Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 11, November-2019

Android based Cost-Efficient AI Attendance System

With Face Recogniton, Web Dashboard and ML based Fraud Detection

Ashwin Kumar U
Department of Computer Science Engineering
St. Joseph's College of Engineering,
Chennai, India

***Abstract*— 'Mark ME!' is an attendance system with enough flexibility in attendance marking through fingerprint verification, facial recognition, QR code scanning and NFC scanning. It also comes with a dedicated web dashboard that allows the manager or the institution to view attendances,automated e-mail alert delivery system and a machine learning based proxy attendance detection system. The app is also reverse compatible, meaning that once it is installed on an android phone, it can either act like a traditional attendance scanner(Reader mode) or in mobile mode, where the NFC tag or QR code is attached on each desk in the work place and the user with the app can scan it to mark the attendance.**

***Keywords*— Machine learning, cloud, facial recognition, NFC, QR, android**

I. INTRODUCTION

Attendance is a critical part of every institution- from schools to industries to even hospitals. But in some cases, the cost for establishing such a system will go as high as $1000, which includes buying scanners and dedicated servers. This becomes a massive challenge for startups, developing and underdeveloped countries. With our app, we aim to minimize the cost for such a sophisticated system with enough flexibility for preferred detection method like fingerprint verification, facial recognition, QR code scanning and NFC scanning, and is also reverse compatible. The dedicated dashboard can be used to track the attendance and comes with a machine learning based algorithm to point out proxy attendance.

II. EXISTING SYSTEM

A. In educational institutions

In schools and colleges, traditional register-based attendances are used which are prone to more human errors. Moreover, these systems also have proxy attendance problem where the attendance of one student is faked by another.

B. In offices and other institutions

In offices and other institutions (such as hospitals, factories) fingerprint or RF ID based attendances are used. These systems need a dozen of entry machines and dedicated server which prove to be costly for such a simple task. Also, the data is under risk from hackers and hardware failure.

III. PROPOSED SYSTEM

The proposed system consists of the following modules:

1.The android app: To record the attendances using NFC, QR and face recognitions
- 1.1 Reader mode
  - 1.1.1 NFC tags
  - 1.1.2 Face recognition
- 1.2 Mobile mode
  - 1.2.1 Marking attendances using NFC tags
  - 1.2.2 Marking attendances using QR code scanning
  - 1.2.3 Fingerprint security

2. Web dashboard: To view and manage attendances
- 2.1 Monthly attendance trends
- 2.2 Machine learning based fraud detection
- 2.3 Cloud plarform
- 2.4 Database
- 2.5 Automated e-mail alert delivery system

1. Android App

The android app is built using android studio with a target SDK version of 28. The app has two sub-apps namely- MarkME! and MarkME!-mobile. Each user assigned with a unique ID, which is generated by the institution. The user logins are managed by the Google Firebase SDK. While the MarkME! sub-app is used to run the system in reader mode, where the android phone acts like a traditional attendance machine and is available only to administrators, the MarkME! mobile sub-app is installed on the user's phone and enables them to mark their attendances from their phone itself.

The app displays a countdown for the remaining time till the attendance closes down. It also displays the people who have checked in recently

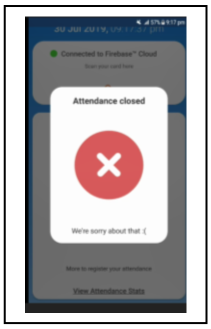

30 Jul 2019, 09:17:37 pm
Connected to Firebase™ Cloud
Scan your card here
Attendance closed
We're sorry about that :(
More to register your attendance
View Attendance Stats

Fig. 1. Attendance denied since the individual came late.

1.1 Reader Mode

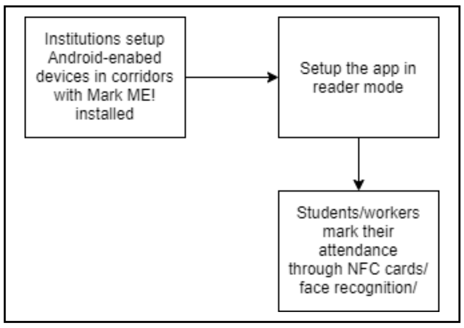

Institutions setup Android-enabed devices in corridors with Mark ME! installed
Setup the app in reader mode
Students/workers mark their attendance through NFC cards/ face recognition/

Fig. 2. Flowchart of how ‘reader mode’ works.

This mode resembles the traditional attendance machine and is used in institutions where the usage of gadgets is strictly prohibited. An android phone or an android enabled *raspberry-pi* with an NFC scanner [1] module or a high-res camera (or both) is placed at the entrance. The individual is required to use an identity card with an NFC sticker attached to it to validate their attendance.
The face recognition feature can be used in sync to improve the degree of authenticity.

The app also allows to view live stats of attendance complete with the number of students present.

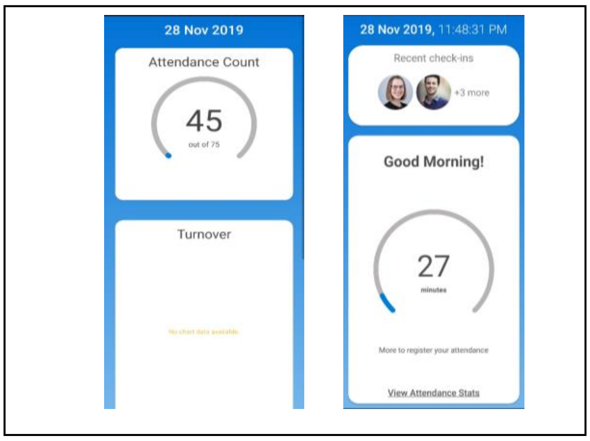


Fig. 3. Interface of the app running in “reader” mode.

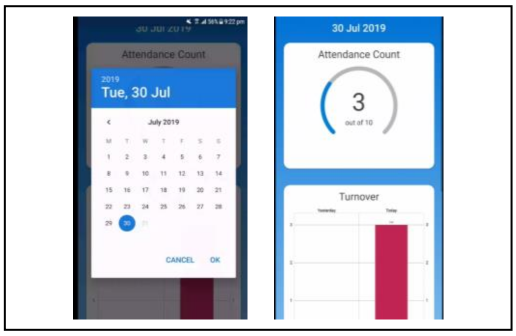


Fig. 4. Live stat checking in the app.

1.1.1 NFC tags

The NFC tags used in this system is NTAG213 manufactured by NXP semiconductors. It supports contactless transmission of data and supply energy with an operating frequency of 13.56 MHz. The maximum operating distance is up to 100mm with a data transfer rate of 106 Kbits/s. It also supports a UID ASCII mirror for automatic serialization of NDEF messages. The input capacitance is 50pF.

These NFC stickers are stuck/embedded into the individual’s ID card. To mark the attendance, he/she can tap on the android enabled raspberry-pi device or an android phone.

The *android.nfc* package is used to read the NFC tags.

1.1.2 Face Recognition [2]

(i) Extracting face embeddings from dataset using OpenCV:

The first step with extracting the embeddings is resizing the image file. All the image files in the dataset are uniformly resized to a width of 600px while maintaining aspect ratio. We than use openCV’s deep learning-based face detector to localize faces from the images. We pass the image’s corresponding blob through a Convolutional Neural Network, to obtain a 128-D embedding of the face in the image. Since doing this is a memory consuming process, the embeddings are saved to a python pickle file so that, these embeddings can be used later.

(ii) Training the Support Vector Machine:
A Support Vector Machine (SVM) [3] is a discriminative classifier formally defined by a separating hyperplane. Here, scikit-learn’s implementation of Support Vector Machine is used. The embeddings extracted from the previous process is used to fit the Support Vector Machine. The trained model is then saved to a pickle file for later use. SVM is used here because, it works well with unstructured and semi-structured data like images.

(iii) Prediction:
The android device/android enabled raspberry-pi device captures the photo of the individual and sends it to a local server/computer for processing. Similar to the previous processes, the encodings are generated for the photo of the individual and predicts it with the SVM model. The system uses a strict minimum confidence threshold level of 0.7

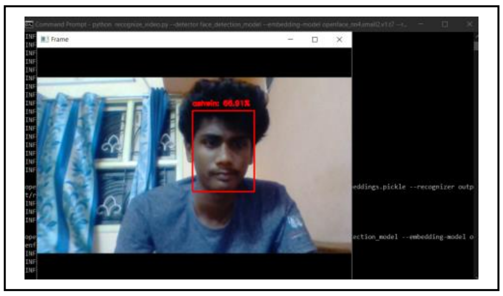


Fig. 5. Model of the face recognition system.

Once the face recognition process is verified, the attendance is marked for the particular person.

1.2 Mobile Mode

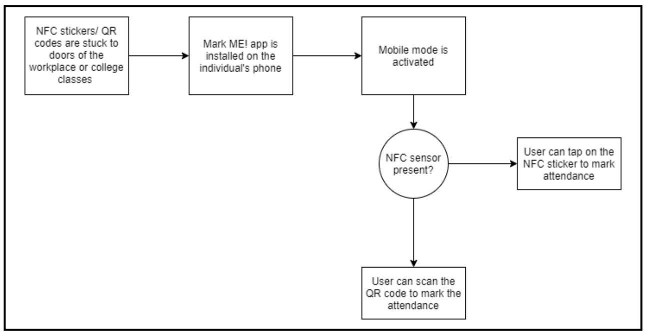

NFC stickers/ QR codes are stuck to doors of the workplace or college classes
Mark ME! app is installed on the individual's phone
Mobile mode is activated
NFC sensor present?
User can tap on the NFC sticker to mark attendance
User can scan the QR code to mark the attendance

Fig. 6. Flowchart for working of the mobile mode.

In this mode, the users can mark their attendances right from their android phone while maintaining authenticity at the same time.

This mode also has an additional feature that allows the user to view his/her attendance summary.

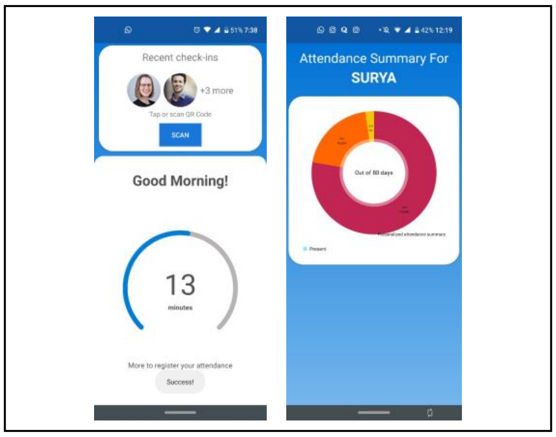


Fig. 7. Interface of the app in "mobile mode".

1.2.1 Marking attendance using NFC stickers:

A NFC tag is attached every desk (or at the entrance) of the institution. When the user registers to the app, his/her device will be assigned a unique code. When the user wants to mark his/her attendance he must tap his/her android device on the NFC stickers. The app then prompts the user to place his fingerprint for fingerprint authentication. Once authenticated, the attendance is marked for the user. Attendance can be carried out on a hourly basis too.

1.2.2 Marking attendance using QR code:

For phones that doesn't have the NFC function, QR code scanning [4] can be used. Similar to marking attendance using NFC tag, when the user wants to mark his/her attendance, he/ she must scan the QR code that's stuck on the desk. After fingerprint authentication, the attendance is marked for the user. The QR code scanner for android is implemented through ZXing android library, that provides facilities to scan QR codes and bar codes.

The scanned code is then written to the Firestore DB with the following code:

```
Map<String, Object> user = new HashMap<>();
user.put("UID", strDate);
user.put("ROLL", res);
user.put("BUS", (int) (Math.random() * 29) + 2);
fb.collection("Attendance").document("x").collection(format
tedDate).document("312318104027").set(user,
SetOptions.merge()).addOnSuccessListener(new
OnSuccessListener<Void>() {
    @Override
    public void onSuccess(Void aVoid) {
     Log.d("WRITEDB","Document successfully written!");
     int present = 0;
     Log.d("STATS", present + "");
     load.dismiss();
    }
        }).addOnFailureListener(new OnFailureListener(){
         @Override
         public void onFailure(@NonNull Exception e) {
           Log.d("WRITEDB", "Error writing document", e);
         }
});
```

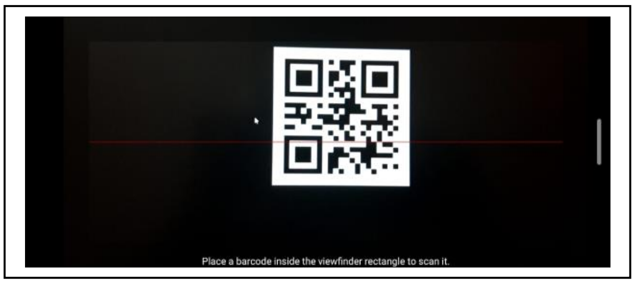


Fig. 8. A student scanning a QR code to mark his/her attendance.

1.2.3 Fingerprint security

Since these attendances could be potentially proxied by the user's friend, wherein the user hand overs his/her phone to a friend, and make them to scan the NFC/QR code, fingerprint authentication is used. A FingerprintHandler class is created extending the *FingerprintManager.AuthenticationCallback*

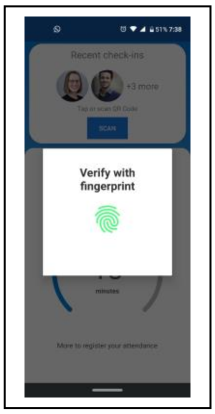


Fig. 9. App asking for fingerprint verification to verify attendance.

2. Web dashboard

The Web dashboard is used to monitor the class attendance and view the attendance list of any date. It is built using Bootstrap, jQuery and flask. Bootstrap [5] and jQuery are used to build the UI of the website dashboard and gives it a neat and professional look. Flask is a python web framework and is based on the Werkzeug WSGI toolkit and Jinja2 template engine. It uses the same FirestoreDB to retrieve the attendance list, which was inserted by the mobile app.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 11, November-2019

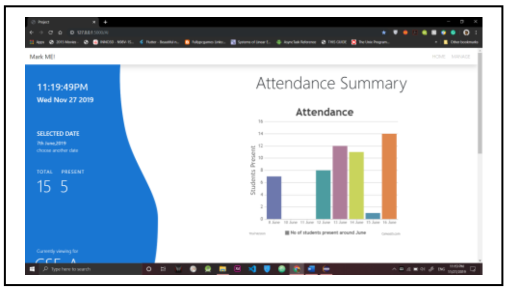
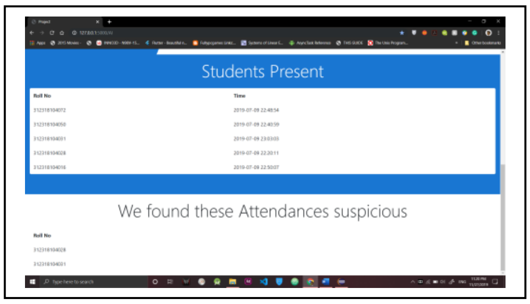


Fig. 10. The web dashboard displaying the attendance summary.



Roll No	Time
312318104072	2019-07-09 22:48:54
312318104050	2019-07-09 22:40:59
312318104031	2019-07-09 23:03:03
312318104028	2019-07-09 22:20:11
312318104016	2019-07-09 22:50:07

Roll No
312318104028
312318104031

Fig. 11. Web dashboard displaying induividuals present on a particular day and attendances found suspicious using machine learning algorithm.

2.1 Monthly attendance trends

The web dashboard uses canvasJS, a data visualization tool to provide a histogram of the attendance summary over the year. We can view the attendance summary of any month by clicking on the chart, which opens a modal. The desired month is chosen, the graph updates itself with the appropriate values.

2.2 Machine learning based fraud detection

Apart from the face recognition-based fraud detection, the MarkME attendance system overcomes one of the most difficult hurdle of fraud attendances by using Machine learning [6]. Note that this feature is highly confined to specific institutions especially colleges, where students arrive either in their own means of transportation or college buses. The dependent variable is taken as the entry time into the college and the variable to be predicted is the sign-in time.

TABLE I. HEAD OF THE DATA

ENTRY	CHECK-IN
0	30
2	44
4	49
6	39

The entry time refers to the minutes past the time when attendances were opened (For example, if attendances were opened by 7:00AM and the individual checks in at 7:04, then the entry is entered as ‘4’). This time is mostly calculated from the time when the school/college bus entered the campus (or) when the individual parks his/her vehicle in the campus Similarly, a pattern is recorded for each individual. The individual, after entering the school/ college may leave for having breakfast or other work before actually marking the attendance. This pattern is usually recurrent in most of the individuals, and by using Linear Regression, we predict a check-in time for when the entry is made. If the actual check in time is not near from what is predicted, the attendance is marked suspicious and he/ she must re-verify the attendance using fingerprint/facial recognition.

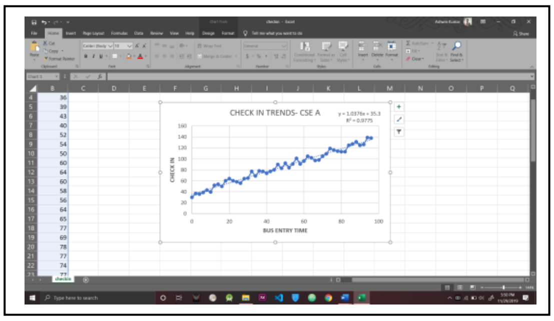


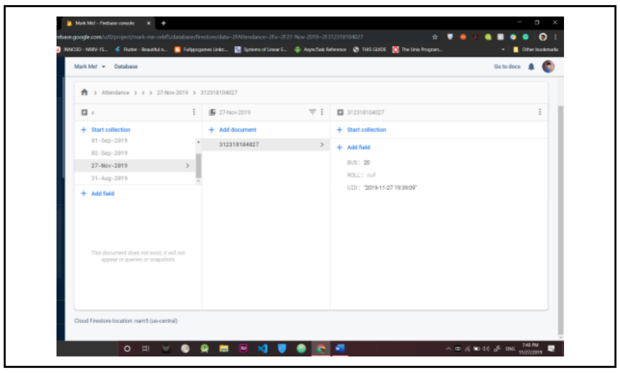
Fig. 12. Graphical representation of the data fit into our LinearRegression model.

2.3 Cloud platform

The app and the web dashboard both use the Google Firebase cloud platform. Firebase is a Backend-as-a-Service. It provides several features like Authentication, cloud database(firestoreDB), cloud storage, hosting, serverless-event driven backend functions and an SDK for common ML tasks like object detection.

2.4 Database

The database used in this project is Firestore DB, a no-sql database. In a no-sql database like FirestoreDB, data is stored in documents, which inturn are stored in collections. A document can contain another collection within it. Data is organized in the form of ‘trees’.


Mark Me! – Firebase console
Mark Me! Database
Go to docs
Attendance > x > 27-Nov-2019 > 312318104027
Start collection
Add document
Add field
01-Sep-2019
02-Sep-2019
27-Nov-2019
31-Aug-2019
This document does not exist, it will not appear in queries or snapshots
Cloud Firestore location: nam5 (us-central)

BUS: 20
ROLL: null
UID: "2019-11-27 19:39:09"

Fig. 13. Data displayed in the FirestoreDB console.

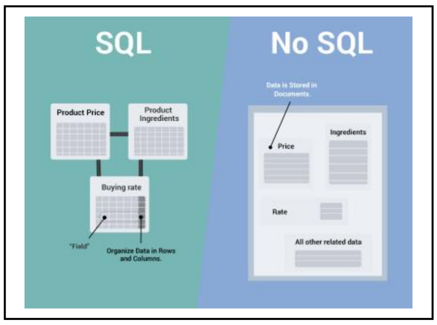


Fig. 14. Differentiation between a SQL and a No-SQL database.

2.5 Automated e-mail alert delivery system

This system also allows the institution to send mails automatically to the students/ workers informing them about their absence. It used smtplib python library to define a SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon. A simplified example of how this works is given below:

```
import smtplib

toaddrs= 'ajayyasodha@gmail.com'
fromaddr  = 'haniel20008@gmail.com'
SUBJECT="Mark ME!- Automated mail generation"
TEXT="Please ensure that you have missed your classes
today.\n STUDENT ID: Ajay Kumar \n COURSE: BE "
msg = 'Subject: {}\n\n{}'.format(SUBJECT, TEXT)

username = 'haniel20008@gmail.com'
password = '******** '

server = smtplib.SMTP('smtp.gmail.com','587')
server.starttls()
server.login(username,password)
server.sendmail(fromaddr, toaddrs, msg)
server.quit()
```

IV. ADVANTAGES

- It is more cost efficient when compared to the existing system.
- It uses future-ready technologies.
- It is flexible in operation to suit the institution’s need.

V. CONCLUSION

The app aims to provide a low cost-yet-versatile attendance system. It also uses future ready technologies such as cloud storage, face recognition and many more. With the current growth rate of AI and the number of start-ups, industries increasing day by day, this app will sure to reduce some operating costs without compromising on quality.

REFERENCES

[1] Ortiz, C. Enrique. "An Introduction to Near-Field Communication and the Contactless Communication API", June 2006
[2] Shervin Emami, Valentin Petrut Suciu. “Facial Recognition using OpenCV”, Journal of Mobile, Embedded and Distributed Systems(JMEDS),2014.
[3] Cristianini, N. and Shawe-Taylor, “An Introduction to Support Vector Machines”, Cambridge University Press,2000.
[4] "QR Code—About 2D Code", Denso-Wave. Retrieved from https://www.qrcode.com/en/about/
[5] “About Bootstrap”, Bootstrap. Retrieved from https://getbootstrap.com/docs/4.4/about/overview/
[6] Alpaydin, Ethem. “Introduction to Machine Learning”, The MIT Press, 2010.
[7] "NoSQL for the serverless age: Announcing Cloud Firestore general availability and updates", Google Cloud Blog.