Automated Attendance Management System

Design, Development, and Evaluation of Three Solutions *

Mohit Sardar
msardar@ncsu.edu

Saravanan Balasubramanian
sbalasu7@ncsu.edu

Siddharth Heggadahalli
ssheggad@ncsu.edu

ABSTRACT

Attendance Management is very important in educational institutes. Traditional attendance systems tend to be time-consuming, difficult to manage and not foolproof. This problem can be solved efficiently by automating the process using available technologies in smartphones. Our project aimed to design and develop an Android application for an Automated Attendance System using three different approaches, (i) Near-Field Communication (NFC) (ii) Global Positioning System (GPS) and (iii) Quick Response Code (QRC). In this paper, we present the basic methodology of the three approaches and the results of the user evaluations conducted with three solutions. The main objective is to learn how much time could be saved and security gained with an Automatic Attendance System.

1. INTRODUCTION

Attendance is mandatory in most schools and even in some courses in universities. While importance of attending classes can vary from student to student, it is very important to many instructors. According to our survey [8] of 50 students we conducted in January, 76% of the students surveyed said they attend class just to satisfy the attendance requirements for the specific course while approximately 73% of teachers/professors take attendance very seriously and in fact spend an average of 7 minutes for a class of size 50 to check attendance. The time spent for attendance is more if the class sizes are bigger.

Hence, having a poor, inefficient attendance system in place could be time-consuming and easily misused. Traditional methods such as manually calling out all the student's name, randomly checking a few students, quiz-based attendance, sign-up sheets and RFID based attendance systems all suffer from one or more of the above mentioned disadvantages.

Implementing an automatic attendance system that is unsupervised and robust would benefit professors greatly. Through the survey we identified mobile phones as the preferred device to implement attendance systems given that 98% of the students carry their mobile phones to class as opposed to only 48% who carry their laptops.[8]

Our project aimed to design and develop an Android application that implemented three different approaches to try and eliminate the problems faced in current attendance systems. The three approaches were via QR Code[13] [9], NFC[12] and GPS[10].

The application was evaluated by 23 users. Each user was asked to go through the basic use cases from both the professor and student perspectives of all three implemented approaches. They were timed for both perspectives and surveyed on the advantages and disadvantages they could think of for each of the three approaches. The results of the user evaluations are presented later in this paper.

2. METHODOLOGY

2.1 Project Software Development Model

Agile Development was used in developing this application. After reviewing various methods of development, agile was chosen because of the flexibility it offers, and and because of usable modules that are available after each stage. This was very important because of the uncertainty regarding implementation and short period of delivery time. The ease of implementation of each solution would be known in the early stages of development in this method, and suitable methods of implementation can be chosen quickly. The Programming language, frameworks and tools were chosen depending on the team members' prior knowledge and experience. At each stage of the project, work was divided among all members of the team depending on member's interest and skills. Hence there was need for continuous collaboration and communication between the members and this was greatly achieved through GitHub.[2] GitHub was helpful in setting milestones, deadlines, acquiring the required code developed by other members, Raising and solving issues and to keep track of the entire project very easily. Various GitHub resources were effectively used to complete the project successfully well within the required time.

2.2 Architecture

The application had two different class of users: Professors/Teachers and Students. Three different solutions were identified and functionality of all three methods was different for both class of users. The look & feel and navigation of the application was designed to be same for both users, but based on the credentials (username, password) the users were differentiated and different functionalities were rendered within the application.

All the three methods are independent of each other. For the sake of simplicity all the three solutions were put into

*This report is Team D's April 7 Deliverable for CSC 510 taught by Dr. Tim Menzies in the Spring of 2017 at North Carolina State University.

one application, but each solution could be developed independently as stand alone projects.

This application was developed based on MVC architecture. [11]

View and Android: The view part was developed on android platform (mobile application). The view design for intuitive and ease of use. The functionalities are: Receiving input from the user. Saving them temporarily on the device and making the necessary API calls (POST method) to the backend. Receiving the JSON response from the server and displaying it in a suitable format to the user. As the functionalities was different for different class of users. Login credentials had to be saved temporarily and sent to the backend for each call.

Server and Database: The controller part was designed on Django framework (Python).[1] It receives POST requests and upon validating the user credentials, it performs the required function and returns results (or messages) to the front end. The data is stored on a NoSQL database. The Server and Database is deployed in a Heroku cloud based container.

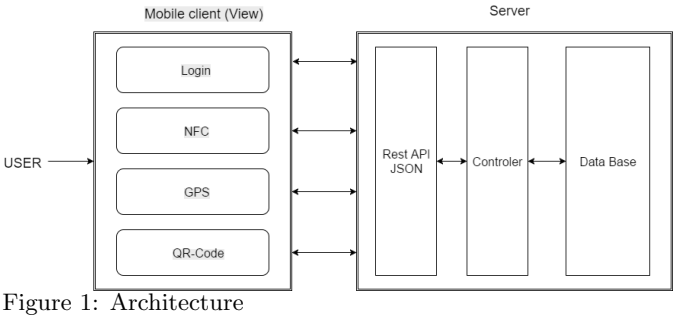

Figure 1: Architecture

3. IMPLEMENTATION

The application opens up to a login page as shown in the Figure 2, once the user enters the username and password, they are validated and username is used to identify whether the user is student or professor and the corresponding pages opens up. Here three options GPS, NFC, QR-Code are available to both users as shown in the Figure 3. For professor: On selecting any one of the options, the corresponding data is send to server and stored in the course database. For student: On selecting any one of the options, the corresponding data is sent to server and it is compared with the already stored information entered by the professor. And, a "successful" or a "failed" message is sent back to the frontend.

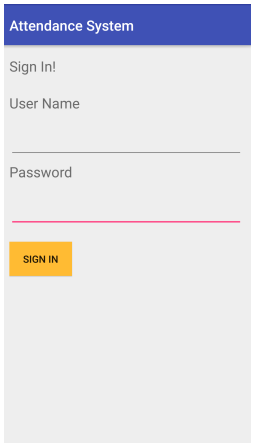

Figure 2: Login screen

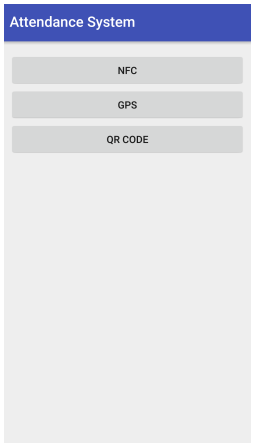

Figure 3: Common UI root to all three solutions

3.1 QR Code

Online Attendance System using QR Code [7] requires two parts, the lecturer's view would allow them to generate a QR Code for each lecture and then the student's view allows each of the students to scan the QR code and register their attendance for the corresponding class. The open-source barcode image processing library, ZXing ("Zebra Crossing")[3] was used to generate and scan QR codes.

The implementation so far allows lecturers to type in any text and generate the corresponding QR Code by clicking on a button. This could have been implemented with multiple inputs for date and time of lecture, course, passcode and so on and use a long concatenated string to generate a QR Code. However, this does not provide any advantage over generating a QR code with just one input. If entered text is empty, a message is displayed that says, "No text entered". Otherwise, the ZXing library is used to encode the input text into QR Code format and stored in a *BitMatrix* with dimensions corresponding to the *ImageView*. Then a *BarcodeEncoder* is used to generate a *Bitmap* from the *BitMatrix* and then is viewed on the screen. The input text is sent to the backend to be stored in the database.

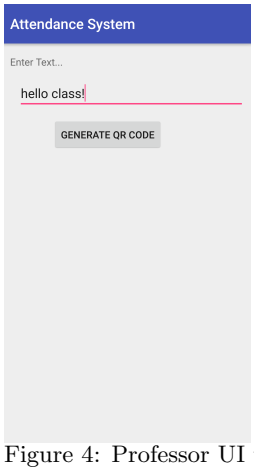

Figure 4: Professor UI to input text to generate QR Code

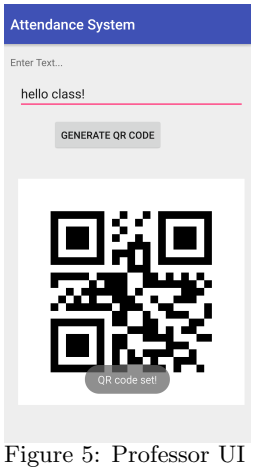

Figure 5: Professor UI showing the generated QR Code

Figure 4 shows the user interface available to the professor to enter text to generate the corresponding QR code. Figure 5 shows the user interface showing the generated QR Code when he/she clicks the "Generate QR Code" button.

From the student's view, students click on the scan button which then opens the camera on the device. The *ZXingScannerView* class allows us to scan QR Code and retrieve the decoded text as a String. This text can be sent to backend to compare and register attendance for the student as required. Figure 6 and 7 showing the UI available to the student to scan the QR code generated by the professor.

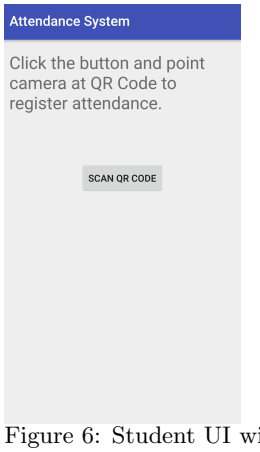

Figure 6: Student UI with instructions

Figure 7: Using Camera to scan QR Code

There are ways we could make it easier for lecturer to transfer the Bitmap which contains the generated QR Code to his lecture slides. For example, saving the image on phone storage upon the click of a button, or automatically uploading the image to a URL which can be accessed through a browser on their computer or upload to a file hosting service such as Dropbox on the click of a button.

We tested our application by generating the QR Code using the professor's login and by reading and decoding the same QR Code using student's login. Students were able to register their attendance when the right QR Code is scanned.

With respect to security, we need to handle two important cases. One being, QR code being sent to a student who is not present in the class by a student who is present in the class to cheat the system. Second, student using two phones and register attendance for a person who is not present in the class by a student who is present in the class. To overcome the first problem, we would like to restrict the attendance registration window to be open for about 15 seconds. Because, it will easily take 15 to 20 seconds to capture a photo and send it to the outside world using some medium such as instant messenger or email. Professor can easily verify students who submit late manually to avoid misuse. Also, when the professor checks randomly one or two students who registered for attendance will make the students less likely to cheat the system. As it will not take much time to check one or two students manually, we believe it would be good solution to this particular security problem. To solve the second problem, we could make use of techniques such as fingerprint or facial recognition to log in to the app for a particular user name.

3.2 NFC

NFC is a wireless communication protocol, like bluetooth and WIFI for data communication. We have made use of the reader/writer mode in our application. In the writer mode, the professor/ teacher will be able to write information or security code onto the NFC tags. In the front end application, we provide the user with a text box, where he/she can write the text, and then with a simple tap of the phone against the NFC tag, the user will be able to write the text in the textbox onto the NFC tag. Figure 8 shows the UI that helps professor to input his code and write it to the NFC tag. When internet or NFC is not enabled in the phone, corresponding error messages are shown to the user.

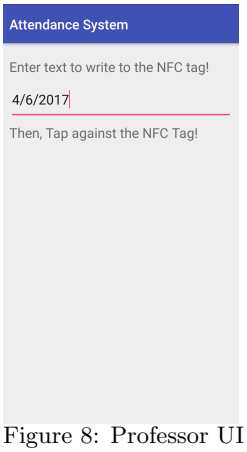

Figure 8: Professor UI to write data into NFC tag

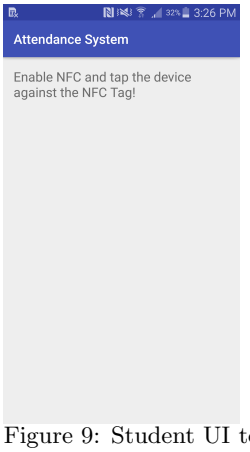

Figure 9: Student UI to read from NFC tag

In the reader mode, the students will be able to read the information from the NFC tag by simply tapping their mobile phone against the NFC tag placed in the classroom by the professor. The registration method can be invoked by the student from the NFC page specific to the student. Figure 9 shows the UI and the instructions on how to use his phone to register for the attendance with the help of NFC.

NfcAdapter, a class in Android provides APIs to communicate with the NFC technology. Intent is a class that defines an action based on a string value. It also contains information about how that particular action need to be executed. *IntentFilter* is a class that is usually used alongside Activities, and services of Android system. The important function of the intent filter is to filter certain intents. For example, when the phone detects an NFC tag, it will fire an intent stating that it has discovered a new NFC tag. When an application uses the intent filter to filter "ACTION_TAG_DISCOVERED", the application will be notified with the intent. Then, the application will decide how to execute it. When there are more than one application that have registered the intent filter, the user will be given a list of applications that can handle the intent, and the user usually selects one of the applications. These intent filters can be registered in the Manifest file or inside the Activity file itself. Activities are the screens that are displayed to the user. When the developers doesn't want other applications

to be notified about a particular intent when his application is running, he may filter those intents inside the Activity rather than in the Manifest file. To enable and disable such a functionality, we have used *enableForeGroundDispatch* and *disableForeGroundDispatch* functions. In the reader mode, "ACTION_NDEF_DISCOVERED" filter is used. And, in the writer mode "ACTION_TAG_DISCOVERED" filter is used. NDEF is the message format used in android along with NFC.

When the intent is filtered and dispatched to the application, *onNewIntent* callback function is invoked. Write and read actions can be performed in the *onNewIntent* function. When a write operation is performed, the text that is written to NFC tag is also sent to the backend for the verification of codes that will be submitted by students for attendance. In reader mode, the code recovered from Tag is sent to the backend for verification. After verifying the code, students will be notified whether they registered their attendance successfully or not.

We tested the application, by writing a message "Hello Class!" to NFC tag using professor login and by reading it using student's login. When the correct tag is read, the students were able to register their attendance and when they read the wrong tag, they were not able to register their attendance. To improve the security, MAC address and Location details of the phone can be collected along with the NFC code.

There are two scenarios where this approach could be misused. One being, friend of a student can bring another phone to register for his/her friend. It can be avoided by using fingerprint or facial recognition to log into the app. Second being, a student can read the data on the NFC tag and send the data to his/her friend. Then, the student can write the same data into his/her own NFC tag and register for the attendance. To overcome the problem, we can make use of his/her GPS location. Or, we can make use of a raspberry pi in the classroom instead of NFC tags. Raspberry pi with NFC capability can run on reader mode and can collect information such as MAC address of student's phone or fingerprint data of the student and submit the attendance to the backend from raspberry pi. As the IP address and MAC address of the raspberry pi cannot be duplicated by student, they cannot cheat the system.

3.3 GPS

Similar to NFC and QR Code solutions, front end for GPS solution is implemented in Android. The idea is to make use of the location and proximity of the student to the professor in determining the student's presence in the classroom.[6] Professor and students send their GPS coordinates to the backend, where the proximity of the student to the professor is calculated and the attendance is provided if he is available close to the professor. Figure 10 and 11 shows the UI available to professor and student respectively.

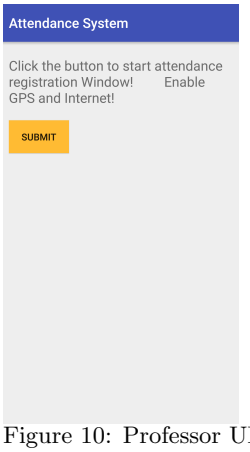

Figure 10: Professor UI to set GPS coordinates

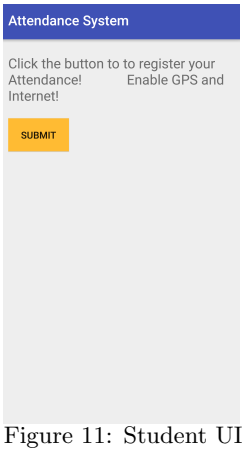

Figure 11: Student UI to submit GPS coordinates

Android provides a class called *LocationManager*, which provides system location services. It also provides a listener called *LocationListener*, which provides callback functions when there is a change with respect to location or the provider, such as GPS and Network, which provides location services. Latitude and longitude information is retrieved using these available functionalities.

We tested the functionality of the system using two phones. One phone logged in using professor's credentials, and another with student's credentials. When both of the mobile phones are present inside the same room, the student was able to register his presence. We also tested our application with student's and professor's phones set apart at a distance of approximately 90 to 100 meters, and the student was not able to register his attendance under this scenario. No internet connectivity and GPS are other scenarios that we tested and we show appropriate messages/notifications to alert the users.

There are a couple of scenarios where GPS method could be misused. One being, a student can carry more than one device to register for a person who didn't attend the class. It could be handled using methods such as fingerprint scanner or facial recognition. Second being, student can just submit his attendance from just outside the class and leave. This could be stopped by a combination of a couple of strategies. The professor can ask the students to register for the attendance at a random given time frame during the class and randomly checking for one or two students presence manually.

3.4 Database

We have four tables in our database:

1. **Student**: To store the information of the students: (id, name, password, courses)
2. **Professor**: To store the information of the professor (id, name, password)
3. **Course**: To store the information of the course (id, name, professor, latitude, longitude, NFC, QR-Code).The latitude and longitude fields store the corresponding

professor's GPS co-ordinates. NFC and QR-Code fields store the professor's text used for generating NFC and QR-Code respectively.

4. **Attendance**: To store the attendance information (course, student, GPS, NFC, QR-Code). GPS, NFC, QR-Code are all time fields, used for saving the date and time when the corresponding method was used for attendance validation.

3.5 Frontend and Backend Communication

Communication between frontend and backend is performed using rest APIs. Server accepts http post messages with JSON payload. On successful processing of the message, server sends back appropriate success and failure notifications back to the client. *HttpsUrlConnection* class is being used by the client to create connection and communicate with the server. Server side the incoming JSON object parsed and converted into a dictionary and used according to dictionary keys available.

4. USER EVALUATION

We evaluated our software with the help of 23 students manually experimenting and testing our client side android application.[5] As our major focus of this project is to reduce the time overhead introduced by the traditional attendance management system, we decided to capture the time taken by each proposed method. We also asked them to think out loud about the advantages, disadvantages and suggestions of each method. We also wanted to know how much they liked the application and will they use our application from both professor and student's perspective. All the details were collected using survey and think aloud method.

We anticipated that most of the students will not have any problem using the application except NFC. So, we decided to repeat the NFC experiment two times. First time, we asked them to experiment the NFC method providing them oral instructions on how to use it. And second time, we asked them use it on their own. When a solution like NFC is available to the students, they may struggle initially, but eventually they will use it with much ease and fast.

5. RESULTS

a. Easiness to use the application

Out of all three solutions, GPS and QR Code are straightforward approaches and they don't need additional assistance. But, user needs little bit of practice to operate NFC as it involves certain physical operation which are not familiar to many people. Thus, we decided to experiment with NFC on how easy it is to use after learning how to use it.

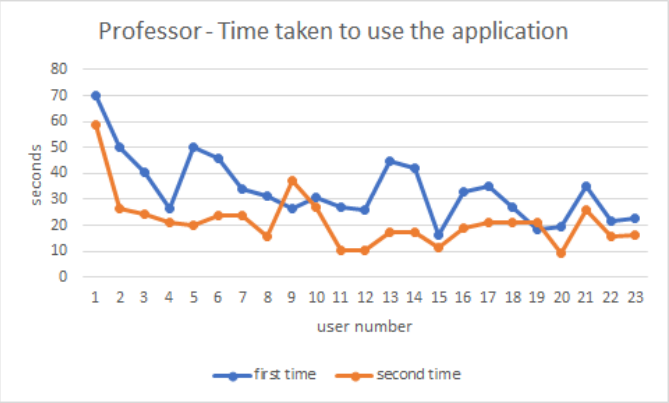

Figure 12: NFC - Time taken by the professor first and second time

We anticipated that the people evaluating the NFC part of the application will feel little difficult to use it as NFC is not widely popular and it involves physically bringing the mobile phone to the nfc tag in a very close proximity. So we decided to repeat the NFC experiment twice. As anticipated, people struggles to use the application first time. Figure 12 clearly shows that first time users took more time to perform the function compared to the second time. First time, the average amount of time taken by the professor is 33.6 seconds. Second time, the same users only took 21.5 seconds. The main reason to repeat the experiment is to prove that with experience it is very easy to use the technology. From the data, we observed that some of the users took only 10 to 15 seconds to complete the all the operations of logging in and writing the data. And, this again proves our point that with experience, that technology could be use much easier.

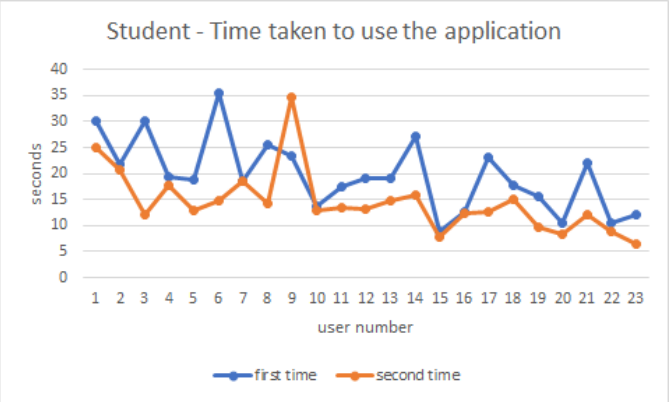

Figure 13: NFC - Time taken by student first and second time

Figure 13 shows the amount of time taken by student first and second time using the NFC to register for attendance. First time, students took 19.6 seconds on average to read data from the tag which also involves logging into the application. Second time, it took 14.5 seconds to perform the same operation. This, again proves that with experience NFC is not a very difficult method to use.

b. Time taken by the professor to operate all three solutions

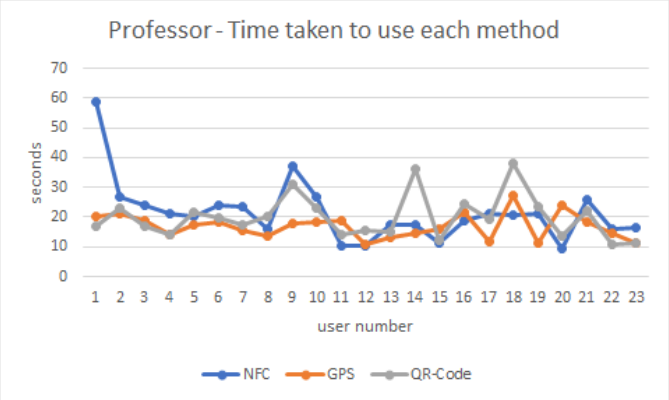

Figure 14: Time taken by professor to use NFC, GPS, QR Code

NFC - From the data collected from the 23 users, it takes 21.5 seconds on an average to log into the application and write data into the tag by the professor. But, if the application is left logged in, it would take around 6 to 7 seconds to complete the operation by a person who knows how to use the application. The NFC write operation involves writing some text into a textbox and then tap against the NFC tag. Let's consider a scenario that for a class of 50 students, 5 tags are placed at the entrance of each seating row. The text professor is going to write into the textbox is only once and thereafter he just need to tap against all the tags one by one. It would approximately take 3 seconds to write data into the tag. Considering moving from one tag to another to write the data, the instructor could spend another 15 seconds in total. It will approximately take 30 to 35 seconds to tag the whole classroom. As the professor could do it even before the class starts, it won't really affect the lecture time.

GPS - From the collected data from 23 participants, we found out that on average it takes 16.9 seconds to perform the GPS operation by the professor. It involves logging into the application and perform the gps operation. The GPS operation includes click of two buttons. First button to enter the gps activity from menu activity and second button to get the current coordinates and sending to the backend. Backend on storing the coordinates, will send back a response. Considering logging in requires approximately 10 seconds, the whole GPS operations can be performed within 7 seconds.

QR Code - From the collected data from 23 participants, we found out that on average it takes 20.0 seconds to perform the QR code generation by the professor. It involves logging into the application and perform the QR Code generation. Considering logging into the application takes 10 seconds, if the application is left logged in, the same operation can be performed in 10 seconds. The time difference between GPS and QR code is due to the fact that GPS doesn't need any text input whereas QR Code method does.

c. Time taken by the student to register attendance using all three solutions

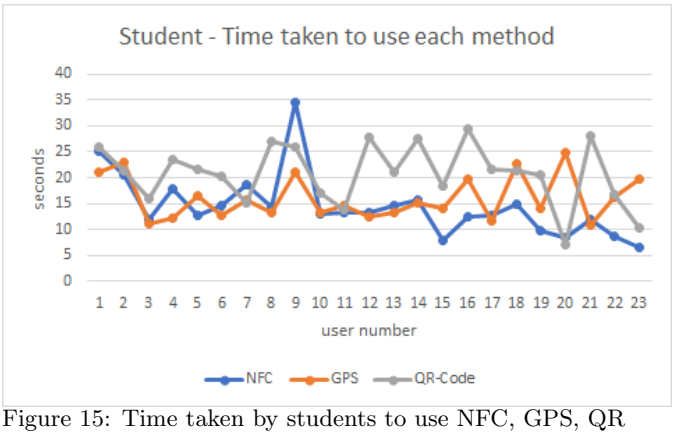

Figure 15: Time taken by students to use NFC, GPS, QR Code

NFC - From the data collected from the 23 users, it takes 14.5 seconds on an average to log into the application and read data from the tag by the students. But, if the application is left logged in, it would take around 4 to 5 seconds to complete the operation by a person who knows how to use the application. The NFC read operation involves just tapping the phone against the tag. The time difference of 2 seconds between write and read operation is due to the fact the professor spend time to write some data into the text box before writing the data to the tag in write mode, which is not the case in the read mode. Let's consider a scenario that for a class of 50 students, 5 tags are placed at the entrance of each seating row. In the worst case scenario, if 10 people are waiting in each line to enter the row containing seats, it would approximately take 40 to 60 seconds to check the presence of all the students.

GPS - From the collected data from 23 participants, we found out that on average it takes 16.0 seconds to perform the GPS operation by the student. It involves logging into the application and perform the gps operation. The GPS operation includes click of two buttons. First button to enter the gps activity from menu activity and second button to get the current coordinates and sending to the backend. Backend on verifying the coordinates, will send back a response. Considering logging in requires approximately 10 seconds, the whole GPS operations can be performed within 6 to 7 seconds.

QR Code - From the collected data from 23 participants, we found out that on average it takes 20.7 seconds to perform the QR code scanning and attendance registration. It involves logging into the application, perform the QR scanning, and verifying the code with backend. Considering logging into the application takes 10 seconds, if the application is left logged in, the same operation can be performed in 10 to 11 seconds.

d. Usefulness of the application

In order to gauge the usefulness of the application we asked each of the users to rate on a scale of 1 to 5 how likely they

were to use each of the three solutions from the professor and student perspective.

NFC - The ratings given to both the professor and student perspective were very similar. Both perspectives got an average rating of 4.22 with 10 out of the 23 users rating it 5 out of 5. The favorable ratings indicate that users felt this was a good solution from both professor and student perspective.

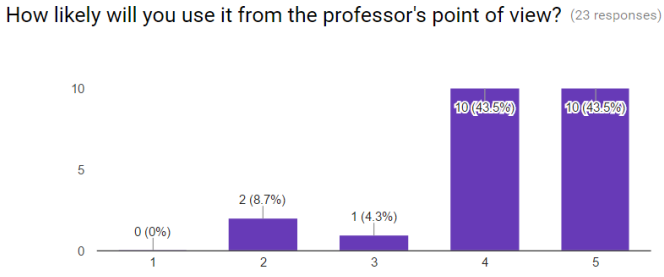

Figure 16: Bar graph for NFC from Professor's point of view

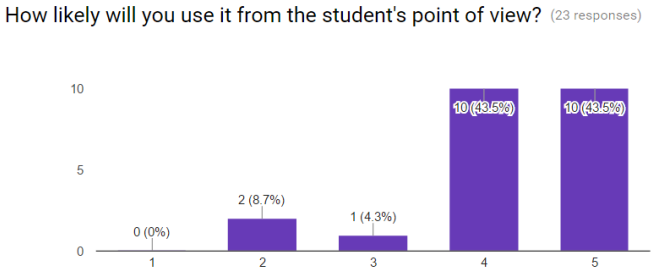

Figure 17: Bar graph for NFC from Student's point of view

GPS - The professor perspective got an average rating of 3.96 with only 6 users rating it 5 out of 5. While still favorably rated by most users, this is the lowest rating for any approach. The students perspective was rated considerably better with an average of 4.48 including 13 users rating it 5 out of 5. This is the best rating received for any approach.

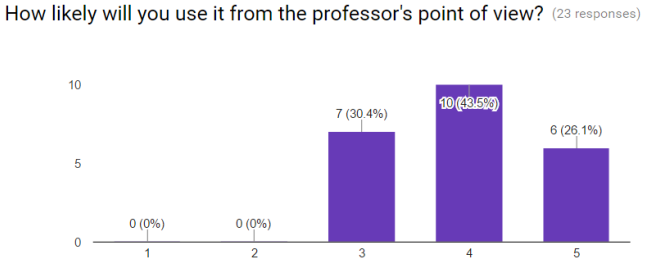

Figure 18: Bar graph for GPS from Professor's point of view

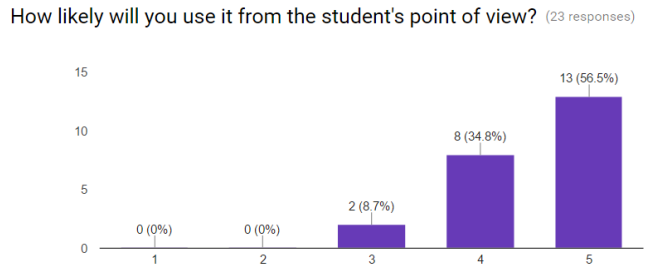

Figure 19: Bar graph for GPS from Student's point of view

QR Code - The professor perspective got an average rating of 4.04 with 10 users rating it 5 out of 5. Hence, most users felt this was a good solution from the professor's perspective even if it was not the best. The students perspective was rated slightly better with an average of 4.26 including 12 users rating it 5 out of 5. This indicates that users felt this was a very good solution from the student's perspective.

Overall, all three approaches were favorably rated which means that they were all good solutions for an automatic attendance system. From the professor's point of view, NFC was most likely to be used and GPS least likely. While from a student's point of view GPS was most likely to be used and NFC was least likely.

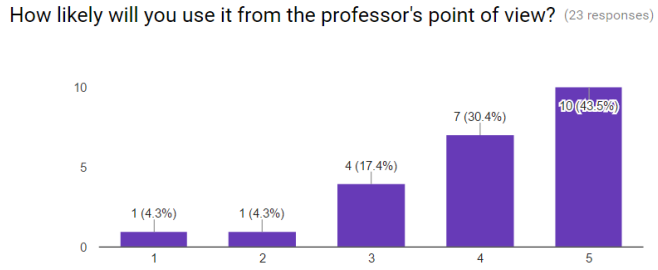

Figure 20: Bar graph for QR Code from Professor's point of view

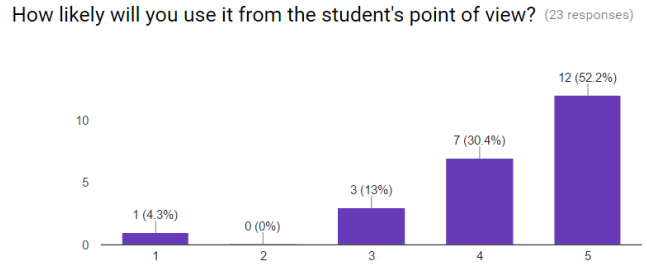

Figure 21: Bar graph for QR Code from Student's point of view

e. Comparison between NFC, GPS and QR Code

Positive and Negative responses of users were collected for each methods from 23 users. All three methods had positive review when compared to traditional methods of taking

attendance. The most common positive review for all three methods (received this from almost all users) was it is very simple to use and very fast. This was resulted due to a very intuitive and simple design of the application. The most common negative review for all three methods was they can be handed over to another user (a friend) and might result in wrongfully giving attendance. This can be prevented by biometric validation of the user. This can be done by fingerprint scanning and facial recognition methods. The gist of reviews for each method is given below:

NFC
Major advantage of this method is that it requires the physical presence of the device to be very close to the NFC chip as the NFC range is very low. This makes it the very reliable and secure.

The disadvantages of this approach is that many phones might not have the necessary hardware to use this method.[4] It requires additional hardware and a learning curve to use it. Also, fewer number of NFC reader installations might lead into long queues.

GPS
This solution is the easiest to use. No additional hardware is required. Both users are only required to click a button.

Disadvantages of this solution are that it is not accurate nor precise. It might not be able to identify if the user is inside the classroom or just near the classroom. It doesn't consider the altitude, so students can register attendance by being in a room directly above or below.

QRC
This solution is also easy to use. Major advantage is it is very reliable since it requires the physical presence of the device to be very close to the screen (where the QR Code is displayed). Also, no additional hardware required.

Disadvantages are that it can be easily cheated by Students taking picture of the QR Code and sending it to other users. Might take time to scan the QR Code image if the screen on which it is displayed is not clear. It requires a display screen in the classroom.

NFC vs GPS vs QR Code

1. Comparison in terms of easiness to use
    - QR-Code is easy for the professor as he has to only enter some text and click a button, but for students it might take a while if the image cannot be scanned.
    - GPS is the most easiest to use, as it requires only 1 click from both professor and students' perspective.
    - NFC might take time for both professor and student to scan.
2. Comparison in terms of cost & additional hardware
    - NFC required external hardware i.e., NFC chip to read and write. NFC chips are inexpensive, but high quality ones might be expensive.
    - GPS doesn't require any external hardware and hence no additional cost.
    - QR-Code requires a screen to display the image on, but all the classrooms are equipped with display screens. Hence, there is no expenditure.
3. Comparison in terms of foolproof/security
   Making the entire application more secure results in reduced misuse of all 3 methods.
4. Comparison in terms of reliability
    - NFC is the most reliable of all the methods as it requires the device to be physically very close to the NFC reader.
    - GPS is not reliable as it is not accurate or precise.
    - QR-code is reliable, but students can take pictures of the qr-code image and send it to other users. But this can be overcome easily.

6. CONCLUSION
Based on the survey that we conducted at the early stage of this project, we found out that approximately 73 percent of the instructors in the educational institutes take attendance seriously and approximately 7 minutes of the lecture time is wasted by the instructors to perform manual attendance maintenance for a class of size 50. As we found out that mobile phones are widely used by the students, we decided to propose 3 solutions to the problem using three different technologies available on the mobile phone namely NFC, GPS and QR Code.

With NFC method, the instructor can save approximately 6 minutes of lecture time. With GPS and QR methods the instructor can save approximately around 6 minutes and 40 seconds.

Each method has its own advantages and disadvantages. In the earlier sections, we saw that GPS and QR Code take very less time compared to the NFC method. But, as discussed, both GPS and QR Code will need little bit of human intervention to make the methods foolproof. But, even if NFC takes more time, with the proposed Raspberry pi model in the place of NFC tags, the NFC method can be made foolproof and doesn't require any human intervention.

7. FUTURE WORK
In all three solutions, the common problem that we discovered was that the system could be cheated with little effort. As discussed in the implementation section, the system could be made more reliable and secure with the help of additional technologies. Fingerprint scanner or facial recognition can be used to login to the application instead of manual login using username and password. This will avoid one student helping another student to register attendance illegally. And, with respect to NFC solution, as discussed in the implementation section, the method can be made reliable and secure by replacing the NFC tag by Raspberry pi with NFC capability connected to the internet.

8. REFERENCES

[1] Django Framework. https://www.djangoproject.com/.

[2] GitHub. https://github.com/SidHeg/se17-teamD.

[3] ZXing Library. https://github.com/zxing/zxing.

[4] NFC phones: The definitive list. https://www.nfcworld.com/nfc-phones-list/, 2017.

[5] User Evaluation. https://docs.google.com/a/ncsu.edu/forms/d/13oLs4_mkbnDnkZop3YC0Gw9hvlc2Ipf7m8XGvF929Cg/edit?ts=58dd6937, 2017.

[6] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low-cost outdoor localization for very small devices. *IEEE personal communications*, 7(5):28–34, 2000.

[7] F. Masalha and N. Hirzallah. A students attendance system using qr code. *International Journal of Advanced Computer Science and Applications*, 5(3):75–79, 2014.

[8] M. Sardar, S. Balasubramanian, and S. Heggadahalli. Automatic attendance system : Initial project description.

[9] P. Sutheebanjard and W. Premchaiswadi. Qr-code generator. In *Knowledge Engineering, 2010 8th International Conference on ICT and*, pages 89–92. IEEE, 2010.

[10] Wikipedia. Global positioning system.

[11] Wikipedia. Model-view-controller.

[12] Wikipedia. Near field communication.

[13] Wikipedia. Qr code.

Chit Numbers:

1. ddkjoaae
2. dmjseeoi
3. fkmmoiuo
4. dgklaoui
5. dbbxeaoi
6. ffbroiie
7. cjlfeoio
8. dkkmueeu
9. dbkfouao
10. chkmeaae
11. fhkyieuu
12. ckkwaoeu
13. fgkgoaiu
14. gbgfoioi
15. dfkvuuaa
16. ddbzeiai
17. fblkoouo
18. cllreoie
19. ffkcuaoo
20. cdhrueoa
21. fmmnaeau
22. fllgaoou
23. gchpueio