IPTEK The Journal of Technology and Science, 35(1), 2024 (e/pISSN:2088-2033, 0853-4098)

DOI: 10.12962/j20882033.v35i1.15241 | Received 19 Dec 2022; Revised 22 Oct 2023; Accepted 28 Jan 2024

ORIGINAL RESEARCH

SMART ATTENDANCE MANAGEMENT SYSTEM

Frank Onaifo[1] | Alexander Akpofure Okandeji[1] | Alex Eluro[1] | Francis Owolabi[1]

[1] Dept. of Electrical and Electronics Engineering, Olabisi Onabanjo University, Ago-Iwoye, Ogun State, Nigeria

[2] Dept. of Electrical and Electronics Engineering, University of Lagos, Akoka, Yaba, Lagos, Nigeria

Correspondence

*Alexander Akpofure Okandeji, Department of Electrical and Electronics Engineering, University of Lagos, Akoka, Yaba, Lagos, NigeriaEmail: aokandeji@unilag.edu.ng

Present Address

Dept. of Electrical and Electronic Engineering, University of Lagos, Lagos, Nigeria

Abstract

The research work considered the development of an attendance system using Artificial Intelligence technology. Using the Android mobile operating system developed by Google, XML, and Java programming, a smart attendance management system (SAMA) is developed for daily student attendance in colleges or schools. SAMA attempts to record attendance through face detection. Accordingly, it enhances academic performance in schools as students must attain minimum attendance requirements before they are allowed to take their examinations. It also prevents examination malpractice, such as impersonation during tests and exams. SAMA was designed and implemented using Olabisi Onabanjo University, Department of Electronic and Electrical Engineering, as a case study. Results showed that the proposed model eliminates or reduces the time wasted during manual collection of attendance, creates a student database management system that is not prone to errors or being manipulated by users, and, above all, aids in better management of classroom statistics for allocation of attendance.

KEYWORDS:

Android Mobile Operating System, Artificial Intelligence, Examination Malpractices, Java Programming, Smart Attendance Management System

1 | INTRODUCTION

The emergence of the electronic paradigm for learning compared to the traditional method and the availability of almost all information on the information superhighway (internet) nowadays have caused students to be less motivated to attend lectures physically. Laziness on the part of students, nonchalance to academic work, extra social activities that have no importance in aiding the institution's objectives, and a lot more may prevent students from attending lectures. In addition, lecturers and administrators in most developing countries have had to develop ways to ensure healthy student participation and keep the student-lecturer interactive relationship intact. In some cases, this has come in simple forms like roll calls, while in more interesting cases, it can be formatted like surprise quizzes, extra credit in class, etc. However, These strategies are time-consuming, stressful, and laborious because the valuable lecture time that could otherwise be used for lectures is dedicated to student attendance and is sometimes inaccurate. In addition to all these challenges, the tutor records attendance manually. Therefore, they

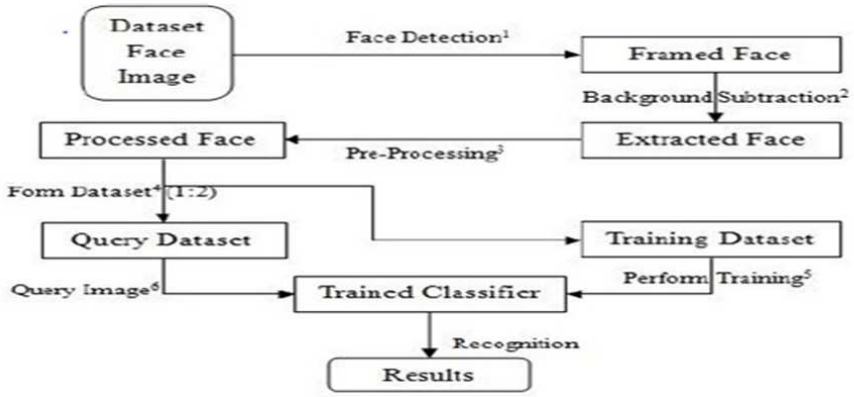

FIGURE 1 The block diagram of the proposed system.

are prone to personal errors. There arises a need for a more efficient and effective method of solving this problem. Effectively, a technology that can solve this problem and even do more is Artificial Intelligence (AI) technology. According to the father of Artificial intelligence, John McCarthy, AI makes machines behave like humans. This makes AI important for usage in hazardous areas where humans cannot cope. AI is implemented using a human model of behavior and reasoning to develop the Application software. Artificial Intelligence can be used in all areas of life if the enabling software is developed. In this paper, AI is deployed for the attendance management system.[1]

A smart attendance management system (SAMA) is an application developed for daily student attendance in colleges or schools. SAMA attempts to record attendance through face detection. The manual attendance record system is inefficient, requiring more time to record and calculate each student's attendance. Hence, a system is needed to solve the issue of manual attendance. While the move towards the digital era is being accelerated every hour, biometrics technologies have started affecting people's daily lives at every instance.[2]

The Smart Attendance Management system uses biometric features of the human face to detect and track students enrolled on the platform. Biometric technologies use human features for authentication, and they are one of the best security systems compared to conventional security systems of passwords or identification cards. The technology in implementing biometric systems ranges from readers to radio frequency identification and Bluetooth. However, they are expensive, and therefore, they have limited use. Hence, a system that does not require a special infrastructure is developed using the Artificial Intelligence Facial Detection and Recognition system. Effectively, this system allows easy attendance management using tace Detection, one of the most acceptable AI techniques.[3]

The existing system largely consists of a physical register where the supervisor manually inputs the attendance records of all students. Other technologies developed to replace this manual system include fingerprinting, retina scanning, voice recognition, etc. The problem with the existing system is that the manual system is time-consuming, and the advanced technologies are too expensive to be implemented on a large scale in any organization[4–7].

In this research work, a smart attendance system, made possible by implementing Artificial Intelligence technology tailored around the Nigerian Universities Commission (NUC) policy of ensuring a 70-percentage course attendance by students before the likelihood of writing a semester examination for any course, is presented. The application of Artificial intelligence technology to student course attendance monitoring problems, especially in developing countries in our proposition, would lead to the elimination or reduction of the quality time wasted during manual collection of attendance, creation of a student database management system that is not prone to errors or being manipulated by users, and above all aids in better management of classroom statistics for allocation of attendance.

This study aims to design and implement a software application using the Firebase Machine Learning Kit to record students' class attendance automatically. Specifically, the objectives of this research work are listed below:

1. To develop the algorithm for the attendance system.

2. To develop an application to design the smart attendance system.
3. To create a database that will keep the attendance record.
4. To implement and test the smart attendance system.

2 | DESIGN METHODOLOGY

The proposed system automates the tedious task of manually maintaining the attendance records. The administrator fills out the details of the teachers and students at the start of the semester. A class list is generated as students register for each course. These details can be used for further semesters with little changes. Figure 1 shows the block diagram of the proposed system.

The student's facial image is stored in a database. When attendance is taken, the face captured is detected with the first frame and extracted in codes. These extracted codes are referred to as the processed face. The processed face is stored in a query data set, where it is trained for recognition so that any time the face is shown, it will be recognized because it has been trained.

2.1 | Face Recognition

According to[8], face recognition uses a gradient orientation method computed on a strongly populated grid of equally spaced cells using one-to-one alignment of local contrast of the facial features. Firstly, the face image gets divided into connected areas called cells, which comprise a pixel. The vertical length of the edge of the facial features is computed using discrete derivative masks. Each pixel in the cell will be a parameter for edge orientation, and the gradient element will be attached. The vertical length called the bins, is a 180-histogram channel spread even over 180 or 360 degrees. This procedure occurs by accumulating a measure of local histogram energy to normalize all cells in the block. The steps involved are:

- Scale-space extreme detection.
- Orientation assignment.
- Descriptor extraction.

The essential thought behind the Histogram of Oriented Gradient (HOG) descriptors is that the geometry of the image can be described by the distribution of how much space the image occupies in the vertical direction concerning the horizontal direction of the pixel. These descriptors can be implemented by dividing the image into small, connected regions, called cells, and for each cell, compiling a changing vertical length direction of the faces as one, moving horizontally along the pixels. The combination of these vertical lengths (histograms) then represents the descriptor.

For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger space area of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in a better invariance to changes in illumination or shadowing. The HOG descriptor maintains a few key advantages over other descriptor methods. Such changes would only appear in larger spatial regions. Moreover, as per the discovery of Dalal and Triggs, coarse spatial sampling, fine orientation sampling, and strong local photometric normalization permit pedestrians' body movement to be ignored so long as they maintain a roughly upright position. The HOG descriptor is, therefore, suited for human detection in images.

The final step in object recognition using a Histogram of Oriented Gradient descriptors is to feed the descriptors into some recognition system using AI. (Badhe, Chaudhari, Kale, and Mane, 2016)

2.2 | Characteristics of the Proposed System

The Smart Attendance Management System has three main characteristics: it is user-friendly, easily generates reports, and is paperless. These three characteristics are explained below.

- *User Friendly:* The proposed system is user-friendly. The reason is that the retrieval and storage of data are fast and maintained efficiently. Furthermore, the graphical user interface is provided in the proposed system, which allows the user to deal with the system very comfortably.

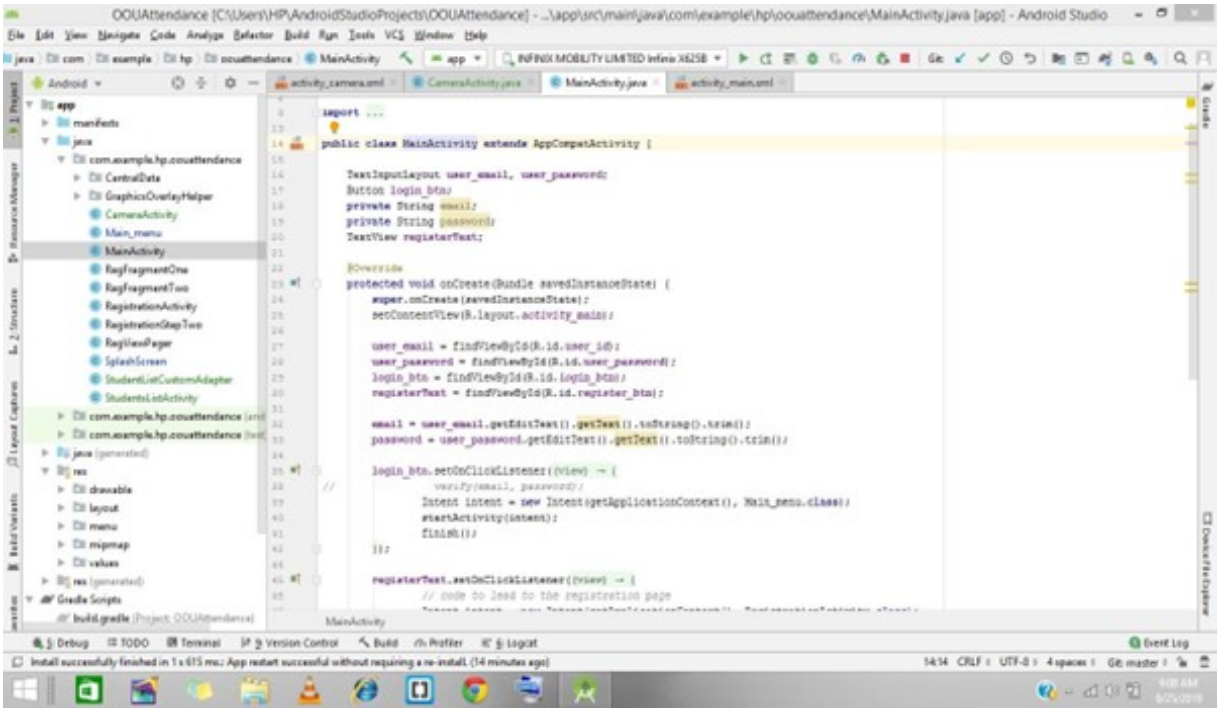

FIGURE 2 The Android Studio UI for XML.

- *Reports are easily generated:* Defaulter Reports can be generated very comfortably in the proposed system so that users can generate the report as per their requirement (monthly) or in the middle of the session. Users can provide the notice to the students to be regular.
- *No paperwork:* The proposed system does not require much paperwork. All the data is immediately fetched into the database, and the lecturers can generate reports easily. Furthermore, work becomes very easy because there is no need to keep data on paper.

The following are the software specifications required for implementing the system.

- Operating System - Windows 7 or higher (32bit or 64bit).
- Software Applications – Android Studio and Figma or Adobe XD.
- Programming language – Java, XML (Extensible Mark-Up language).

The hardware specification required for implementing the system is any Android device with an OS not less than Jelly-Bean(Android 4.2) that can take a clear picture.

3 | TECHNOLOGY OVERVIEW

3.1 | Android Studio

Java is a general-purpose programming language that is class-based and object-oriented. It is intended to let application developers write once and run anywhere (WORA), meaning compiled Java code can run on all platforms supporting Java without recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The android studio for different user interfaces is shown in Figures 2 and 3 .

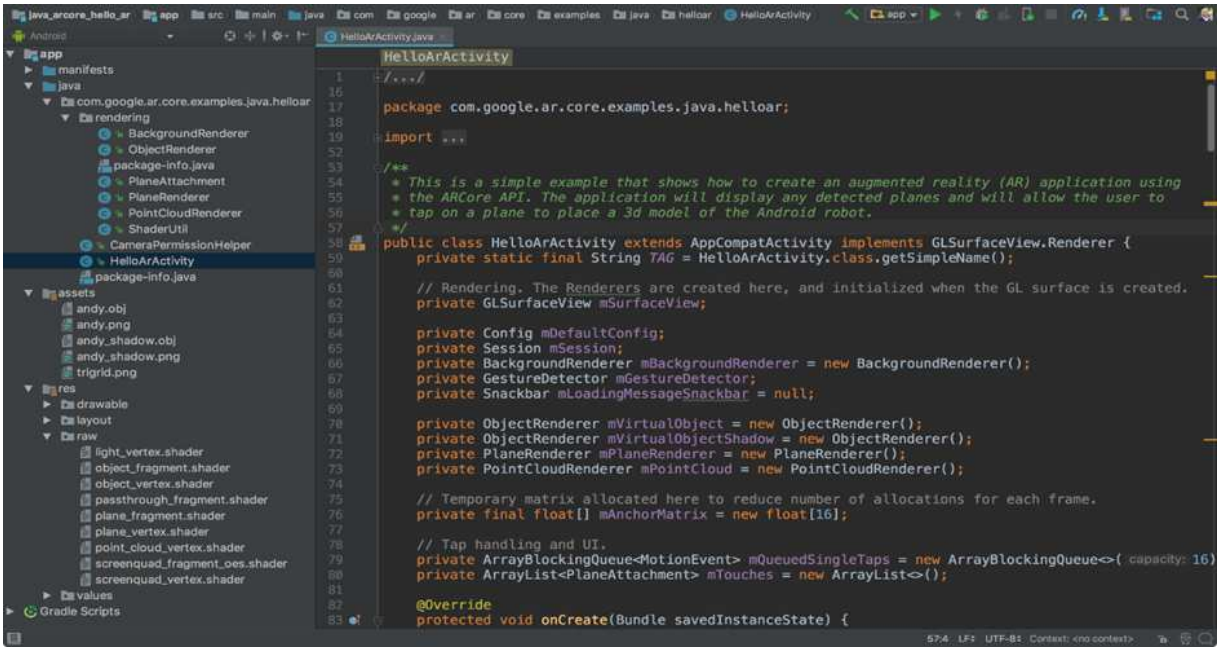

FIGURE 3 The Android Studio UI for Java.

3.2 | Android Device

The study uses an Android device to run and debug the application for this design. There are x reasons for this implementation decision. First, the Android device allows easy development. Plenty of updated modules support the implementation of various functionalities of our system. Second, the Android device allows easy deployment. Finally, Android devices are relatively cheaper compared to other mobile platforms.

4 | TESTING AND IMPLEMENTATION

4.1 | Implementation

The implementation of this study is divided into two parts, namely, the software interface and the hardware interface.

4.2 | System Functionality

When the registered user logs into the app, it takes the user (lecturer) to the page to accept the students' attendance through facial recognition. During this process, the mobile device checks the details of the face detected and compares them to the details of faces stored in the database. It is saved if it finds a match and the student's attendance is changed from absent (by default) to present.

The implementation mainly depends on the software since it has to be done manually, unlike the Android device, which can be purchased from any mobile store. The software interface is where the code was written to bring the application to reality. This was achieved using Android Studio.

4.3 | Software Interface

This comprises the software and applications used to complete this design. We used Android Studio as the integrated development environment (IDE) and built an Android mobile application.

4.4 | Develop an Android Application

The Android application was developed using the Android Studio IDE for the study.

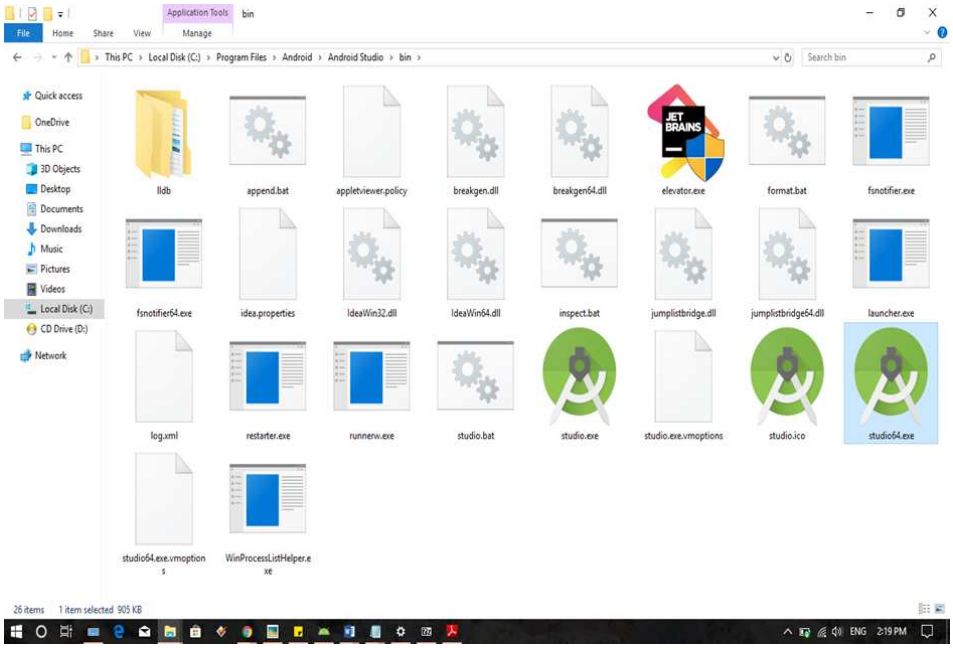

FIGURE 4 The Android Studio executable file.

4.5 | Steps Involved

We followed a general development methodology for the Smart Attendance Management System. The methodology follows a traditional software development life cycle and consists mainly of four steps: downloading the development environment, creating a project, compiling the project, and deploying the application. The steps are described in more detail below.

1. We set up the development environment. For this purpose, we choose the Android Studio IDE (Figure 4). This environment was selected because of its compatibility with the mobile application we would like to develop. It is robust and equipped with rich modules that developers can use.

2. We created a new project (Figure 5) and named it "Home Automation". We used the Model-View-Controller architecture for the application implementation.

3. The built codes are tested using and later compiled using gradle to produce an apps (Figure 6 7 8 9 10).

4. We used the IDE for building the project and produced the deployable application (Figure 11).

4.6 | Hardware Interface

We deployed the mobile application to the device using general means. This involves using a physical component, such as a USB cable, and setting the mobile device to allow the developer access to the mobile file system. The application could also be published through a store like Google Play.

4.7 | Android Device Using MediaCable

An Android device with version 9.0 (Pie) was used to execute the compiled program written in the Android studio mentioned earlier. The code written in the Android studio is then built and executed on the Android device using a media cable, as shown below. Figure 12

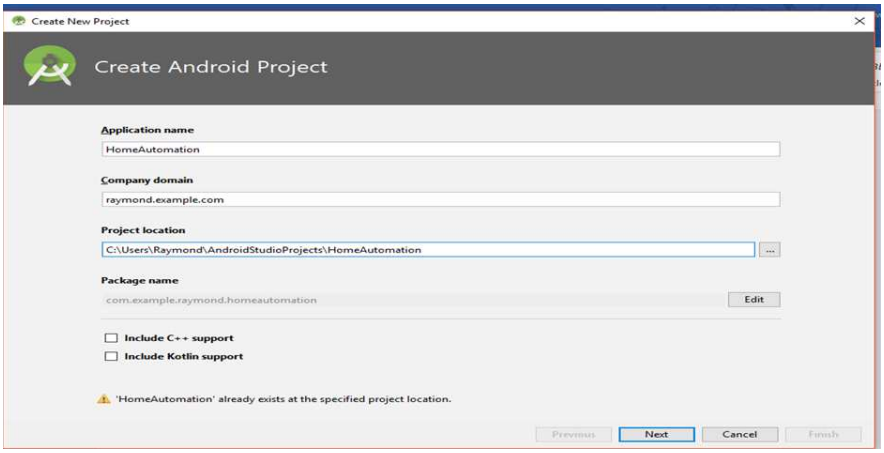

FIGURE 5 The Android Studio creates a new project page.

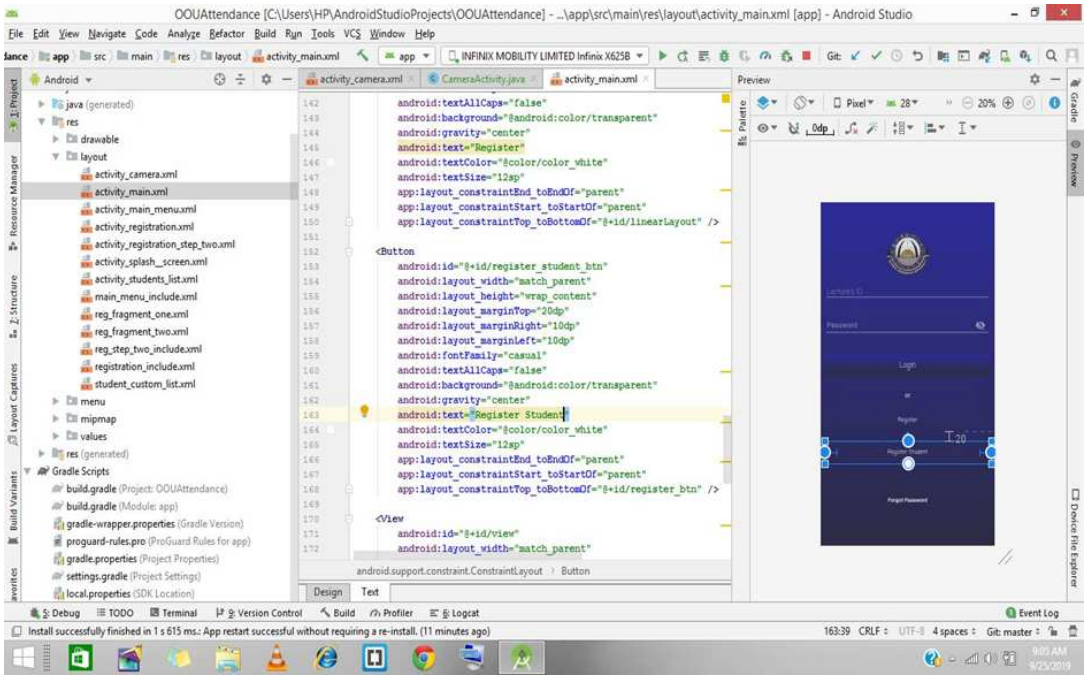

FIGURE 6 The Main Activity java page.

4.8 | Testing

After implementing the project, we tested the application on the production environment using controlled parameters. We designed four independent testing scenarios to provide four different scenarios, i.e., (1) individuals far from the device, (2) different distances between individuals from the device, (3) individuals close to the device, and (4) backlight situation. The first three scenarios aim to provide evidence regarding how far the object must be from the device to be detected by the system. Scenarios 2 and 3 also aim to provide evidence that the application can detect multiple faces. The fourth scenario aims to provide robustness of the device when dealing with noise. In this situation, the noise is the backlight.

The result of the first testing scenario shows that zero faces were recognized in the facial recognition capture (Figure 13 a). This problem may arise because of the distance barrier. Due to the gap between the faces and the capture system, it could not recognize the faces.

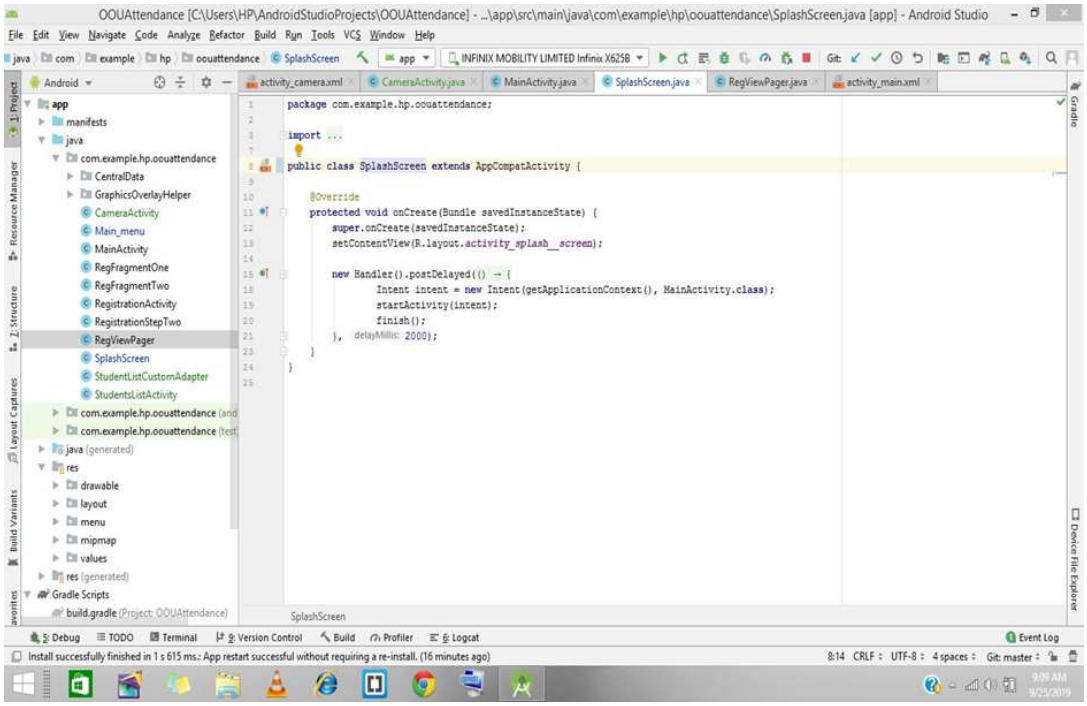

FIGURE 7 The attendance/splash screen page.

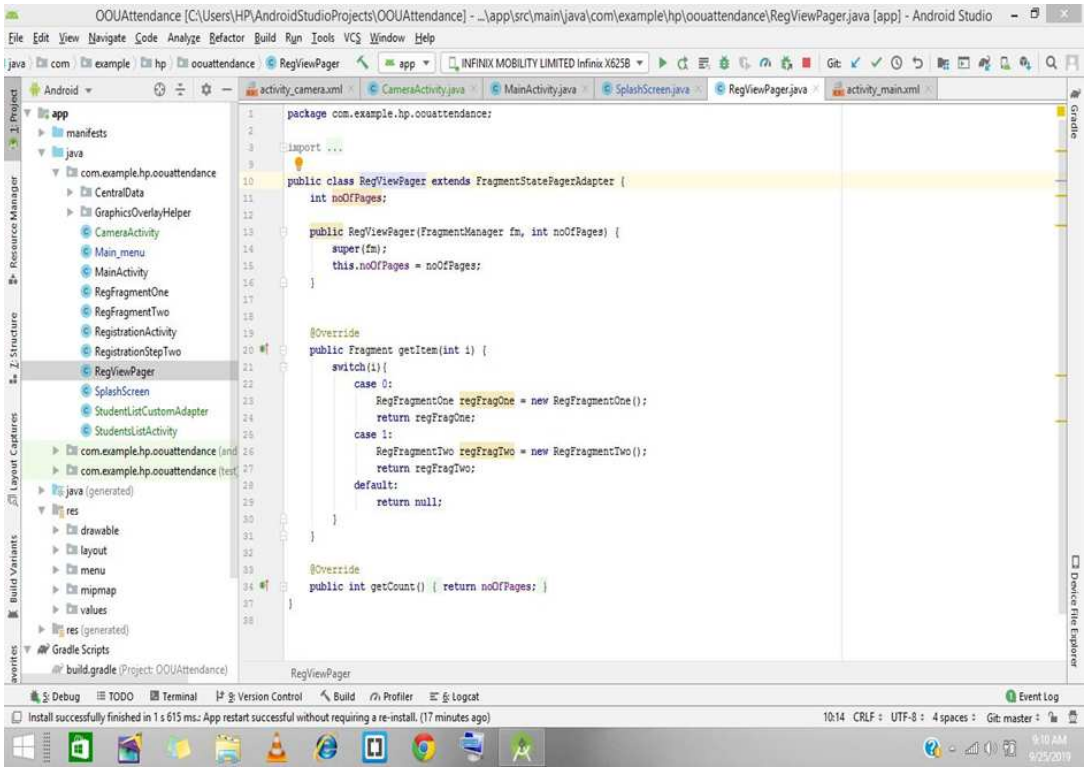

FIGURE 8 The attendance/Regview Pager page.

The result of the second testing scenario shows that only one face was captured by the device. The individuals who stand far in the distance could not be captured. This problem may arise because the recognition system can only recognize faces and not other sides of the human head. It scans to ensure it is the face of the person it captures (Figure 13 b).

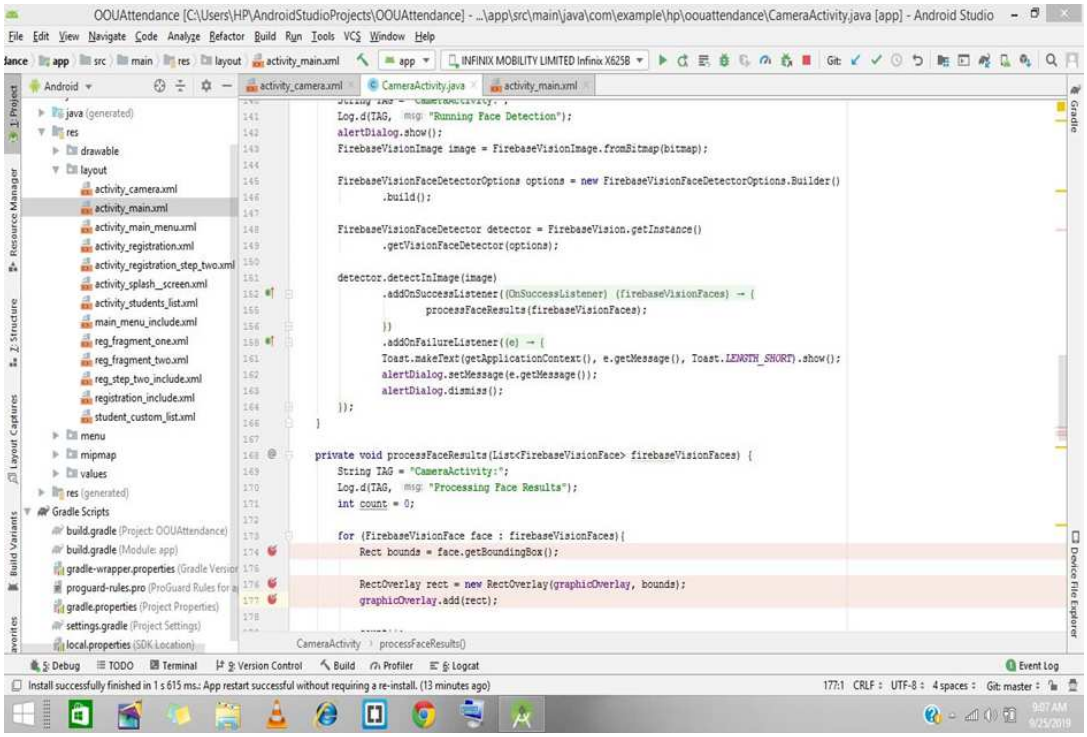

FIGURE 9 Camera Activity page

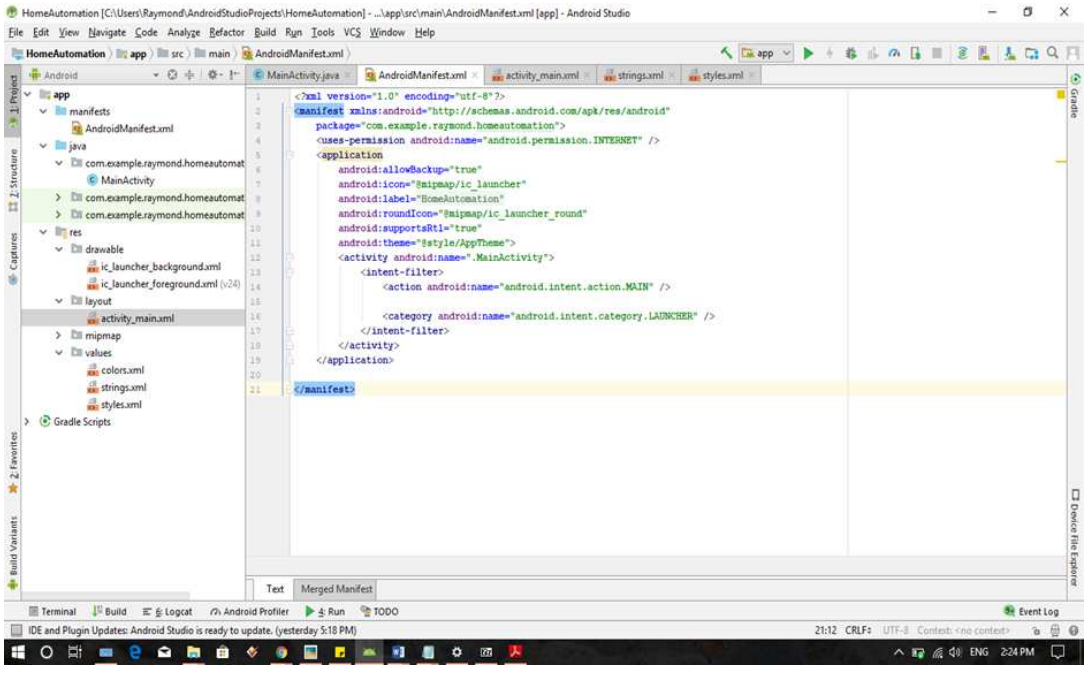

FIGURE 10 Oou Attendance/Main Activity page

The third testing scenario shows that the device successfully captured all the faces within the area. This is because all detected individuals were close enough to the device and facing the camera.

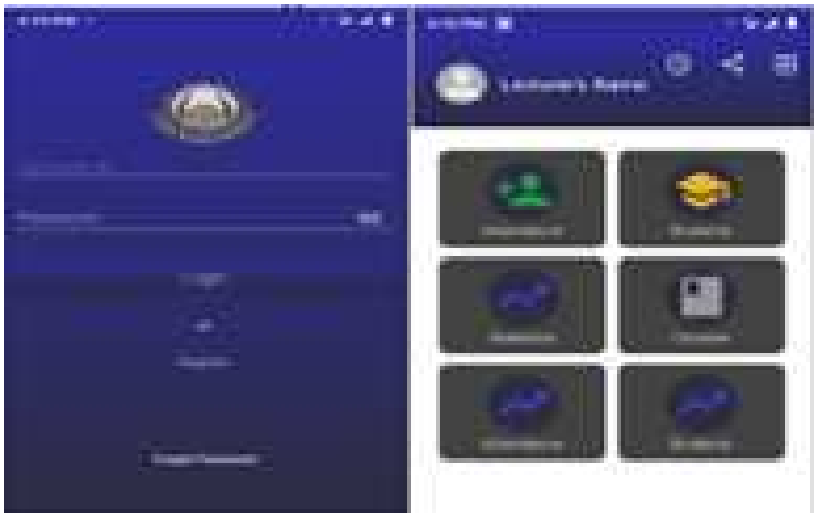

FIGURE 11 Android Application interface.

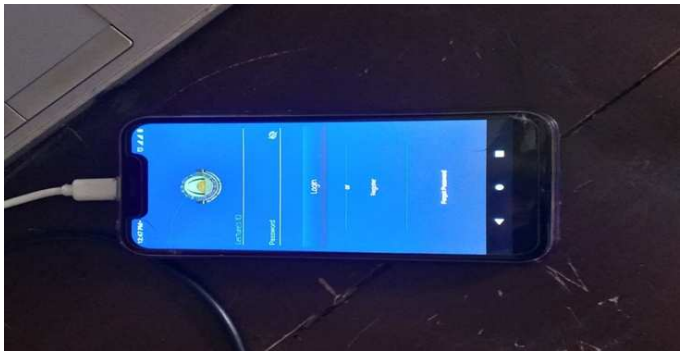

FIGURE 12 Execution of the compiled Program on Android Device.

The result of the second testing scenario shows that the device could not capture any face. This is because individuals are between the device and the external light source. This situation, called backlight, often happens when we do not have ambient light, i.e., there is a one-direction light source (Figure 13 d).

These results indicate that the face detection method still has weaknesses. In this case, the weaknesses are distance and noise (brightness). Better image processing methods can solve this weakness.

5 | CONCLUSION

A Smart Attendance System (SAS) has been developed with an Android Application program and executed through an Android device. It can automatically record students' attendance. The system has been developed to ensure a smart and efficient way of recording attendance in schools and learning institutions. This will improve the accuracy of statistical analyses conducted by lecturers and various heads of institutions and remedy the lapses that occur with manual attendance recording.

This study demonstrated the purpose and benefits of smart attendance system applications. This system is also useful for people with disabilities and older people as they can easily and comfortably control lighting, temperature, and home appliances. The android-based attendance application was developed and tested to measure its effectiveness in performing tasks and operation efficiency. The effectiveness showed that the Android-based application can perform to an appreciable level required. Some features were not implemented due to tie and equipment restrictions.

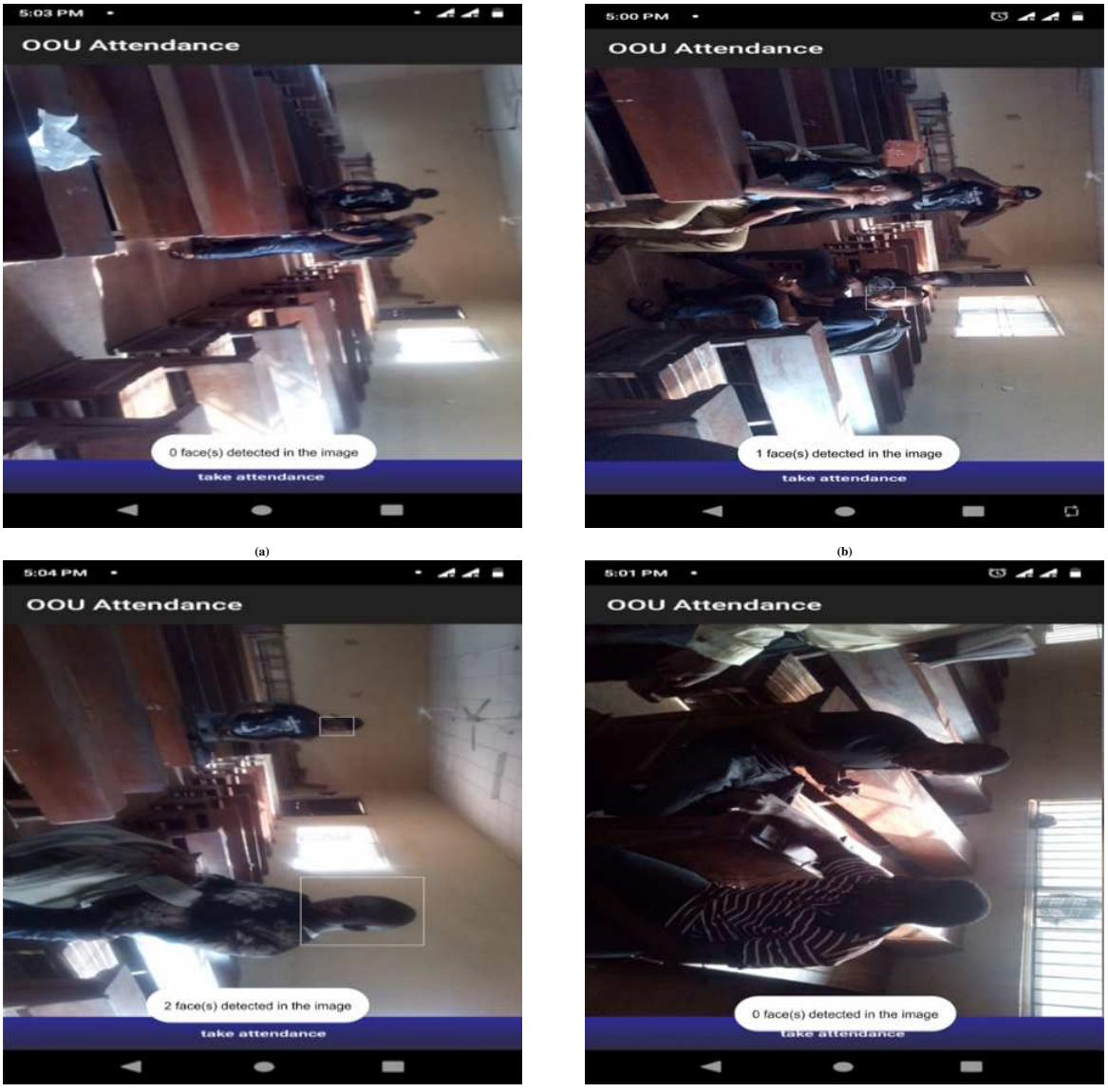

FIGURE 13 The result of four different testings: (a) testing-1, (b) testing-2, (c) testing-3 and, (d) testing-4.

The system's efficiency was measured to be very high. It performed with a lower error rate in the worst-case scenarios (which may include but are not limited to, poor lighting and camera vision) and with perfect efficiency in the best-case scenarios. This proves that the application is suitable for use in everyday scenarios. This study provides the foundations for further research and development through the developed application.

This system is recommended for every organization and institution that has to do with having a large number of people converging under the tutelage of such an organization, either for learning or other specified reasons. The technology used for this device can take face detection; if more research and testing is put into it, it will be more accurate, robust, and smarter.

CREDIT

Frank Onaifo: Conceptualization, Methodology, Resources, Data Curation, Writing - Original Draft, and Visualization. **Alexander Akpofure Okandeji:** Methodology, Validation, Writing - Review & Editing, and Supervision. **Alex Eluro:** Validation, Investigation, and Formal Analysis. **Francis Owolabi:** Validation, Investigation, Review & Editing and Formal Analysis.

References

1. Badhe S, Chaudhari K, Kale S, Mane T. Smart attendance management system using AI. International Journal of Computer Applications 2016;4(4):10–14. https://www.ijcaonline.org/proceedings/ncacit2016/number7/24739-3098/.

2. Chew CB, Mahinderjit-Singh M, Wei KC, Sheng TW, Husin MH, Malim N. Sensors-Enabled smart attendance systems using NFC and RFID Technologies. International Journal of new Computer Architectures and Their Applications 2015;5(1):19–28. https://www.invivotherapeutics.com/.

3. Das A, Khan HU. Security behaviors of smartphone users. Information & Computer Security 2016;24(1):116–134.

4. Al-Mallah RHA, Alhelal D, Abdulhammed R. Assas: An automatic smart students attendance system based on normalized cross-correlation. Bulletin of Electrical Engineering and Informatics 2021;10(2).

5. Dhawale SP. Face with Mask Detection and Recognition for Smart Attendance System. International Journal for Research in Applied Science and Engineering Technology 2021;9(7).

6. Raj A, Raj A, Ahmad I. Smart Attendance Monitoring System with Computer Vision Using IOT. Journal of Mobile Multimedia 2021;17(1-3).

7. Bharathy GT, Bhavanisankari S, Tamilselvi T. Smart Attendance Monitoring System using IoT and RFID. International journal of advances in engineering and management (IJAEM) 2021;3:1307.

8. Selvi1 KS, Chitrakala2 P, Jenitha AA. Attendance marking system. International journal of computer science and mobile computing 2014;3(2):337–342. https://ijcsmc.com/docs/papers/February2014/abstracts/V3I2201468.pdf.

How to cite this article: F. Onaifo, A.A. Okandeji, A. Eluro, F. Owolabi (2024), Smart Attendance Management System, *IPTEK The Journal of Technology and Science*, *35(1):48-59*.