Pi Attendance Management

Karan Jadhav
Masters of Computer Science
North Carolina State University
Raleigh, North Carolina
Email: kjhadav@ncsu.edu

Samantha Scoggins
Masters of Computer Science
North Carolina State University
Raleigh, North Carolina
Email: smscoggi@ncsu.edu

Qaiss Alokozai
Masters of Computer Science
North Carolina State University
Raleigh, North Carolina
Email: qalokoz@ncsu.edu

***Abstract*—Attendance checking is done to keep track of the targeted user group and ensure participation. Attendance and its management can be done manually as well as with the help of technical solutions. In this report, we aim to improve attendance management in terms of reduced time required, reduced errors and eliminating the possibility of impersonation. We developed three solutions for attendance management as part our project and then conducted user evaluations to help us compare the three solutions and report the best of the three based on user feedback and performance measurement.**

***Index Terms*—Attendance Management, NFC, RFID, Blutooth, biometric, facial recognition attendance, error prone,.**

I. Introduction

In any organization attendance is primarily taken to ensure participation of the user group. In case of educational institutes, the reason for this is to make sure that students attend classes regularly so as to learn effectively. The reason, in case of professional organizations, is to ensure that employees aren't missing work and for increased productivity.

We highlighted the need for an effective attendance management solution that is not time consuming, eliminates the possibility of errors impersonation, is accessible and effortless in record keeping, in our January report. We looked at and researched on several platforms on which to develop our attendance management solution. After much deliberation, we decided to develop solutions in the form of Android and Java desktop applications.

We chose Android because of the widespread and common use of Smartphones by people today, especially students. In addition to this, it makes for a very handy, accessible, and lightweight device to be used for this purpose. The reason we also chose to develop two of our solutions as web applications, is to have even more accessibility compared to smartphones. Java applications have high portability in which many systems support the Java Virtual Machine. An added advantage of this is being able to have solutions that have a greater complexity of features and interaction, something that might not be as easy to implement for a smartphone app.

After this, we turned our attention to the possible solutions we could implement. We found the field of biometrics[reference] to be a good field to choose an implementation from. This is because they eliminate the possibility of impersonation and are generally quite fast and hence chose facial recognition and voice recognition as two implementations. The third solution we chose was a live quiz based one. This was chosen as it virtually requires no time because it makes use of the lecture time for engagement using quizzes, which facilitates learning while doing attendance recording in the same time period.

The first solution is called Farecat, short for facial recognition attendance. It has three features – enrollment, recognition and viewing attendance. It requires a picture of the target to be taken during enrollment and for marking attendance, i.e., the recognition feature. Attendance can be viewed by date.

The second solution was the voice recognition based one, VoiPi. It has features to enroll and recognize students, same as the first solution. Participants must record a sample of their voice during registration and provide a voice sample every time they need to be marked for attendance.

The last solution was the live quiz based one, Tendance. It has separate features for the lecturer and students, unlike the first two solutions which target all usage from the lecturer's account. The lecturer can schedule quizzes for a subject for a later time, which the students must complete and submit at the specified time.

In subsequent sections of this report we will provide details for each of the solutions, the evaluation we used for feedback, the target audience and the results of our experiment analysis.

II. Synopsis

A. Previous Work

Conducting research through literature and surveys, we determined that a key problem with taking attendance was with how much time was wasted with manual logging. Survey data coming from both professors and students indicated a solution would need to decrease the time needed to take attendance. However, there were some comments indicating the potential to make use of the time attendance takes, for example, use with quizzes or for cold calling students. In addition, mobility combined with a way to decrease attendance errors and cheating were needed.

We found many different technologies that have been used, tested, or have potential for addressing issues like time and mobility. We could classify many of these as using Student ID cards and a card reader, Barcode and similar coding, Biometrics, One time Passwords and Near Field Communication (NFC)[1]. Biometrics seemed to be an interesting and efficient way to help prevent cheating, errors, and time consumption.

Examples of these are Facial and Voice recognition. There was an experiment done to test the effectiveness of a selfie and signature based mobile program where students passed a mobile device around the class room. The results from that experiment showed that it was "more interesting, convenient, fresh innovative, and special [2]. So, we thought of creating a similar system that uses facial recognition only because signatures could also be open to cheating. A program using voice recognition could also have potential in the same way.

Our goals were to create solutions based on the previous implementations found to address attendance issues improving on usability, timing, and cheating. However, with the plethora of options from which to focus on and develop solutions. Finally, we decided to develop three programs. Two of which focused on Biometrics with facial and voice recognition. The third, went in a different direction and is based on quiz taking for attendance.

B. Features

Farecat

Farecat or facial recognition attendance is the first implementation. It was implemented as an Android app. The reason for this was the availability of a camera on smartphones and also because it is a very handy, accessible way of marking attendance. It only has an instructor account, not a student account. The main reason for this was that student's should be able to mark attendance only from the instructor's device to ensure their presence in class and to avoid cheating.

There are three features in this application. The first is enrollment of a student. The student's name needs to be entered and a picture taken and the application will register the student. A check was used where a picture of a person cannot be used to enroll a student under a different student name. The second feature is recognition or in other words, marking attendance. In this case students only need to take a picture of themselves and the application will recognize them based on their enrollment picture and mark them present. The last feature was a simplistic viewing of attendance records. An SQLite database was used, which uses device storage, to store attendance records for students. The instructor can view a list of present students by date.

This implementation was made possible by the use of the Kairos[3] facial recognition API. Kairos is a REST based API that is very simple to use. As stated before, enrollment and recognition just require two and one parameters in the request body and another parameter called gallery name. This is just a grouping of a number of subjects. Multiple galleries can be created to segregate people, if required.

VoiPi

The voice recognition system is focused on enhancing the capabilities of the teachers/instructors to take effective and efficient attendance in class. Another focus was to address students cheating with another student so they can skip class and be counted as present. Pi-Voice is based on a simple design with a user-friendly interface. The system comprises of an interface with a functionality such as recording the student's records, capturing voice commands, searching and verification of voice samples, and reporting on daily or weekly basis from database. The user (instructor) first enters all the information of the students which is saved into a database (mysql) along with the voice data. The daily attendance button will ask for user name as well as voice and that will in turn be compared to the existing data (voice) in the database through musicg API(4). The API compared the bitwise information (frequency equivalence) of the voice to what is saved in the database and shows a message of present or not present through a message box.

Tendance

Tendance is a Java desktop application that tracks attendance through quiz taking. It allows professors to set up quizzes ahead of time by date, as well as search for and view previous quiz "attendance" and results. When setting up a quiz for a particular date, the professor can set the start time, duration (max 75 minutes), and type of quiz, Default and Added questions. The default quiz consists of a simple button that logs student attendance; it is also included in the Added questions type. Added questions allows professors to set up many discussion and multiple choice questions. When searching for set up quizzes, all questions, which may include the different choices and a right answer, can be seen as well. A professor cannot create more than one quiz for a specific date, and they cannot create quizzes for past dates.

Logged in as either a student or professor, it is an easy switch between classes to check the availability and timing of a quiz taking place on that current date. When a quiz is open, students simply click one button to take it. Once taken, the student gets confirmation that the quiz was taken on the home screen. The database stores all student answers and logs the student's attendance as present for that date. When a professor goes to look at the results of the quiz, a list of all students who have taken the quiz shows up. Clicking one button randomly selects a student from that list that allows for cold calling students for discussion. Also, all answers submitted by students anonymously show up sorted by question in a list so the professor can see and discuss the responses with the class. This data can be seen for any quiz that has already been closed as well.

This implementation was developed in Java using Netbeans IDE, dependent on a MySQL database using 10 relational tables. While these features do not address decreasing the time needed to take attendance, it makes use of the time needed for use in class. It can be used primarily for class discussion purposes. Secondly, it gives the professor flexibility to use the quiz system to check anything they wish, like whether students payed attention to the class period before or whether the homework was done. This can also allow for attendance tracking for Distance Education students as well as it would not necessarily require the student to be in a specific location to take a quiz. Over all there are a fair amount of features in Tendance so that the professor can effectively make use of the time needed to take attendance.

III. Experiment

A. Methodology

To check the effectiveness of our solutions we followed a two-tiered approach. We asked the users to use the features of each implementation one by one and then asked them to fill a feedback form, rating each of them on a scale of 1-5 on common criteria. The following were the questions asked to the participants:

1. Rate each implementation on a scale of 1-5 on the following criteria
a. User Interface
b. Usability
c. Central idea
d. Effectiveness
e. Overall rating
2. Positive aspects of each implementation
3. Negative aspects of each implement. Areas of improvement

Parallelly as the users evaluated the solutions, we measured the time taken to use the two main features - enrollment and registration. This was done for facial recognition and voice recognition but not for the live-quiz implementation. The reason for this was that recording attendance in case of the quiz solution would depend on the length of the quiz set by the lecturer, so it would not be fair to judge it on this criterion.

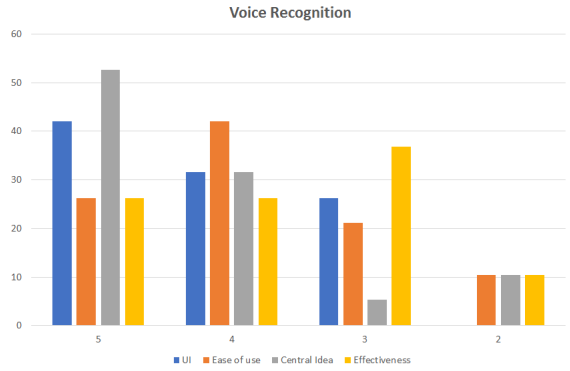
B. Participants

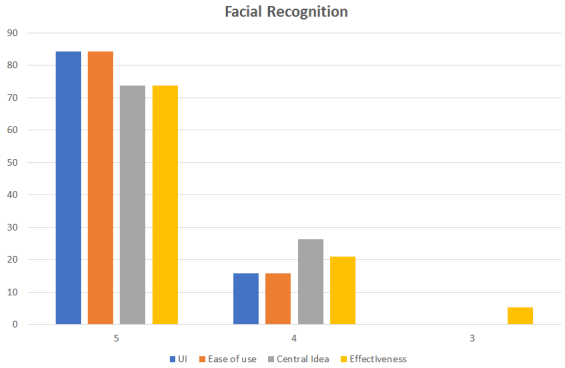
The participants were selected on a first come first serve basis. They were all graduate students in Computer Science at North Carolina State University and one student from Civil Engineering. Prior experience was not required of any kind as the solutions were applicable to all students.

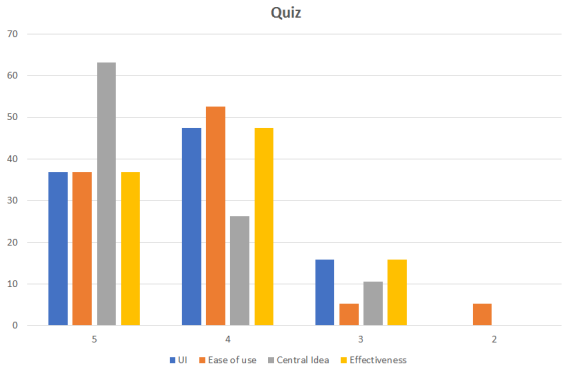
IV. Results

A. Graphs

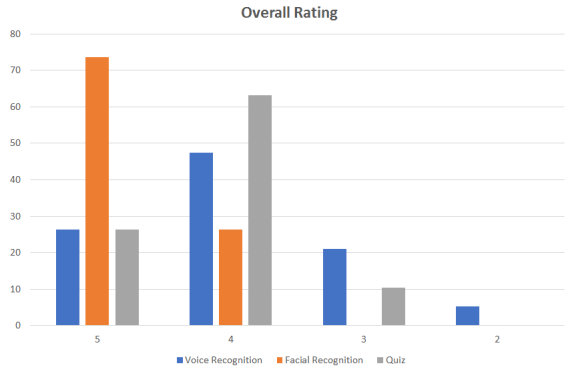
A total of 19 participants took part in the evaluation. The results of the evaluations are represented as graphs below:







The above three graphs are for voice recognition, facial recognition, and quizzes respectively. On the Y-axis is the percentage of participants and on the X-axis, is the rating given for each criterion on a scale from 1-5. The criteria are the ones mentioned in the Methodology section namely, User Interface, usability, central idea and effectiveness of the solution.



The above graph is the overall rating given to each of the implementations. On the Y-axis is the percentage of participants and on the X-axis, is the rating on a scale of 1-5.

B. Comments

We got many interesting comments on the positive aspects of the three solutions, things that the participants liked about each of them. In addition we also received criticism or suggested improvements that could be made to the solutions. A summary of these can be seen in the next section.

V. Analysis

A. Graph Analysis

We can see from the first graph that the voice recognition solution had a more spread out rating on all the criteria. About 70 percent of the participants gave it a rating of 4 and above on three of the criteria and a little over 50 percent of the participants gave it a similar rating on effectiveness. The remaining participants gave it a rating of 3 and below. Effectiveness had the lowest rating relatively, possibly because of the low accuracy of voice matching.

Compared to this, all the participants gave the facial recognition solution a rating of 4 and above on all the criteria as can be seen from the second graph. Over 70 percent of these gave it a rating of 5 on all criteria. There was just one outlier that gave it a rating of 3 on effectiveness.

From the third graph we can see that the quiz based implementation received a very favourable rating. About 85 percent of the participants gave it a rating of 4 and above on all the criteria. The remaining gave it a rating of 3 except for an outlier that gave it a rating of 2 on ease of use. We suspect this might have been because of the added functionality that this solution had compared to the other two solutions which may have made it seem more complex to the participants.

B. Comments Analysis

We received some interesting comments in the feedback, both positive and constructive. Quite a few people mentioned their appreciation about the UI, simplicity and speed of the facial recognition solution. A few others thought of voice recognition as a good idea. One person mentioned that these solutions were a fun way of marking attendance.
In terms of possible improvements, a few people suggested that voice recognition could be more accurate. A couple of others suggested that the UI for voice recognition and quizzes could be improved on, making them slightly more attractive.

C. Compare and Contrast

All three implementations, while maintaining similarities, have features that address attendance issues differently. In addition each has advantages and disadvantages. For example Farecat and VoiPi both can test for cheating attendance and can speed up attendance taking significantly. Farecat has a quick and simple interface on a mobile device that students do themselves during class. The functionality for VoiPi is similar, only, as a desktop application. A key advantage Farecat has over VoiPi, though is the accuracy with recognizing individuals. Farecat can recognize individuals in images with approximately 85 percent accuracy, as high as 95 percent; the threshold to accept recognition is set to 60 percent. VoiPi can only recognize with 35 percent accuracy, as high as 50 percent. This is primarily due to the Musicg API used(4). However, storing sensitive information such as photos of students can pose a security risk, though storage of initial photos would be secured and won't be retrieved by clients. VoiPi compares frequencies of voices and a security breach would not pose as much of a threat. Tendance is fundamentally different than both VoiPi and Farecat. With Tendance, students could take quizes for each other, and does not speed up the process as taking a long quiz can be time consuming. Instead, Tendance uses the time for class related activities like discussions and attention checking. Essentially the time for attendance taking is covered by those activities. However, it could pose the same security risks as Farecat, by the way attendance is stored and viewed. The attendance information itself could be considered sensitive. The user interface of each can be compared as well. Farecat has a smooth mobile GUI, while Tendance could be considered cluttered. Voipi has a simple interface as well, but because it is a desktop application, it could be considered more cumbersome to use than Faircat.
The technology required by each differs as well. Farecat assumes that each classroom has at least one mobile device that can be passed between students. While Tendance, and VoiPi assumes all students have access at least to a laptop or PC to take a quiz or check in with voice. Voice recording requirements also add on the need for an accessible and working microphone.

VI. Future Work

For future work, each implementation can be improved individually. However, to address the basic drawbacks of both Farcat and Tendance, they can be combined. Instead of the Default quiz button, a selfie would be required for each quiz

to log attendance. Then the student can proceed to take the quiz if there are added questions. This would solve Tendance's drawback in which students can cheat for each other, and increase the features for Farecat, where professors can then search for and set up availability for allowing selfies.

Farecat has a lot of scope for future work, beginning with course creation and segregation. Currently students are enrolled in one course. With this feature, different courses can be created and students enrolled in them accordingly. Every time an enrollment, recognition or attendance viewing activity needs to be done, a course will first need to be selected and the action performed. Attendance reporting could also be more dynamic in terms of viewing attendance on a daily, weekly or monthly basis or based off a custom date range.

The next feature that could be implemented is the creation of a super user or administrator account. This account would be affiliated with a higher institute such as a University or college. Instructor accounts would hierarchically be below this administrator account. Since enrollment is an administrative activity, it could be shifted to the administrator account along with creation of courses and their assignment to different instructors, who would be responsible only for attendance marking, as is the case in real life. The administrator account would further be able to remove students from different courses, remove entire courses and be able to view attendance for all courses.

The last feature could be the creation of a student account. As state earlier, recognition capability is restricted to the instructor account to stop students from cheating and avoid skipping class. However, in the student account students will be able to view their attendance for each dat of each course.

VoiPi has vast potential of future work. The application can be extended so that a separate email is sent to the instructor as well as to students stating that the student was present on this day of the class. Moreover, the application is open to enhancing the graphical user interface with a separate statistics portion on the GUI showing daily and weekly averages of class participation along with graphs of the total attendance.

The quiz based system, Tendance would be better suited as a Web application and not a desktop Java application. Future work will be implementing this as a web app. This will allow for more mobility and a better GUI interface using css. Added features would let students see total attendance, let professors search and see total attendance by student, not just by date, and finally allow for the grading of any multiple choice quizzes for easy grade calculation. Increased flexibility for professors to edit future quizzes are also needed.

VII. Conclusion

Overall, each implementation can improve in some way the attendance management experience from manual logging. Farecat and VoiPi drastically decreases time consumption and helps prevents cheating. Tendance makes use of that time for class related activities. Each has advantages and disadvantages. User testing revealed that the facial recognition implementation was generally rated higher in all areas: UI, Ease of Use, Central Idea, and Effectiveness. Though the Central Idea of the quiz based system was rated much higher than the voice recognition, all other areas had similar ratings between the two. These results combined with the mobile capabilities of Farecat have influenced us to determine that farecat's facial recognition is the best solution for the Attendance Management problem.

Acknowledgment

The authors would like to thank...

[**1**] Iio, J. (2016). Attendance Management System Using a Mobile Device and a Web Application. 2016 19th International Conference on Network-Based Information Systems (NBiS). doi:10.1109/nbis.2016.44

[**2**] O. Shoewu, Ph.D. and O.A. Idowu, B.Sc. Department of Electronic and Computer Engineering, Lagos State University, Epe Campus, Nigeria. Department of Electrical and Electronics, University of Benin, Edo State, Nigeria

[**3**] https://www.kairos.com/docs/api

[**4**] https://github.com/loisaidasam/musicg

Chit Numbers:
djbsoaie
djmhouui
fjllouao
ckhhoeiu
chgzuaea
cjmgioue
ddjguiao
dbhfoaui
fghtioau
Dmmvauaa
dcgnauou
gcbxueoe
dklyuiae
fdkbuiou
clkxuuao
cmhkuoie
dhbpeauo
bmbwuiee