and are normal.

|  |  |
| --- | --- |
|  |  |

5.4.a 5.4.b

# **EVALUATION**

## **Results**

****

After testing this application 10 times, each time with 10 students, we got the following results:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No students**  **(/person )** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **Recognition rate (%)** | 100 | 99.99 | 95, 01 | 91,03 | 85.82 | 75.02 | 53.12 | 38,35 | 21,75 | 15, 02 |

Our mobile application is easy for the user to become familiar with and competent in using the user interface on the first contact because the application has a friendly customer interface as well as view the web clearly. Accuracy detection is up to over 70% and only takes 10-15 seconds to give a result. Coupled with using SQLite, our app does not limit objects so it is not a problem to check a large number of students . The greatest achievement is the app has enough main functions: check attendance, manage and have an extra function is emotion recognition. Users can flexibly use the front or rear camera. Can swap width and height sizes when in portrait, since it will be rotated 90 degrees

## **2. Limits**

The problem of human face recognition is a problem that has been studied since the 70s. However, this is a difficult problem so current studies have not yet achieved the desired results. Therefore, this issue is still being studied by many groups around the world. After researching and conducting this project, we found that the difficulty of the human face recognition problem can be mentioned as follows:

* Shooting position, shooting angle: Face shots can vary a lot because of the angle between the camera and the face. For example: straight shot, 45o left diagonal shot or 45o right diagonal shot, shot from the top down, shot from the bottom up, etc. With different poses, facial components such as eyes, nose, mouth may be partially or even completely hidden.
* Presence or absence of certain components of the face: Features such as mustaches, beards, glasses, etc. may or may not be present. This problem makes the problem even more difficult.
* Facial Expression: The expression of a human face can have a significant influence on facial parameters. For example, the same person's face, but may be very different when they smile or get scared, etc.
* Change of face before and after saving to database
* Less resolution: Resolution of image may be very low, which is also difficult for detection.
* Lightning effects: Lighting effects may not be similar in the image. Some part of the image may have very high contrast and others may have very low contrast.
* Distance: Far distance between camera and human face may low the detection rate of human faces in image.

However, the biggest challenge we are facing is the limit on the number of students because if we check more students, the app needs more time to load the database as well as confusion that leads to wrong results. It’s almost absolutely within 5 objects. The following objects may have some confusion

## **3. Future improvement**

We can surely say that the majority of the current face recognition algorithms are not optimal in different conditions. This is the fact that there is no study or project that can be absolutely accurate of the recognition programs. The next generation of facial recognition systems should recognize people in real time and in circumstances far less limited.

The potential developments of this subject are countless; the technology for facial recognition can be used in the examination room, where just by photographing the exam room, the teachers will know exactly who is present or absent. Before we do that we will try to update the app to be able detect more faces and show the specific time, not just update. In Particular, studying more from successful applications to improve accuracy is awful vitaly.

# **CONCLUSION**

To create this application, each member has worked so hard. Initially, we planned to build a face recognition website, then realized that it may be a difficult challenge to study and design. Therefore, after discussion, we decided to change it to an app that runs on android devices. During the working process, there were many disagreements and quarrels between the members. In particular, we felt a lot of pressure when the source code kept crashing. However, thanks to the efforts of each member, in the end, we have builded this application with the essential functions that are planned.

We have created this app with great potential and it can be easily integrated in the future. The application will help teachers and students save time for managing the attendance of students, teachers can easily manage students with just a few touches. Furthermore, our project is created with the aim of meeting the needs of the general consumer to the greatest extent possible. We hope that you will see all the features and benefits that our application has to offer.

## **References**

[1] N.Bhaskar, Kapili Maheswari Das, **“**A Survey on Face Recognition Techniques,) June 2018. Available at: <https://www.academia.edu/36982315/A_Survey_on_Face_Recognition_Techniques>.

[2] A. Vinay, Vinay.S. Shekhar, K.N. Balasubramanya Murthy, S. Natarajan, Face Recognition Using Gabor Wavelet Features with PCA and KPCA - A Comparative Study, Procedia Computer Science, Volume 57, 2015, Pages 650-659, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2015.07.434>. Available: <https://www.sciencedirect.com/science/article/pii/S1877050915019638>.

[3] L. Nanni and D. Maio, “Weighted sub-Gabor for face recognition,” Pattern Recognition Letters, vol. 28, no. 4, pp. 487-492, 2007. Available at: https://dl.acm.org/doi/10.1016/j.patrec.2006.09.002.