**Demo Video:**

<https://drive.google.com/open?id=1_QVQiajylb1gwbyZ9rRlSiZL5KAtOGzs>(2020.03.05 from Unity)

<https://drive.google.com/open?id=1xVpCAXwzabO1rrKbp1qdqMXfCXG3U_ye>(2020.04.05 from mobile)

**Android Apk:**

<https://github.com/Ludolab/AudioGWAP2020/blob/master/prototypes/Mobile%20App%20New/AudioGWAP20200409_fixed.apk>

**Flow in the cross-validation part:**

1. **Data**

Dynamic downloading audio files:

* When the user gets into the home page, download 10 audio files in the cache, waiting for the cross-validation.
* When the user clicks "save"(successfully validate one file), remove this audio file and download another one from the server.

1. **User Experience**

* The user can re-listen to the sounds, skip validating current sound, save their choice, and report the problematic sound.
* When the user doesn't choose any option, he cannot press the save button to continue. Once there is one option pressed, the save button will be activated.
* The user can change their choices before clicking the save button. Once the user clicks “save”, do the compare function: whether the chosen option matches the label in the database.

**Things need to be done in the cross-validation part:**

1. Connect with the new server--CMU Cloud

(using this url for test now <https://echoes.etc.cmu.edu/api/game/events/fake/sound>, should be charged to the url that saved all the sounds for validation)

1. Communicate the data to the server

(log in the unity for now, should be sent to the server when the new server is set up)

* Whether the user’s choice is match to the tag that saved with the sound
* When the data is collected and when the data is uploaded (timestamp when the user hits the record button & the submit button)
* Define the type of problematic sounds(when the user chooses to report this sound)

1. Other data that may serve for the cross-validation (TBD with the server side)