

Purpose

This report includes the research on feasibilities of some functionalities. We are new to Android so we want to do some research to see how feasible it is to implement our desired functionalities.

Audio recording

After researching, it seems like audio recording is a fairly standard and simple feature in Android that can be implemented with ease.

<https://developer.android.com/guide/topics/media/mediarecorder>

Android offers detailed documentation on the implementation of audio recording.

MediaRecorder is used for recording.

MediaPlayer is used for playing.

A sample program for MediaRecorder is provided by Android.

While it definitely requires some more study into the tutorials and trials with coding, I think there is nothing unrealistic about implementing an audio recording feature.

<https://medium.com/mindorks/developing-an-audio-recorder-app-2fc972df62f0>

This tutorial provides steps and code for building a very basic and simple app for recording audio. It has 3 buttons, start, stop, play. This tutorial shows that it's simple to record audio and save it in some file path we want. It also shows that it's simple to use MediaPlayer and the file path to play an audio file.

<https://guides.codepath.com/android/Audio-Playback-and-Recording>

http://www.tutorialspoint.com/android/android_audio_capture.htm

Check-in

I thought of multiple ways to implement check-in and I will discuss their feasibilities one by one here. As a background, check-in is about letting meeting attendants check-in to meetings. The meeting organizer should be able to enable/disable check-in. Attendants check-in only after organizer enabled check-in. Ideally, there should be a way to check if the attendant is actually present in the meeting when they check-in, and check-in should be successful only if the attendant is present.

Bluetooth

Based on research, this would require a Bluetooth beacon, which is a specialized hardware for transmitting Bluetooth signals. To make this work, the organizer's smartphone needs to be turned into a Bluetooth beacon. This I think will be hard to do.

An easier way to do this is to simply form pairing of bluetooth devices between the organizer and the attendant. The organizer's device should create an open socket, accepting connections. The attendant's device will connect to the open socket. After the two devices are connected, the organizer's device can send some information to the other device and make the check-in

successful. However, the pairing is between 2 devices, so there will be multiple connections when multiple people want to check-in. That might cause some challenges regarding speeds, cpu usage, etc. due to the number of connections.

<https://developer.android.com/guide/topics/connectivity/bluetooth#ConnectAsAClient>

NFC

Android has Android Beam which allows an android NFC device to serve as the NFC tag, when two device touch, information can be exchanged. So, the organizer's device can serve as the NFC tag and attendants tap their devices with the organizer device. The drawbacks is that the organizer and attendants' phones need to be tapped together.

<https://developer.android.com/guide/topics/connectivity/nfc/nfc#p2p>

GPS

One way to achieve this is with GPS. We can obtain the users' location by using GPS when they are using the app. From my research, seems like GPS are fairly accurate, the error is below 10m. So, when organizer enable check-in, a GPS coordinate can be uploaded to the database, and when attendants check-in, their coordinates will be obtained and compared to the one in the DB, if they are in a certain range, the user check-in successfully. This only works if we assume that people will attend the meetings if they are a few meters away. If users stand in somewhere close to the meeting room, but don't actually attend the meeting, and check-in, the app probably can't filter them. However, considering the target audience, this case will be unlikely.

<https://stackoverflow.com/questions/42361598/android-gps-location-accuracy-issue>

Geofencing seems to have some potential for this application, but it is more complex than just getting the GPS coordinates.

QR code

The organizer can generate a QR code, aka barcode, when he/she enables the check-in, then the QR code can be projected to a screen and attendants have to scan this QR code to check-in. The barcode probably has some sort of message encoded, such as the ID for this meeting, when an attendant scans the barcode and decodes the message, the app send a message to the database that this user checked-in.

This isn't hard to implement because there are various libraries for generating barcodes, such as Zxing library:

<https://medium.com/@aanandshekharroy/generate-barcode-in-android-app-using-zxing-64c076a5d83a>

Similarly, scanning and decoding QR code is a fairly common feature so there are many tutorials for that.

However, this method doesn't provide the best user experience. The organizer has to create a QR code and project it, the users have to scan the QR code. The projection part might be perceived as cumbersome. Still, this is a possible solution.

Manually

This is not a very desirable way to implement this, but could be used as a last resort if other methods have difficulties. Let the meeting organizer view a list of attendants and click them (or a slider button) to check them in. Not optimal because the organizer has to do more work.