TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

A Minor Project Report
On
SMART ATTENDANCE SYSTEM

Submitted By:

Nikesh DC / 075BCT052

Ravi Pandey / 075BCT065

Rohan Chhetry / 075BCT066

Yukta Bansal / 075BCT096

Submitted To:

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

LALITPUR, NEPAL

(March 21, 2022)

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

A Minor Project Report
On
SMART ATTENDANCE SYSTEM

Submitted By:

Nikesh DC / 075BCT052

Ravi Pandey / 075BCT065

Rohan Chhetry / 075BCT066

Yukta Bansal / 075BCT096

Submitted To:

Department of Electronics and Computer Engineering

Pulchowk Campus

Kathmandu, Nepal

In partial fulfillment for the award of the Bachelor's Degree Computer Engineering.

Under the Supervision of

Er. Lok Nath Regmi

March 21, 2022

DECLARATION

We hereby declare that the report of the project entitled “SMART ATTENDANCE” which is being submitted to the Department of Electronics and Computer Engineering, IOE, Pulchowk Campus, in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Computer Engineering, is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this work.

Nikesh DC (075BCT052) _____

Ravi Pandey (075BCT065) _____

Rohan Chhetry (075BCT066) _____

Yukta Bansal (075BCT096) _____

Date: 21st March,2022

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the Department of Electronics and Computer Engineering, IOE, Pulchowk Campus, a minor project work entitled “SMART ATTENDANCE SYSTEM” submitted by Nikesh DC, Ravi Pandey, Rohan Chhetry, Yukta Bansal in partial fulfillment for the award of Bachelor’s Degree in Computer Engineering. The Project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfillment of Bachelor’s degree of Computer Engineering.

_____

Project Supervisor

Er. Lok Nath Regmi

Department of Electronics and Computer Engineering, Pulchowk Campus

_____

External Examiner

Er.

Computer Engineer, Government of Nepal Singhadurbar, Kathmandu, Nepal

_____

Head of the Department

Prof. Dr. Ram Krishna Maharjan

Department of Electronics and Computer Engineering, Pulchowk Campus

COPYRIGHT

ACKNOWLEDGEMENT

Firstly, we want to convey our thanks to our department, the Department of Electronics and Computer Engineering, IOE Pulchowk Campus for giving us this opportunity to work on our project of choice. We extend our sincere gratitude to our seniors who have provided us the necessary materials and motivated us to take upon this project. All our teachers and classmates deserve humble appreciation for providing us their suggestions on the selection and enhancement of project ideas.

ABSTRACT

The field of face recognition has been considerably researched and improved over recent decades immensely that it has become one of the most efficient, economical and reliable biometric recognition method. In this report, a novel way of recording attendance of students by employing face recognition technology has been presented. The teacher is responsible for starting attendance period during which students can capture their image and process via a mobile application which generates an embedding of face data to be sent to an attendance server. The server then marks their attendance by first validating the face and person match and provides feedback in real time. Also roll number of student has been used as constraint to reduce face recognition from k-NN space to simple 1x1 face verification for improved accuracy and performance. This record is saved in database and can be accessed at later time through a web application.

KEYWORDS: *Attendance, Face recognition, Android based, Real time.*

TABLE OF CONTENTS

1 Introduction 1

1.1 Background 1
1.2 Problem Statement 2
1.3 Objective 2
1.4 Scope 2
1.5 Applications 2

2 Literature Review 3

3 Related Theory 5

3.1 Biometrics Recognition 5
3.2 Face Recognition 5
3.3 Viola-Jones Algorithm 6
3.3.1 Haar-Like Features 6
3.3.2 Creating an Integral Image 7
3.3.3 Running Ada-Boost Training 7
3.3.4 Creating Classifier Cascades 8
3.4. Face-Net 8
3.5. Network Socket 9

4 Methodology 11

4.1 System Block Diagram 11
4.2 Working Principle 12
4.2.1 Student's Application 12
4.2.2 Attendance Server 13
4.2.3 Teacher's Application 14
4.2.4 Web Application 14
4.3 System Workflow Diagram 15
4.3.1 Attendance Procedure 15
4.3.2 Attendance Record Retrieval 16
4.4 Project Requirements 17
4.4.1 Used Frameworks 17

4.4.2 Required Resources 17

5 System Design 18

5.1 UML Diagrams 18

5.1.1 Use Case Diagram 18

5.1.2 Class Diagram 19

5.1.4 Activity Diagram 20

5.2 Project Outcome 21

5.3 Results 21

5.4 Conclusion 22

5.5 Future Enhancements 22

6 References 23

7 Appendices 24

7.1 Appendix A: Snippets of Student App 24

7.2 Appendix B: Snippets of Teacher App 26

7.3 Appendix C: Snippets of Web App 28

7.4 Appendix D: Github Link 30

LIST OF FIGURES

1	Face Detection using Viola-Jones Algorithm	6
2	Smart Attendance System Block Diagram of student’s and server’s Interaction	11
3	System workflow during attendance	15
4	System workflow during attendance retrieval	16
5	Use Case Diagram	18
6	Class Diagram	19
7	Activity Diagram	20

1. INTRODUCTION

1.1 Background

Records of attendance and verified authentication is crucial for many organizations and thus require an effective and efficient system for maintaining the attendance records or identifying correctly an individual. Manual book-keeping and verification have been in use since ancient times but they are cumbersome, labor- intensive, prone to record loss and false verifications. Keeping such problems in mind, various automations have been introduced in recent years. The use of biometric recognitions is one of the fields that has been gaining widespread popularity. Biometric authentication relies on use of distinctive and measurable traits of human beings such as fingerprint, palm print, palm vein, iris/retina, voice, electroencephalogram, electrocardiogram, face, etc. Biometrics seems to be a robust candidate technology for individual authentication. More conventionally, token-based identification such as RFID, barcode, QR code, NFC have also been used but their limitation on enforcing validness is pretty obvious. For selection of biometrics for certain application Jain et al. [1] identified seven factors: universality, uniqueness, permanence, measurability, performance, acceptability and circumvention.

For fingerprint recognition, a portable fingerprint recognizing device is first configured with the fingerprints of attendees and then the attendees would register their fingerprint later for attendance. This is usually carried in serial fashion/queue (when there are large number of attendees at a time). Also, a problem with fingerprint or palm recognition is that its accuracy plummets when the finger is wet or dirty. Similar problems are observed with using iris or retina-based systems for attendance. In recent years, work in face recognition has been expanding rapidly and accuracy as well as adaptability have been considerably increased making it a strong contender for biometrics-based system.

Several methods have been introduced recently for authentication through face recognition, which can be classified as feature-based and image-based. Feature-based methods rely on predefined traits on a human face and use them to distinguish one face from another. However, this approach is slowly being dominated by the image-based approach which statistically compares the actual face image for recognition.

According to Center for Strategic and International Studies (CSIS) (2020) [2], verification algorithms used to match subjects to clear reference images can achieve accuracy scores as high as 99.97% on standard assessments like NIST's Facial Recognition Vendor Test (FRVT) which is comparable to best results of iris scanners.

Most of the university students carry smartphones with them capable of taking self-images and face recognition algorithms have been getting more accurate as well as efficient. Also, mobile phones have been gaining considerable computation capability. In this context, mobile phones are prime candidate for autonomous activities such as attendance through face recognition.

1.2 Problem Statement

A traditional method of recording student attendance is by calling out their roll numbers and marking the attendance manually. The task of taking attendance of students manually is time-consuming and prone to proxy attendances such as marking someone present despite their physical absence. A change in the attendance marking system is long overdue and our motivation for this project.

1.3 Objective

To mark the attendance of a student economically and efficiently by using facial recognition and manage attendance logs of different classes in an attendance database and query out required subject attendance within a particular time frame of a particular class.

1.4 Scope

Through the proposed system is mainly intended for economically and efficiently automating attendance of students in educational institutions, similar or slightly modified methods can be installed in varieties of organizations for different purposes such as surveillance.

1.5 Applications

- Semi-autonomous and reliable attendance of students during lectures in educational institutions.
- Attendance in programs with large number of attendees such as workshops/seminars.

2. LITERATURE REVIEW

A brief look at recent works shows that the concept of face recognition and its variations have been incorporated into attendance management and similar systems by several groups and organizations. Kawaguchi, Yohei ,et. al.[3] illustrated a system that takes attendance of students for a classroom lecture by continuous observation of the lecture's video feed. They highlighted difficulties in estimating attendance precisely using face recognition independently as the accuracy of face recognition is not sufficiently high leading to misrecognition of faces. To remedy this, they suggested using constraints of correspondence between sitting position of students and their faces.

An android based face recognition attendance system using linear discriminant analysis was proposed by Sunaryono et. al.[4] with 97 percent accuracy and required 0.000096s to recognize a face image in the server. Their proposed system mandated students to capture their face image and displayed QR code using their smartphone.

Gomes et. al.[5] put forth an automated attendance system which automatically detects the student face when he/she enters the class using a camera mounted near the entrance and then marks their attendance based on face detection and recognition algorithms.

Kadry Seifedine and Smaili Khaled[6] proposed a system in which the attendance is done through biometric recognition employing iris matching. The minutiae are extracted by employing feature extracting algorithm from a digital image of the person's eye taken by the system which is then stored as template for verifying later. For verification, a person places his eye on the iris recognition sensor which then extracts the minutiae from the captured image and then sorts through a matching algorithm using the previously stored template.

The system proposed by Khuhawar et. al.[7] records daily attendance of students automatically subject-wise according to an administrator defined schedule. Their system uses a fixed camera which automatically snaps image of the classroom at scheduled time. The system then applies face detection using Histogram of Oriented Gradient and deep learning techniques to calculate-and-compare 128-d face features

for face recognition. The data of successfully recognized students is recorded in an excel sheet automatically.

These approaches towards creating an attendance system using face detection and recognition have varying applications in their respective use cases. In this project, we take motivation from these systems and several other similar works and aim to create a robust system for making the attendance taking procedure semi-autonomous and hassle-free.

3. RELATED THEORY

3.1. Biometrics Recognition

In myriad of authentication and verification systems, biometrics is seen as a possible replacement to traditional ID cards, keys, and passwords. This method has a high level of commonality, uniqueness, ease of acquisition, persistence, mobility, and fraud resistance. Individual's physical, biological, or behavioral traits, such as fingerprints, iris and retina scans, face recognition, voice recognition, signatures, palm prints, hand geometry, and gait, are some features that have been quantified using biometrics.

Most identification systems use 1-to-N matching (where N specifies the number of enrolled clients) to train classifiers to determine the best candidate, whereas most authentication systems create personal classifiers for each client in the database and use 1-to-1 matching to validate each claim. To limit the risk of fraud, identification systems can use 1-to-(N + 1) matching to categorize invaders, and authentication systems can specify a threshold for invalid claims rejection in each personal classifier.

3.2. Facial recognition

Facial Recognition is a biometric identification approach that employs features extraction and quantification of facial landmarks and their patterns, to authenticate a person's identity. To identify, verify, and/or authenticate a person, the technology captures a collection of unique biometric data related with their face and facial expression. Biometric face recognition, unlike other forms of identification employs unique mathematical and dynamic patterns that make it one of the safest and most successful. The goal of face recognition is to discover a series of data of the same face in a database from an incoming image.

Although it is simple for humans to recognize face, seemingly requiring little to no effort, it is can be quite challenging for computer systems to mimic such ability. Conventionally, a face recognizer (face recognition algorithm) works as four step process. In the first step a face detection algorithm, such as Viola Jones object detection framework, is used to extract only useful information i.e. face of a person and remove background image from a static 2d image. Then the extracted image undergoes preprocessing to enhance photographic properties like alignment of face,

illuminations, etc. so that the accuracy of face recognition step can be improved. After this some feature vectors are extracted that uniquely identify the facial pattern. Finally, this extracted pattern is compared with preexisting facial data to find the best match, if any.

3.3. Viola-Jones Algorithm

The Viola-Jones algorithm [8] takes a long time to learn yet can recognize faces in real time at a high rate. The method looks at several smaller sub-regions of an image (this algorithm works on grayscale images) and tries to discover a face by looking for certain attributes in each sub-region. Because a picture might have numerous faces of varied sizes, it must verify many distinct locations and scales. In this technique, Viola and Jones employed Haar-like properties to recognize faces.

The Viola Jones method comprises following four basic phases:

3.3.1. Haar Like Features

Digital picture characteristics called Haar-like features are employed in object recognition. Some universal qualities of the human face may be found in all human faces, such as the eyes area being darker than its neighbor pixels and the nose region being brighter than the eye region. Summarizing and comparing the pixel values of both regions is a straightforward approach to determine whether zone is lighter or darker. The total of pixel values in the darker area will be less than the sum of pixel values in the brighter area. This may be performed by employing Haar-like characteristics, which allow us to understand the various elements of a face.

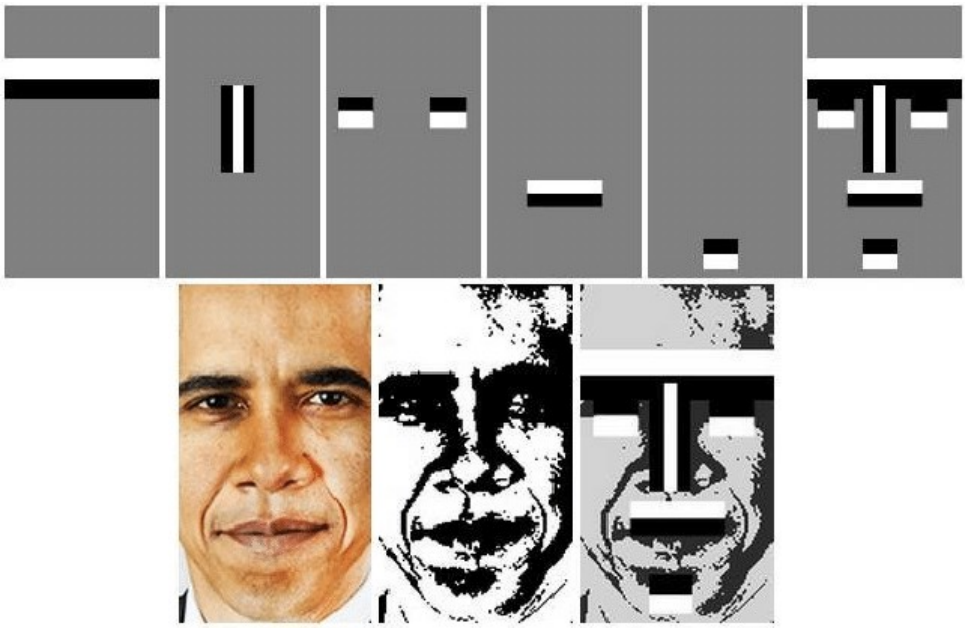

Figure 1 Face Detection using Viola-Jones Algorithm

Viola and Jones observed three sorts of Haar-like characteristics in their research:

1) Edge features
2) Line-features
3) Four-sided features

Edge and line characteristics can be used to identify edges and lines, respectively. Diagonal features are found using the four-sided characteristics. The feature's value is determined by subtracting the sum of pixel values in the black region from the sum of pixel values in the white area. A plain surface has a value of zero since all of the pixels have the same value and hence give no valuable information.

3.3.2. Creating an Integral Image

The number of pixels in a huge feature is substantially higher, the computations from Haar like features can be quite time consuming. The integral picture contributes to our ability to swiftly run these extensive computations in order to determine whether a feature or multiple features meets the requirements.

An integral image (also known as a summed-area table) is the name of a data structure as well as the technique used to create it. It is used to calculate the sum of pixel values in an image or a rectangular area of an image in a rapid and efficient manner.

3.3.3. Running Ada-Boost Training

After that, usually Ada-Boost is employed, a Machine Learning technique, to select significant elements for face identification. For example, nearly 160,000 characteristics are available in the 24x24 detector window, however only a few of these elements are significant for identifying a face. As a result, we apply the Ada-Boost method to choose the best characteristics among the 160,000. Each Haar-like feature in the Viola-Jones method represents a weak learner. Ada-Boost evaluates the performance of the classifiers provided to determine the kind and amount of a feature that will be included in the final classifier. The classifier will respond strongly to some sub-regions. Positives indicate that the classifier believes the image has a human face. In the perspective of the classifiers, sub-regions that do not generate a significant reaction do not include a human face. Negatives will be assigned to them.

The relevance or weight of the classifiers that performed well is increased. The end result is a strong classifier, also known as a boosted classifier, which combines the best weak classifiers. So, when we train the Ada-Boost to recognize essential characteristics, we feed it information in the form of training data and then train it to learn from that data to forecast. In the end, the algorithm determines whether anything may be classed as a helpful characteristic or not by defining a minimal threshold.

3.3.4. Cascading Classifiers

Perhaps the Ada-Boost will ultimately choose the greatest features around 2500, but calculating these features for each area is still a time-consuming procedure. We have a 24x24 window that we will glide over the input image to see whether any of the regions contain the face. The cascade's goal is to swiftly eliminate non-faces in order to save time and calculations. As a result, the essential speed for real-time face detection is achieved.

For this a cascaded method is used in which the process of recognizing a face is divided into many steps. The classifier in the first stage is made up of our best features; in other words, the sub-region goes through the best characteristics in the first stage, such as the feature that identifies the nasal bridge or the feature that identifies the eyes. All of the remaining features will be added in the following stages. The first step evaluates an image sub-region as it enters the cascade. The stage's output is maybe if it assesses the sub-region as positive, implying that it believes it is

a face. When a sub-region receives a maybe, it moves on to the next step of the cascade, and so on until we reach the final stage.

The image is eventually categorized as a human face and given to the user as a detection if all classifiers approve it. If the first step determines that the image does not contain a human face, it is immediately dismissed. It is also deleted if it passes the first step but fails the second. Essentially, the image can be deleted at any point throughout the classification process.

3.4. Face-Net

Face-Net [9] is a unified system for face verification (is this the same person), recognition (who is this person) and clustering (find common people among these faces). This method is based on learning a Euclidean embedding per image using a deep convolutional network. The network is trained such that the squared L2 distances in the embedding space directly correspond to face similarity: faces of the same person have small distances and faces of distinct people have large distances. Once this embedding has been produced, then the afore mentioned tasks become straight-forward: face verification simply involves thresholding the distance between the two embeddings; recognition becomes a k-NN classification and clustering can be achieved using off-the-shelf techniques such as k-means or agglomerative clustering. FaceNet directly trains its output to be a compact 128-D embedding using a triplet based loss function based on LMNN. The triplets consist of two matching face thumbnails and a non-matching face thumbnail and the loss aims to separate the positive pair from the negative by a distance margin. The Triplet Loss aims to minimize the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity i.e. we strive for an embedding f(x), from an image x into a feature space $\mathbb{R}^d$, such that the squared distance between all faces, independent of imaging conditions, of the same identity is small, whereas the squared distance between a pair of face images from different identities is large. The FaceNet trained the CNN using Stochastic Gradient Descent (SGD) with standard backpropagation and Ada-Gradient. The FaceNet achieves a classification accuracy of 98.87%±0.15 when using the fixed center crop and 99.63%±0.09 standard error of the mean when using the extra face alignment on the Labelled Faces in the Wild (LFW) dataset.

3.5. Network Socket

A network socket is a software structure within a network node of a computer network that serves as an endpoint for sending and receiving data across the network. The structure and properties of a socket are defined by an application programming interface (API) for the networking architecture. Sockets are created only during the lifetime of a process of an application running in the node. The application programming interface (API) for the network protocol stack creates a handle for each socket created by an application, commonly referred to as a socket descriptor. In Unix-like operating systems, this descriptor is a type of file descriptor. It is stored by the application process for use with every read and write operation on the communication channel.

At the time of creation with the API, a network socket is bound to the combination of a type of network protocol to be used for transmissions, a network address of the host, and a port number. Ports are numbered resources that represent another type of software structure of the node. They are used as service types, and, once created by a process, serve as an externally (from the network) addressable location component, so that other hosts may establish connections. Network sockets may be dedicated for persistent connections for communication between two nodes, or they may participate in connectionless and multicast communications.

An application can communicate with a remote process by exchanging data with TCP/IP by knowing the combination of protocol type, IP address, and port number. This combination is often known as a socket address. It is the network-facing access handle to the network socket. The remote process establishes a network socket in its own instance of the protocol stack, and uses the networking API to connect to the application, presenting its own socket address for use by the application

A protocol stack, usually provided by the operating, is a set of services that allow processes to communicate over a network using the protocols that the stack implements. The operating system forwards the payload of incoming IP packets to the corresponding application by extracting the socket address information from the IP and transport protocol headers and stripping the headers from the application data. API that programs use to communicate with the protocol stack, using network sockets, is called a socket API. Development of application programs that utilize this

API is called socket programming or network programming. Based on the socket address, Internet sockets deliver incoming data packets to the appropriate application process. A socket that has been connected to another socket, e.g., during the establishment of a TCP connection, also has a remote socket address.

4. METHODOLOGY

The attendance system has been developed in three major divisions: An attendance server, a mobile application for attendance marking and finally a Web application that runs directly with the server for easy query of attendance data from the server of different classes. The mobile application has two variations: one for the attendance taking entity, hereafter referred to as teacher and one for the attendee, hereafter referred to as student. The applications for teacher and student will have different features according to their usage.

4.1. System Block Diagram

The design of the various parts of the system is explained along with system block diagram below:

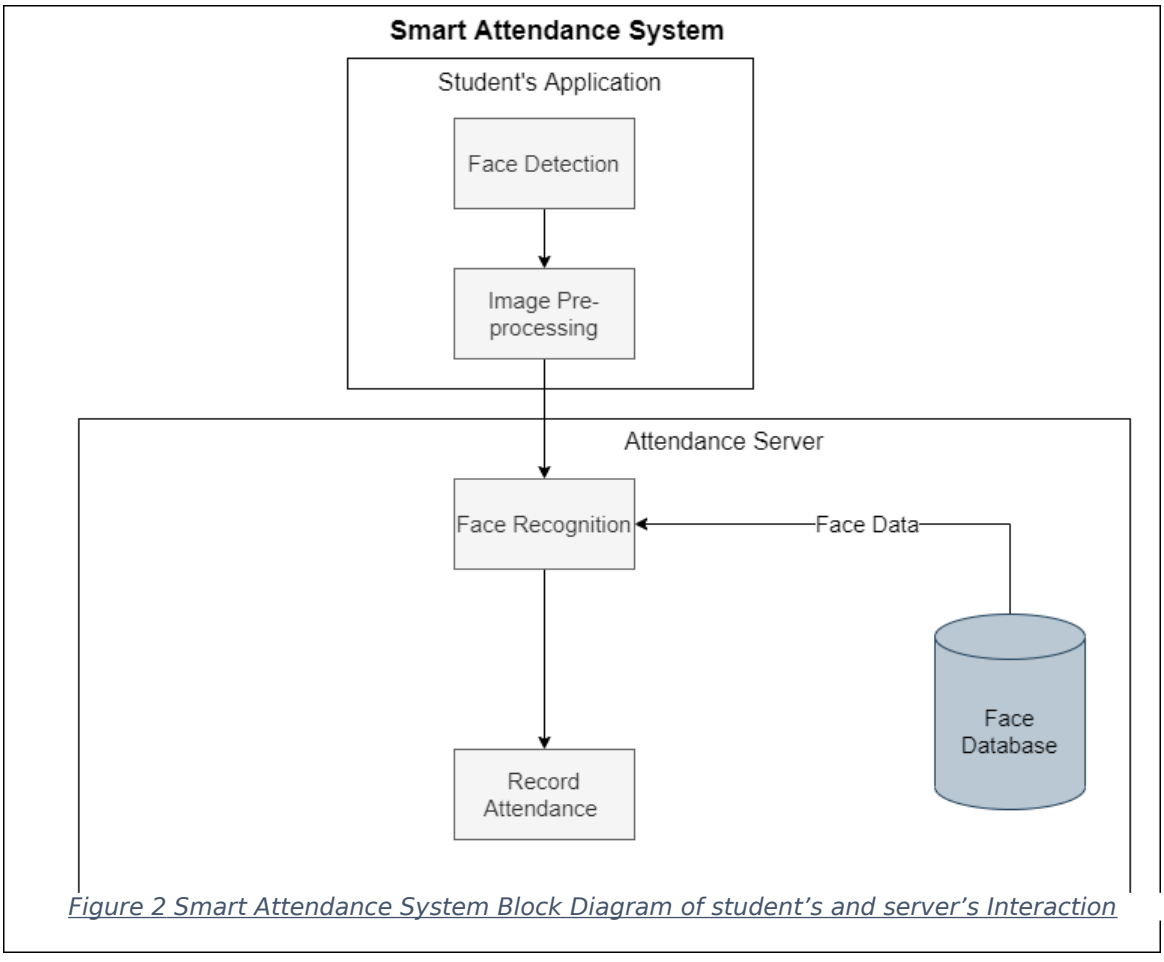

Figure 2 Smart Attendance System Block Diagram of student's and server's Interaction

4.2. Working Principle

When the teacher wants to take attendance, he/she connects to the attendance server through his/her mobile application after entering his/her teacher id and the respective class that attendance is to be taken. The attendance server then generates a random unique code and sends it to the teacher's mobile application, which is then shared with the students manually to initiate the attendance for a lecture. The attendance code is unique and its lifespan is only until the particular instance of attendance is taken. The students then enter the code into their application with their respective roll numbers and capture a self-image. The captured image is processed and sent to server for face recognition along with their unique identifier which for most case is their roll no which act as constraint for improving accuracy. Upon successful recognition of a student from the lecture's attendee, the student is marked as present and the information is feedback to student. If the student can't be recognized the student will be prompted to recapture their image. The teacher can turn off the attendance period when all present students have been recorded or after certain interval from start of attendance taking process so that proxy attendance of late students will be discouraged. The teacher receives a list of present and absent students of the particular class from the server. All the data of attendance is saved in the database which can be later queried by utilizing the web application by respective department or teacher.

Each component of the block diagram is explained in detail below:

4.2.1. Student's Application

A mobile application for the teachers is used to receive the unique code generated by the attendance server. The code must to entered by the student in order to start the attendance process along with his/her unique roll number. This is done to ensure the proper timing of the attendance irrespective to stringent routines but with respect to the event (such as lecture delays, changes or cancellation). After entering a valid code, the application will prompt the user to capture a self-image for face recognition and his/her entered roll no are captured. The captured image is then processed as follows:

4.2.1.1. Face Detection

The first step of any face recognition is face detection. The presence of a face in the captured image is detected by applying a frequently used face detection algorithm

proposed by Paul Viola and Michael Jones [8]; the Viola-Jones object detection algorithm. The algorithm uses Haar-like Features to localize the probable location of the face in the image, uses an integral image created from the original image for efficient computation and applies cascaded classifiers to detect the face. The portion in which the face id detected which is represented by a rectangular bounding box containing the face features is then passed over to the image pre-processor.

4.2.1.2. Image Pre-processor

This system first aligns face based on facial landmarks to obtain a canonical alignment based on translation, scale and rotation as it could provide higher accuracy[9]. Since the proposed system is based on the feature-based approach of face recognition, so the face image is processed to extract 128d face embedding data[9]. These feature data, here referred as face embedding, are unique to each distinct face and hence can be used for recognition. The face embedding is then sent to the attendance server. Since the face embedding is only of 128 bytes it places minimal constraint in network and server's performance.

4.2.2. Attendance Server

A server is used to manage the attendance logs of all students for corresponding lectures. The server utilized the popular SQL language and we have used it with PhpMyAdmin for easy handling. The server uses socket-based connection and ports for the student and teacher are separate and connected on call by respective entities to minimize overhead to the server. The data transfer is done in the popular JSON format.

4.2.2.1. Face Recognition

The server receives face embedding data obtained after the processing of the image along with roll no from the student application. Due to use of student's roll no as added constraint, face recognition is simplified to face verification which makes computation immensely faster as the comparison is reduced from k-NN classification to 1x1 comparison. This should allow our attendance server to handle large number of users simultaneously. The face data is then compared against the face data of students stored in the database for similarity using Euclidean L2 distance[9]. Recognition is considered successful when the similarity score between the images exceeds a

threshold value and the respective student is marked as present and notified to student in real- time. In the lower possible case of failure of recognition, the server prompts the student application to recapture image of the student.

4.2.2.2. Record Attendance

The attendance of all successfully recognized students is marked and the logs are stored in a central database. A list of present students and absentees is sent to the teacher application. The database holds the records with its respective time stamps and date with are necessary for the query out of data of the web application.

4.2.2.3. Face Database

A central database is maintained locally in the attendance server which holds face data for all the students enrolled in the organization. The face data is generated at the time of enrollment of a new student. The database holds the 128-byte embedding of each student in the database with is used in future attendance validations.

4.2.3. Teacher’s Application

A separate application is provided to the teachers. Through this app, they will start the process of attendance and control its active time or stop attendance after class is over to prevent proxy late attendance. They will also be able to view records list of class sent by the attendance server.

4.2.4. Web Application

A web application has been developed for easy access of the records of attendance in the database of particular subject within a particular time for a given class. The system is able to accurately query out required data from the database and present it to the system user. This application can be very useful at the end of semester when the total number of present days of students is required by the teacher.

4.3. System Workflow Diagram

4.3.1. Attendance Procedure

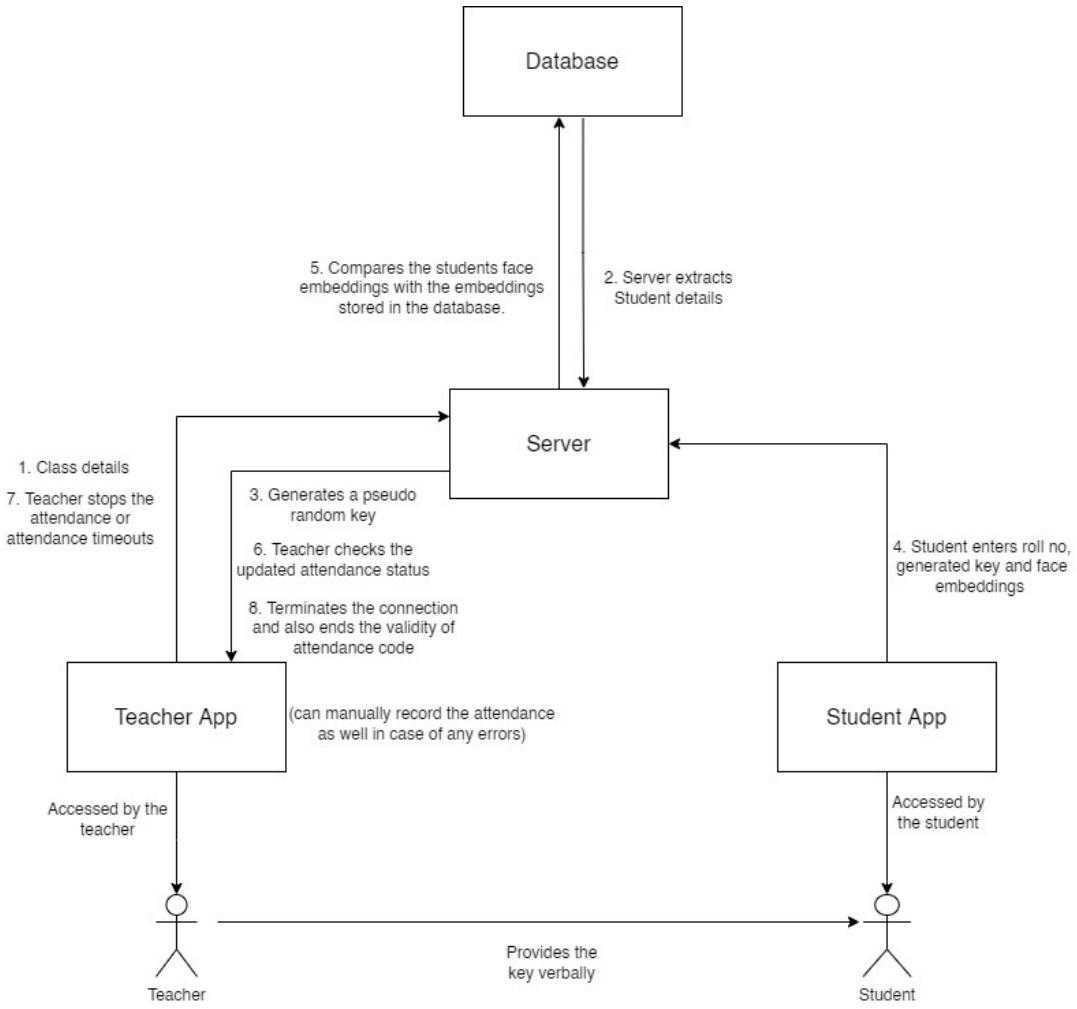

Figure 3 System workflow during attendance

4.3.2. Attendance Record Retrieval

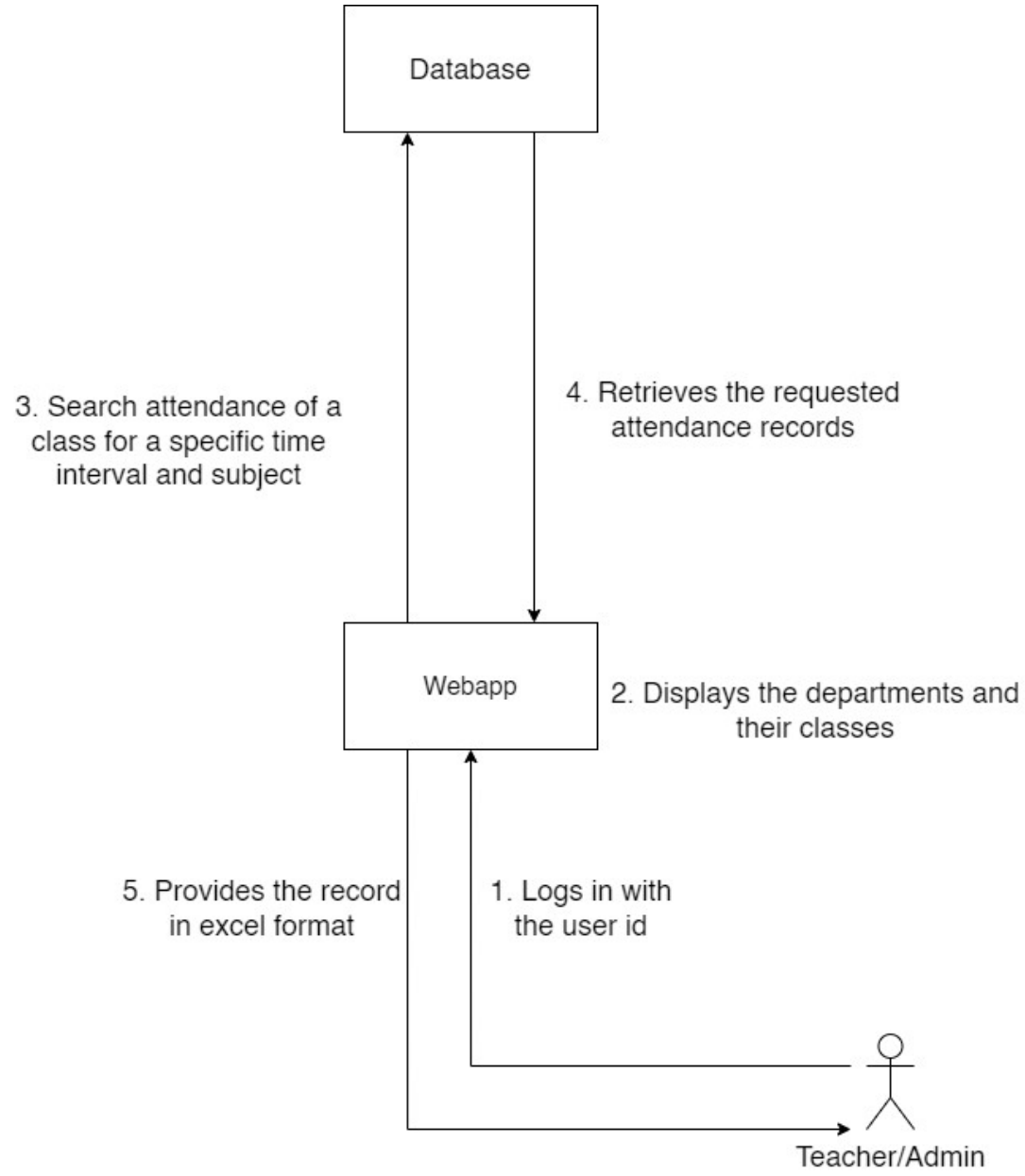

Figure 4 System workflow during attendance retrieval

4.4. Project Requirements

4.4.1. Used Frameworks

The system is built using the Open Source Computer Vision (OpenCV) library, which provided the necessary functionalities for face detection and recognition, in Object Oriented programming paradigm using python programming language. We used Kivy, a cross-platform python framework, to build the student and teacher applications. We used HTML, CSS and JavaScript along with Php for the creation of the web application. The server is built using SQL (MySQL) and database was managed by PhpMyAdmin, data transfer between server and applications is done in the JSON format. The server utilizes socket-based connections with the teacher and students application with connection on request feature which is done to decrease the overhead for the server, so the server will be continuously listening to it ports for requests from teacher and student applications.

4.4.2. Required Resources

- List of students records along with pictures for creating Face database
- A high-performance computer for attendance server
- LAN connection for server and WIFI for smartphones of teachers and student

We also referenced various materials such as OpenCV documentation, Python documentation and Kivy documentations and manuals along with the papers cited in the references section to gather the necessary information to initiate the development process. We followed according to some tutorials and articles on face recognition systems online in order to code the various modules of the system and after proper unit testing of each modules, we finally assembled them into a complete attendance system.

5. SYSTEM DESIGN

5.1 UML Diagrams

5.1.1. Use Case Diagram

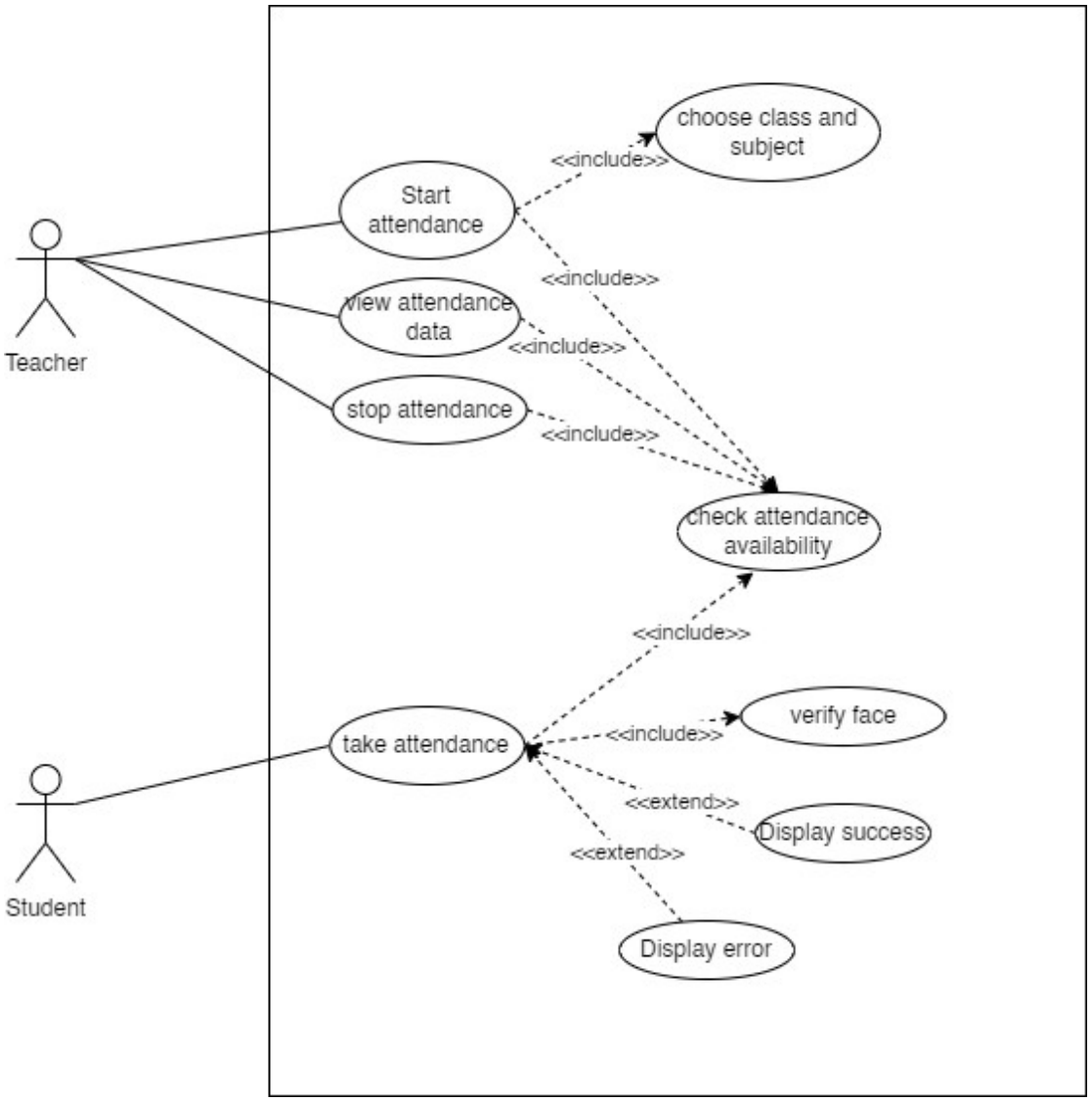

Figure 5 Use Case Diagram of Teacher's and Student's interaction with Smart Attendance System

5.1.2. Class Diagram

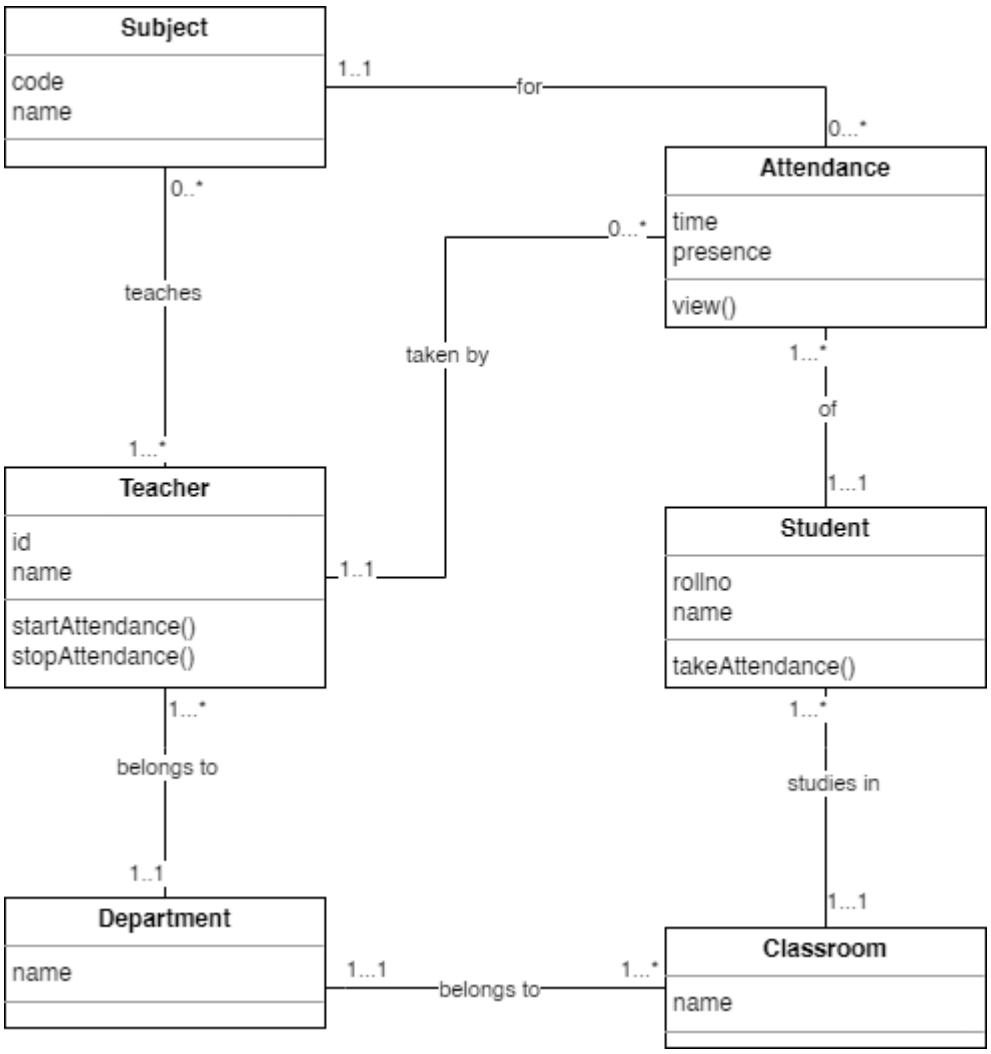

Figure 6 Class Diagram of Smart Attendance System

5.1.4. Activity Diagram

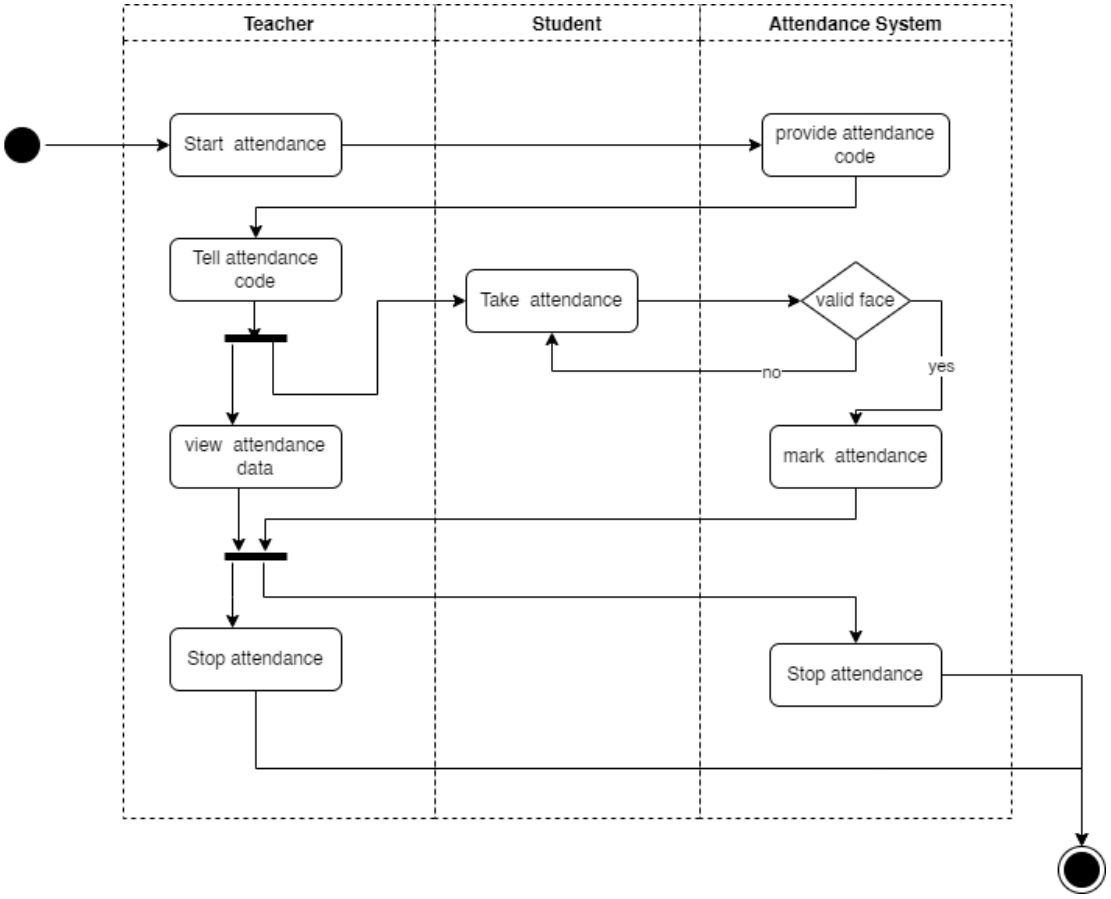

Figure 7 Activity Diagram of Smart Attendance System

5.2. Project Outcome

Teacher connects to attendance server to start attendance for corresponding lecture through his/her application which will generate a student list with present and absent marked and updated by the server. Student will then be able to mark himself present with just a click of button and receive feedback almost instantly from the server if he/she has been marked present. This should reduce effort considerably required for taking attendance and make it a more systematic way to store and view records.

The teacher can also utilize our Web Application which refers to the attendance database directly and queries out the given subject-wise and provided time frames attendance of required class.

5.3. Results

Our Smart Attendance program can function as we had envisioned at a much smaller scale. We have included all the features we had proposed with a functioning teacher application to start attendance, a student application to capture the student face for attendance purpose, a server which listens for requests of students and teachers request and does according changes to the database and also a web application that can be utilized to query out the required presence of a particular class for a given subject over a selected time frame.

Due to various constraint our product is capable of doing all the given functions but only to the extend in the same computer at the moment. We have limited the functionalities as such to configure and test all the features for a demonstration. Also we have coded the program for student and teacher application in a format suitable for mobile development but due to our inexperience in the field of android development and kivy-python platform being a relatively new framework for mobile application development we were unable to make the android apk work on mobile platform. The coding of socket programs also have been configured to work locally only in a particular computer environment which to go into the deployment phase requires migration into internet protocol with its necessary security measures from various internet attacks to safe guard the data of the database.

5.4. Conclusion

The above described Smart Attendance System is our submission for the minor project for the 6th semester, a face recognition-based attendance system which utilizes database to digitalize our attendance system for our college and will help us develop our knowledge on, image face recognition techniques, database management and teamwork in a software project.

The proposed system is aimed to digitalize the attendance process and make it easier for teachers to get the required presence of students throughout the semester of a particular subject.

5.5 Future Enhancements

As we have discussed in the entire project with some ways the Smart Attendance product can be lacking and may require improvements in various of its functions. Here we will be listing all the possible ways we think the project can be taken further making it to a much better state. The points are listed below:

1. QR code instead of attendance code
2. Deployment into the internet protocol
3. Android and IOS stable mobile application
4. Updating face embedding dynamically.

References

[1] A.K. Jain, R. Bolle & S. Pankanti. *Biometrics: Personal Identification in Networked Society*. Boston: Kluwer Academic Publications, 1999.

[2] W. Crumpler. "How Accurate are Facial Recognition Systems – and Why Does It Matter?".Strategic Technologies Blog, 2020. Retrieved From: https://www.csis.org/blogs/technology-policy-blog/how-accurate-are-facial-recognition-systems---and-why-does-it-matter

[3] K. Yohei, T. Shoji, W. Lin, K. Kakusho & M. Minoh. *Face Recognition Based Lecture Attendance System.* 2005.

[4] D. Sunaryono, J. Siswantoro & R. Anggoro. *An Android Based Course Attendance System Using Face Recognition*. Journal of King Saud University - Computer and Information Sciences. 2019.

[5] C. Gomes, S. Chanchal & D. Jadhav. *Class Attendance Management System using Facial Recognition*. ITM Web of Conferences, 2020.

[6] S. Kadry & K. Smaili. *A Design and Implementation of a Wireless Iris Recognition Attendance Management System.* Information Technology and Control, 2007.

[7] F.Y. Khuhawar, K.L. Bhatti, L. Mughal, S.A. Memon. *Smart Attendance Management System Using Face Recognition*. EAI Endorsed Transactions on Creative Technologies, 2018.

[8] P. Viola, M. Jones. *Rapid Object Detection using a Boosted Cascade of Simple Features*. Accepted Conference on Computer Vision and Pattern Recognition 2001, 2001.

[9] F. Schroff, D. Kalenichenko & J.Philibin. FaceNet: *A Unified Embedding for Face Recognition and Clustering*. 2015 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2015

APPENDICIES

APPENDIX A: Snippets of Student App

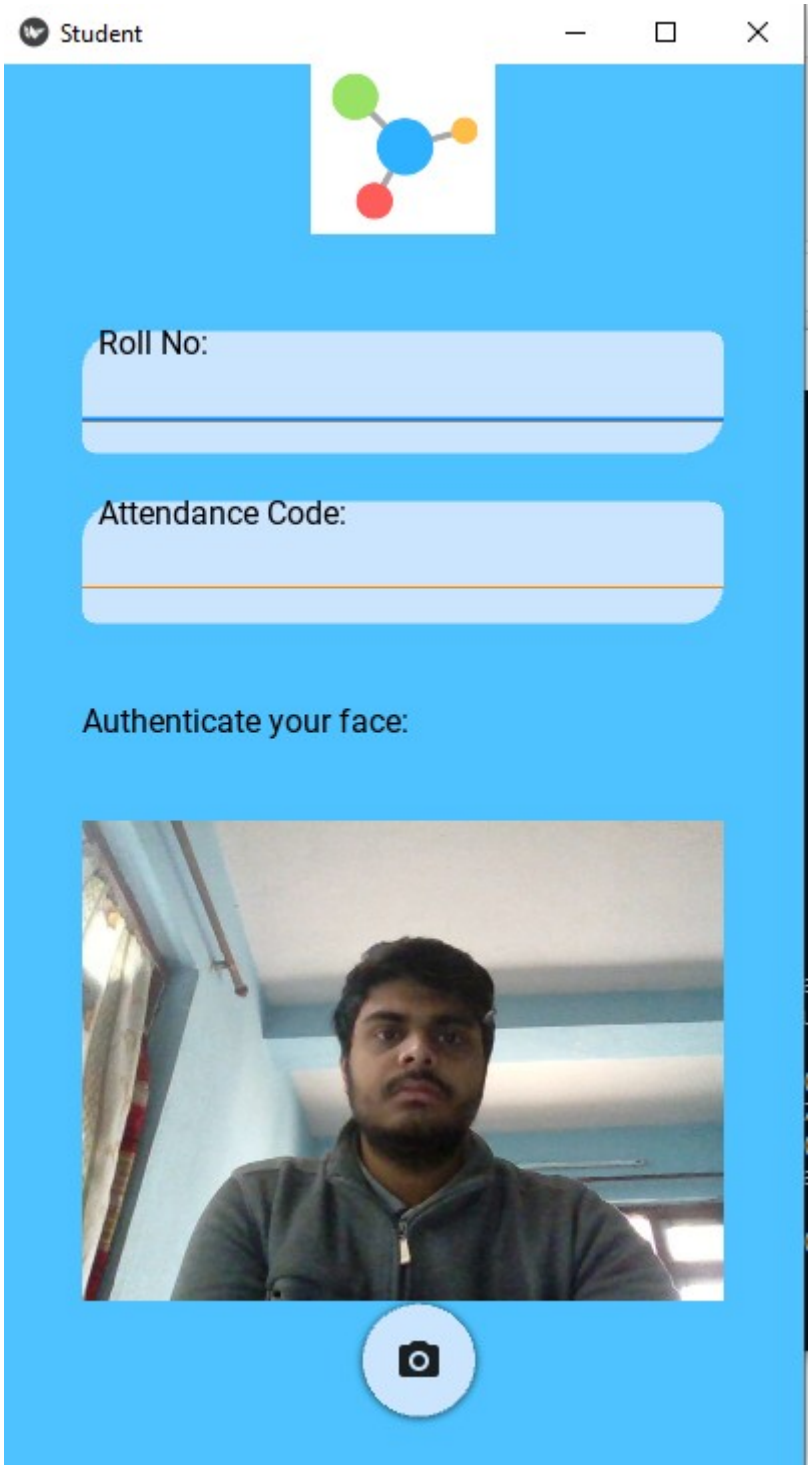

Figure 8 Student App Screenshot 1

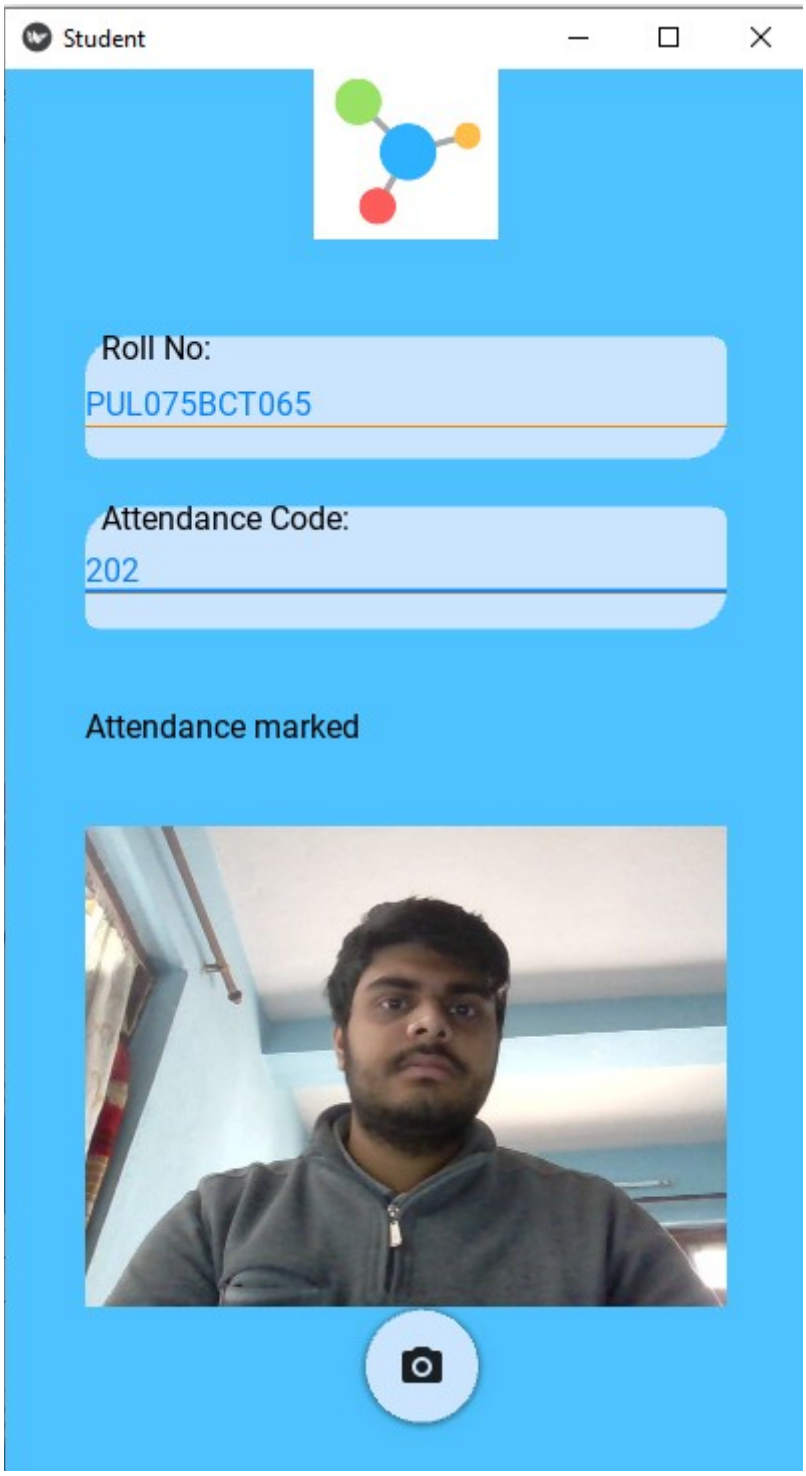

Figure 9 Student App Screenshot 2

APPENDIX B: Snippets of Teacher App

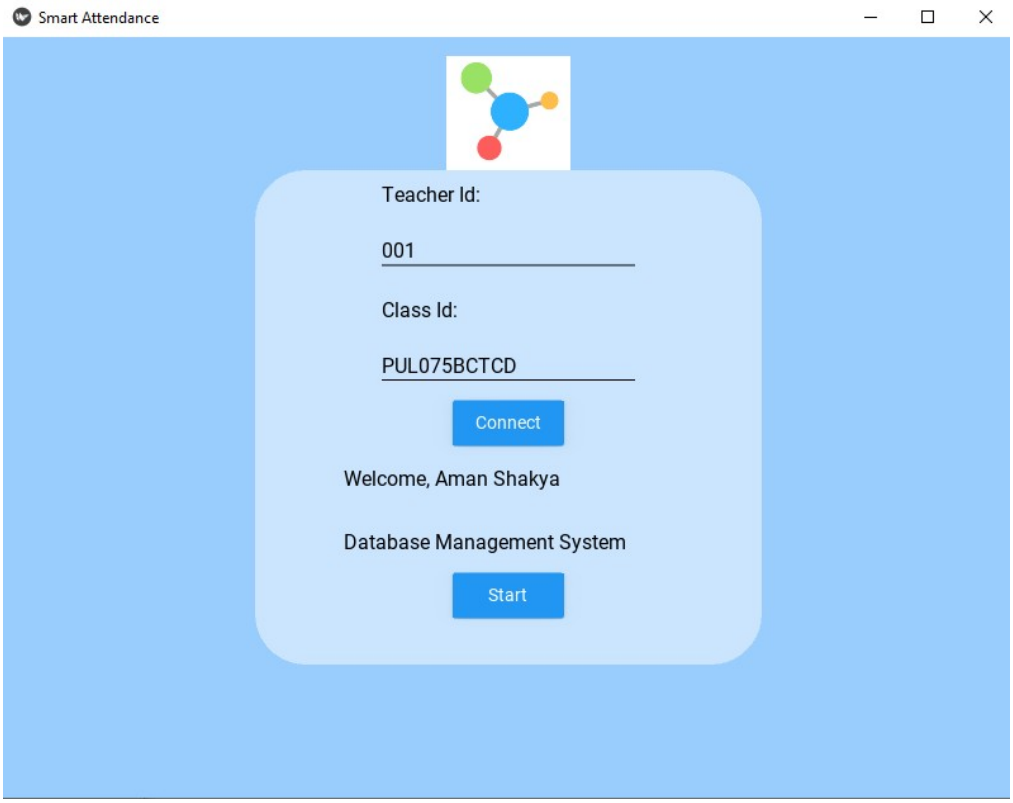

Figure 10 Teacher App Screenshot 1

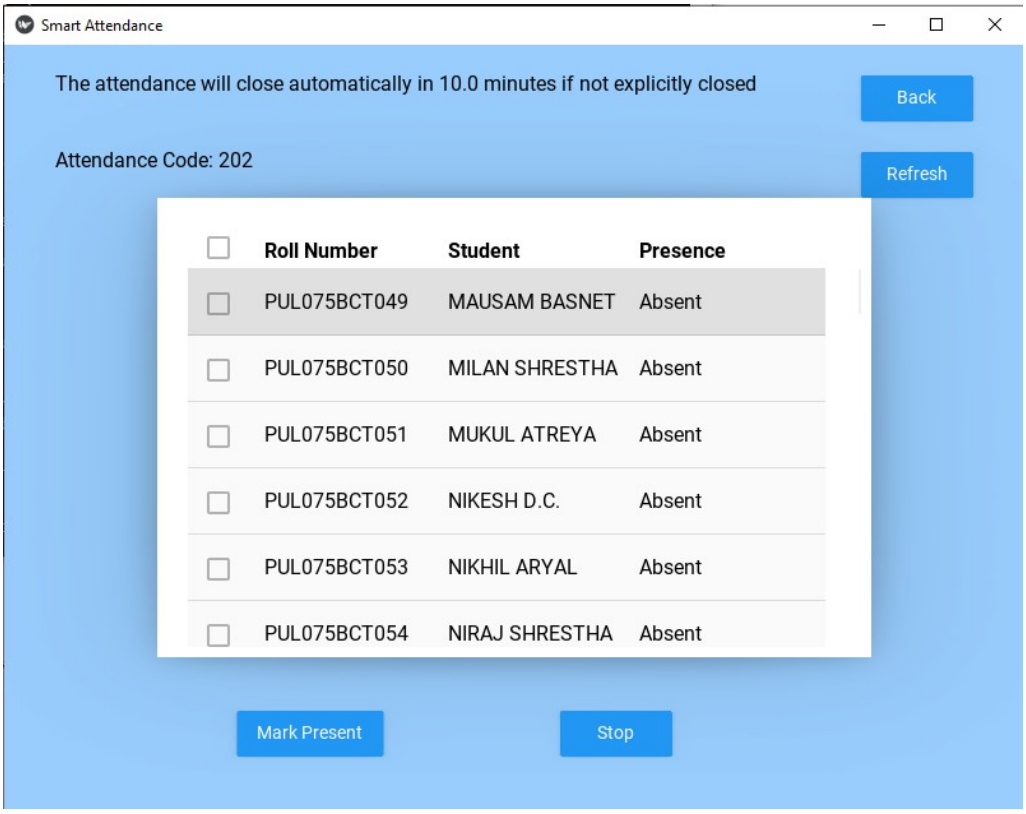

Figure 11 Teacher App Screenshot 2

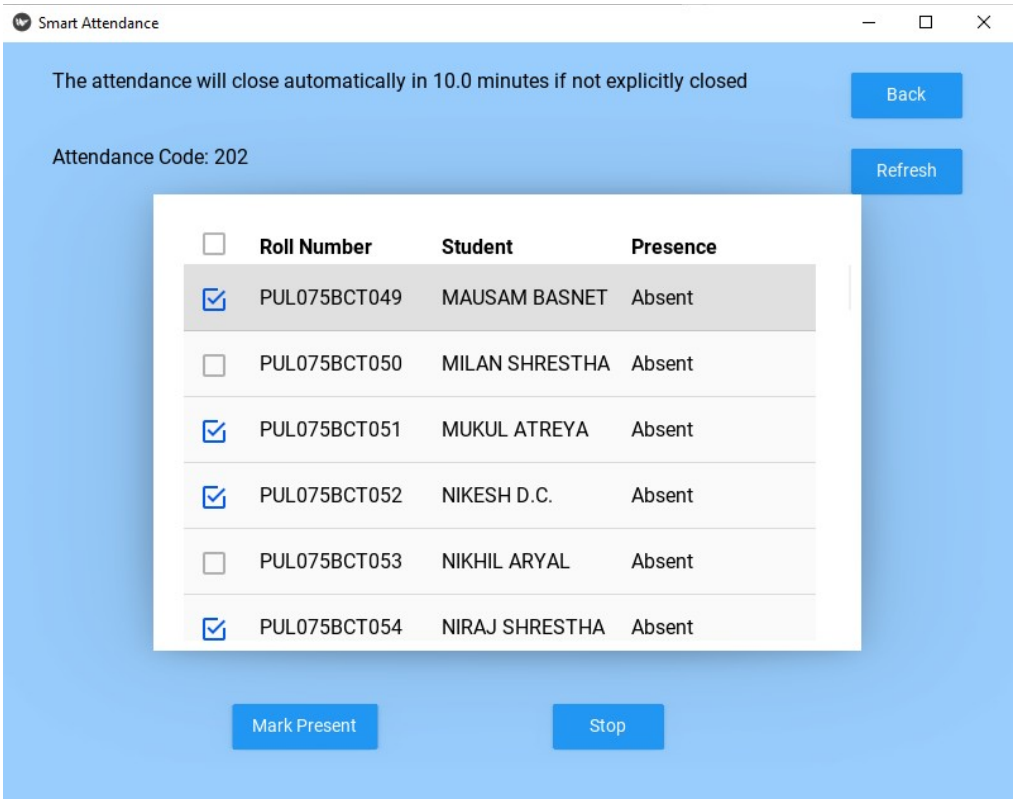

Figure 12 Teacher App Screenshot 3

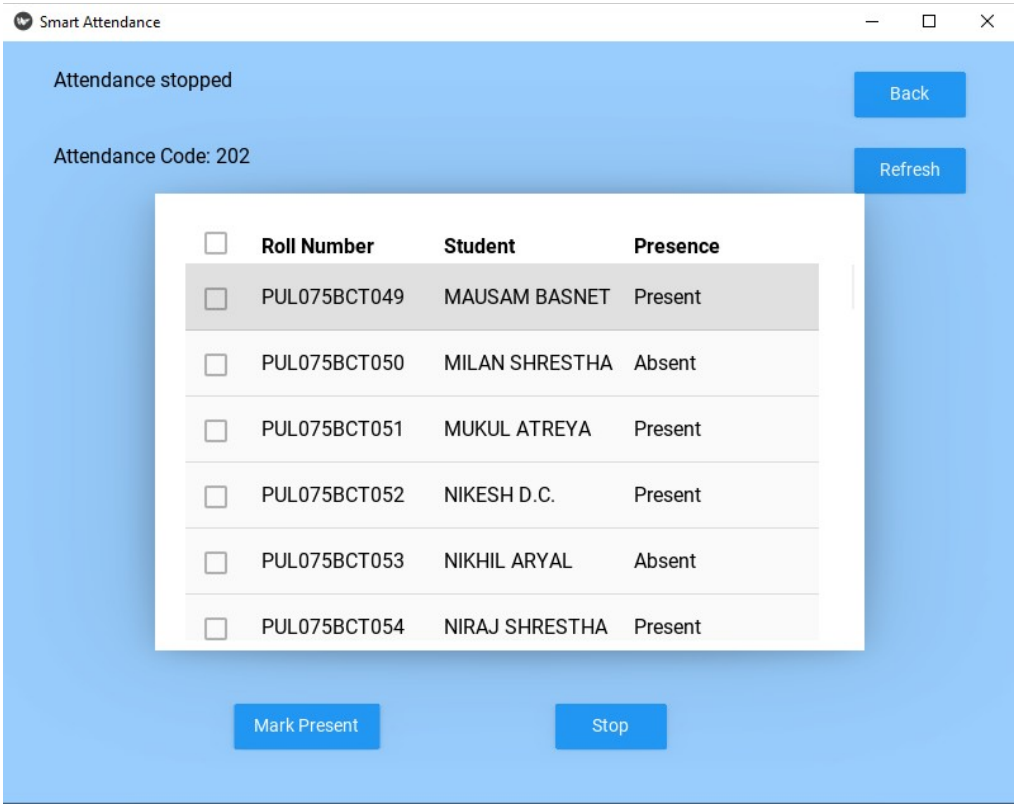

Figure 13 Teacher App Screenshot 4

APPENDIX C: Snippets of Web App

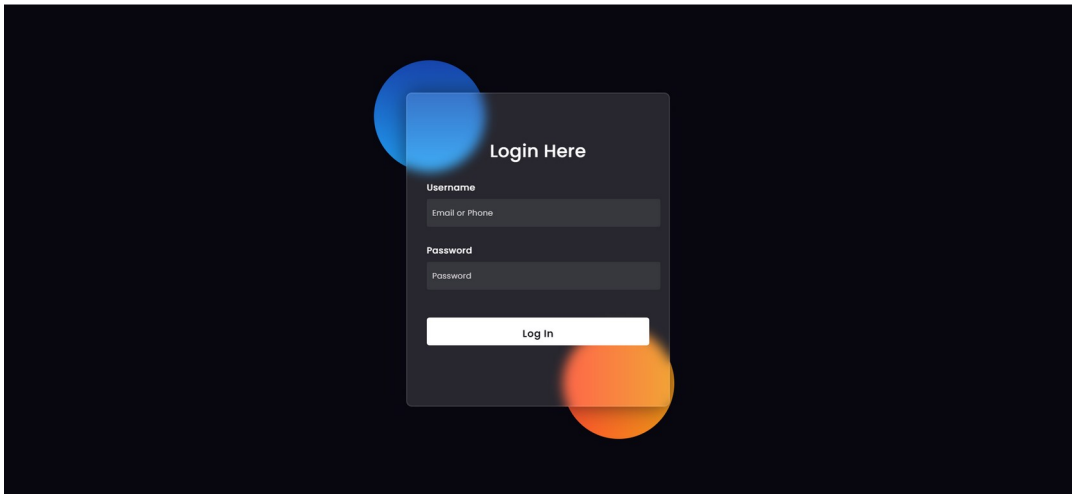

Figure 14 Web Application Screenshot 1

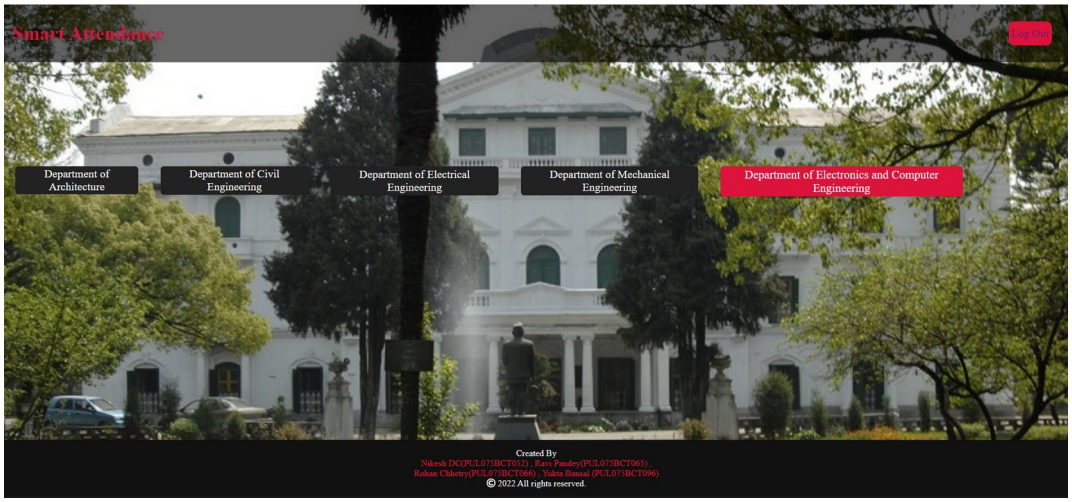

Figure 15 Web Application Screenshot 2

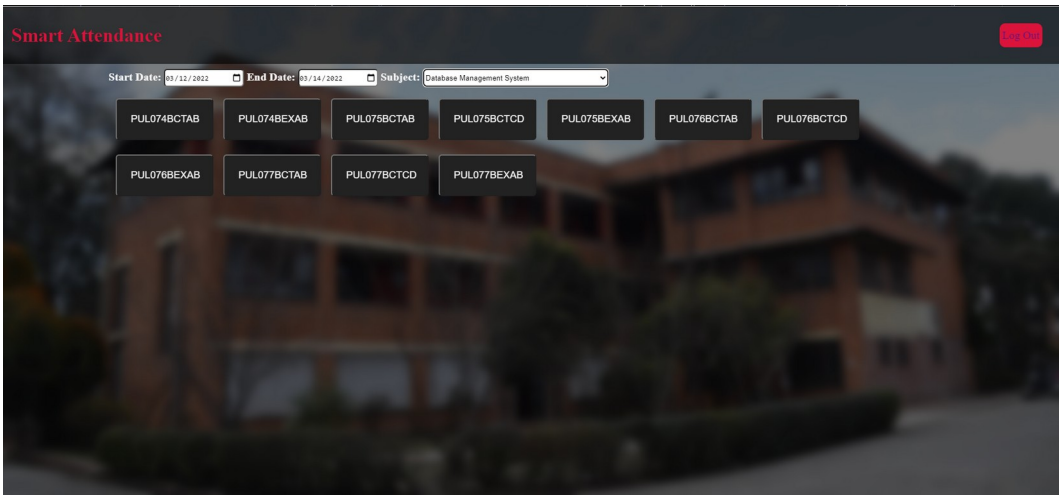

Figure 16 Web Application Screenshot 3

Figure 17 Web Application Screenshot 4

APPENDIX D: Github Link

github.com/ravip5921/SmartAttendanceSystem