



Go Listen: An End-to-End Online Listening Test Platform

SOFTWARE METAPAPER

DAN BARRY 

QIJIAN ZHANG 

PHEOBE WENYI SUN 

ANDREW HINES 

**Author affiliations can be found in the back matter of this article*

][ubiquity press

ABSTRACT

Subjective listening tests are routinely conducted by academic researchers and industry professionals to assess the quality of various speech and audio processing algorithms and transmission services. Listening tests often take place in controlled environments for the sake of consistency, but in many cases, listening tests could be undertaken remotely using a suitable web interface. Despite the work of several projects in the past, there is no publicly available, fully hosted listening test platform which allows for easy test creation, deployment and data collection. Here, we present a fully functional end-to-end listening test platform which allows a user to create and share MUSHRA, ACR and A/B tests within minutes. Collected data can then be downloaded in various forms. For users who would prefer to host the system on their own servers, we provide Docker and AWS images for easy installation.

CORRESPONDING AUTHOR:

Dan Barry

Developer – School of
Computer Science, University
College Dublin, Ireland

danielnbarry@gmail.com

KEYWORDS:

Subjective; Listening; Quality;
Tests; Perceptual; Evaluation;
AB; MUSHRA; ACR; P.800

TO CITE THIS ARTICLE:

Barry D, Zhang Q, Sun PW,
Hines A 2021 Go Listen: An
End-to-End Online Listening
Test Platform. *Journal of Open
Research Software*, 9: 20.

DOI: <https://doi.org/10.5334/jors.361>

(1) OVERVIEW

INTRODUCTION

Advances in audio processing technology often employ a series of objective and subjective tests to compare performance against previous algorithms. In the case of subjective listening tests, the most commonly used methodologies include; *Multiple Stimuli with Hidden Reference and Anchor* (MUSHRA) [1], *Absolute Category Rating* (ACR) [2] and *A/B comparison testing* [2]. Increasingly, these tests are being conducted online using browser based technology and it has been shown that the results for online tests correlate well with those obtained in controlled laboratory conditions [3]. Typically, a researcher will design a bespoke testing system or extend from an existing framework. The most common frameworks are briefly described here.

The common open-source frameworks developed for research purposes are *BeagleJS* [4], *Web Audio Evaluation Tool* (WAET) [5], *webMUSHRA* [6], *TheFrageboden* [7], and the *P.808 Toolkit* [8]. These JavaScript-based frameworks allow for online listening tests without proprietary access to third party software such as MATLAB [9] and MAX [10]. The advancement in the Web Audio API also made synchronous and flexible playback possible in the browser, which contributes to the versatility of the tests forms available. Most frameworks have incorporated the standard subjective assessment procedures including *ITU-R.BS.1284* (for general audio), *ITU-R.BS.1116* (for audio with small impairments), and *ITU-R.BS.1534* (for audio with intermediate impairments) to ensure the tests are standard complied. The forms of the standard test procedures can also be used for customised research scenarios.

The frameworks differ in their supports for customisation to suit various experiment requirements. *BeagleJS* mainly supports A/B and MUSHRA tests or any modified tests based on the two predefined tests forms; the *webMUSHRA* and *WAET* allow researchers to build their tests based on a wider range of tests forms and rating scales; *TheFragebogen* has extra support for behaviour research by incorporating behavioural scales (the NASA task load index (TLX) and the visual analogue scale (VAS)), allowing for free-hand input, and documenting user's response time; the *webMUSHRA* provides 2D and 3D GUIs to facilitate the report of localisation judgement for spatial audio assessment. In response to the challenge of the environment control in remote tests, different frameworks also provide solutions accordingly. *TheFragebogen's* response time records are used to scrutinize anomalous user behaviours; the *P.808 Toolkit* implements the standard crowdsourcing methods [11] which test participant's hearing ability, headphone usage, and the environment to control for the test conditions remotely. The *P.808 Toolkit* is a welcome addition but currently only runs on the Amazon Mechanical Turk (AMT), a common platform for

crowdsourcing experiments, to integrate the experiment design, setup, and recruitment procedures in one place. These frameworks are admirable but they require coding knowledge and a significant time investment to successfully configure and connect them to a suitable database layer along with deployment on a secure publicly accessible server.

Our goal in creating this system was to allow any researcher to create and share listening tests easily without the need for coding, managing deployment or hosting. To that end, we have created *GoListen* which is a fully hosted end-to-end listening test platform with support for several popular listening test types along with a variety of survey question types. We are making the source code available but the key advantage of our system is that we are also providing a fully hosted web application. In total, we provide three options for using the system.

- **Web Application** – Designed for users with no coding skills or those who want the convenience of a fully hosted system. <https://golisten.ucd.ie/>.
- **Docker and AWS images** – Designed for users with moderate technical skills who would prefer a self hosted option.
- **Source code** – Designed for those with expert coding skills who wish to extend or customise the system for specialised use.

SOFTWARE FEATURES

The system we present here, allows users to create and share a test in minutes with no code or setup required. Currently supported test types include A/B, ACR and MUSHRA with more test types to follow. Survey questions can be inserted before, after and between audio examples. Question types include multiple choice, check box and free input text answers. Audio playback for each question can be set to loop infinitely or a fixed number of times. The subject can also be required to listen to each audio example in full any number of times before being allowed to proceed. Switching between audio examples is fully synchronised, meaning that audio playback continues uninterrupted regardless of which audio condition is being listened to. Subjects may also loop sections of audio if enabled by the test creator. Individual questions can be optional or required for both audio examples and survey questions. Questions and audio examples can be viewed on individual pages or the full test can be configured as a single page view. Audio test order can be randomised for all test types and stimulus presentation order can be randomised within a MUSHRA test. The system also allows for timing each question. In terms of the test builder, questions can be reordered and edited easily. Tests can be created from a blank template or a predefined template. They can also be edited and

duplicated. When the results have been collected, they can be downloaded in CSV and JSON format.

In the following sections, we present the user interface and illustrate how it can be used to create tests.

Dashboard View

The user interface is organised in a familiar manner whereby the main navigation is presented in the left pane and the content is presented in the right pane. When a user logs in, they are presented with the dashboard. This is the “home” view and most actions start from here.

Creating Tests

Creating a new test is achieved by tapping the *Add Test* button, [Figure 1](#). The user has the option of selecting a blank test or a pre-populated template which is created by administrators of the system. Upon selection, the test design screen opens which is discussed in a later section.

Copying Tests

Another way to create a test is to copy an existing test which you may have created in the past. Tap on the *Copy* icon and a new test will appear in your dashboard with the same name as your copied test and “copy” appended to the end. The new test can be renamed and will not contain the responses from the parent test.

Edit a test

Any existing test can be edited from the dashboard by tapping on the *Edit* icon. This brings the user back into the test design screen which will be discussed below. Note, if a test has already been shared and received responses, it cannot be edited without making a new copy.

Sharing Tests

A test can be shared by tapping on the *share* icon. This presents a modal allowing the user to copy the public

link for the test or to launch the test in a new browser window. The test can be shared from the dashboard or from the test design screen.

Designing Tests

Designing a test is easy using the set of editing tools the system provides. Every test is given a name and description so that the user can locate the test on the dashboard later on, [Figure 2](#).

Add Survey Questions

Survey questions can be added by tapping the *Add Question* button which reveals a context menu listing various question types, [Figure 2](#). Options include:

- **Checkbox Group** – Allows for multiple options to be selected simultaneously. [Figure 3](#) shows the question design interface.
- **Radio Group** – Allows for a single option from a list of options to be selected. [Figure 4](#) shows the question design interface.
- **Text Input** – Allows answers to be entered as free text. [Figure 5](#) shows the question design interface.
- **Text Label** – Insert a large text label to introduce a new section in a test.

Adding Audio Questions

Audio questions can be added by tapping the *Add Question* button which reveals a context menu listing various question types, [Figure 2](#). Audio Options include:

- **Audio Test** – Allows for an AB, ACR or MUSHRA test question to be added. Each audio test type is discussed in more detail in a later section. [Figure 6](#) shows the question design interface for an A/B test. Audio can be uploaded by clicking on or dragging to the dropzone in the interface. The preference

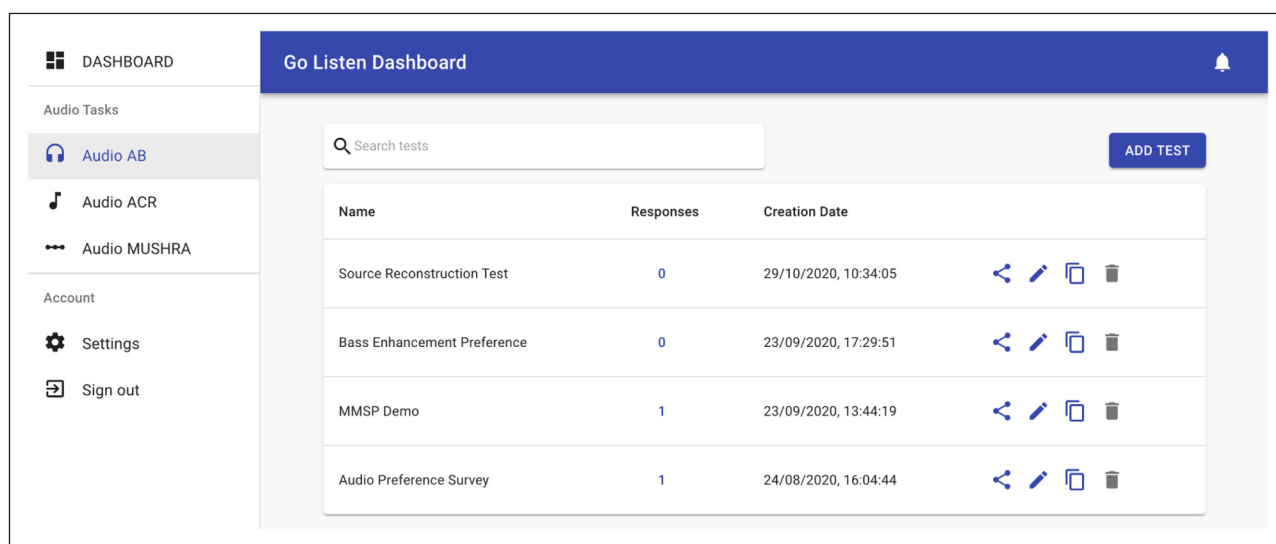


Figure 1 Dashboard screen, where tests can be added, copied, edited and shared.

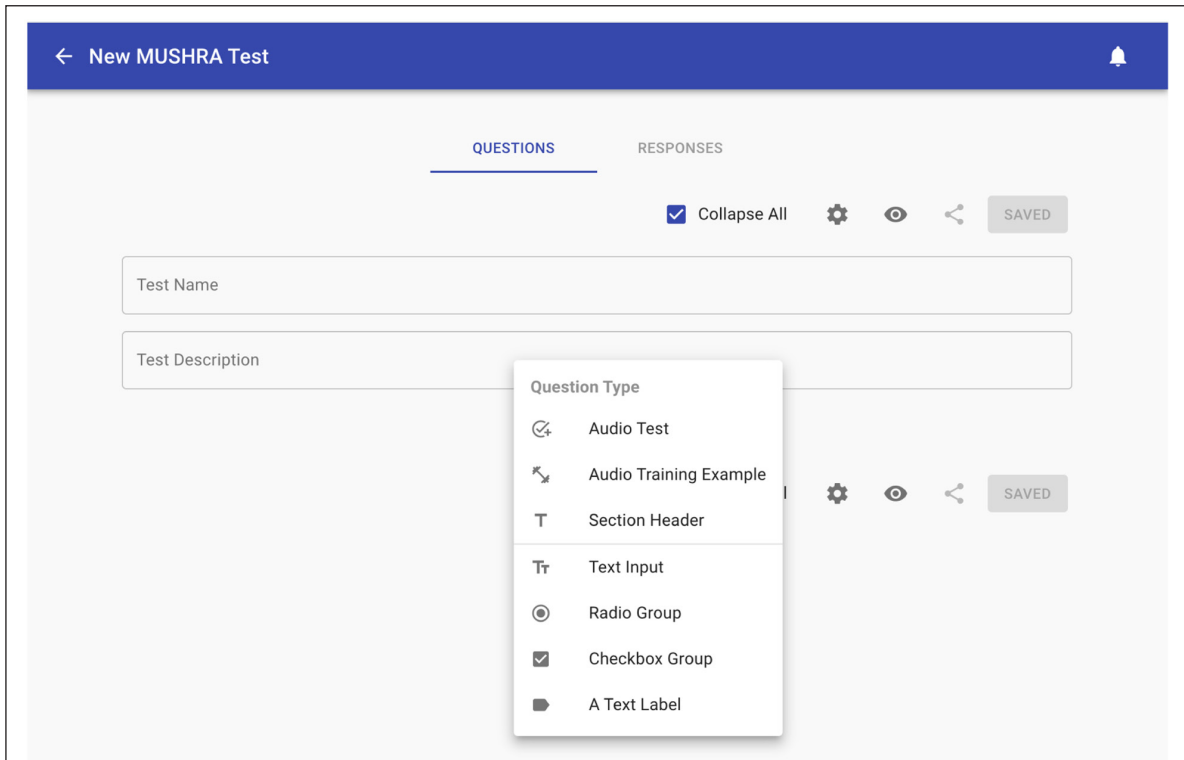


Figure 2 Adding Survey Questions.

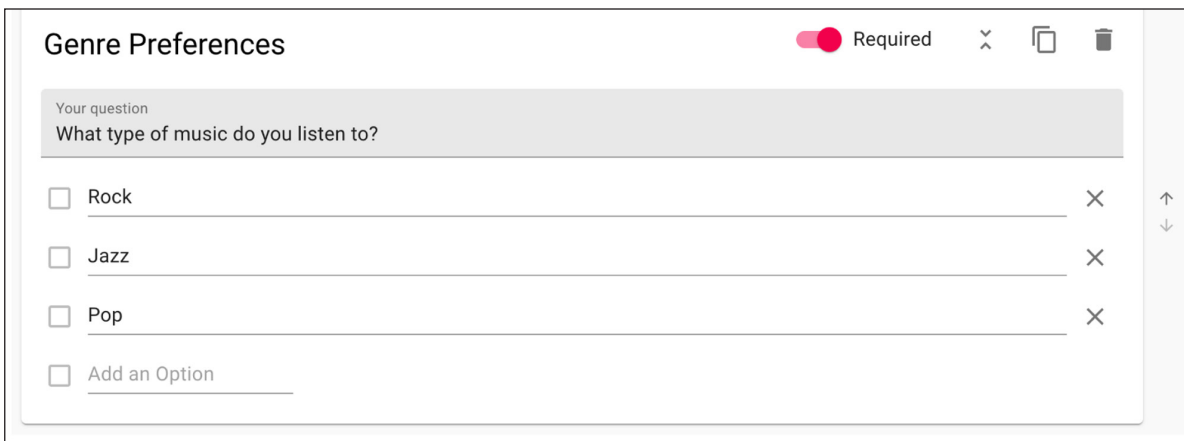


Figure 3 Multiple Selection Question.

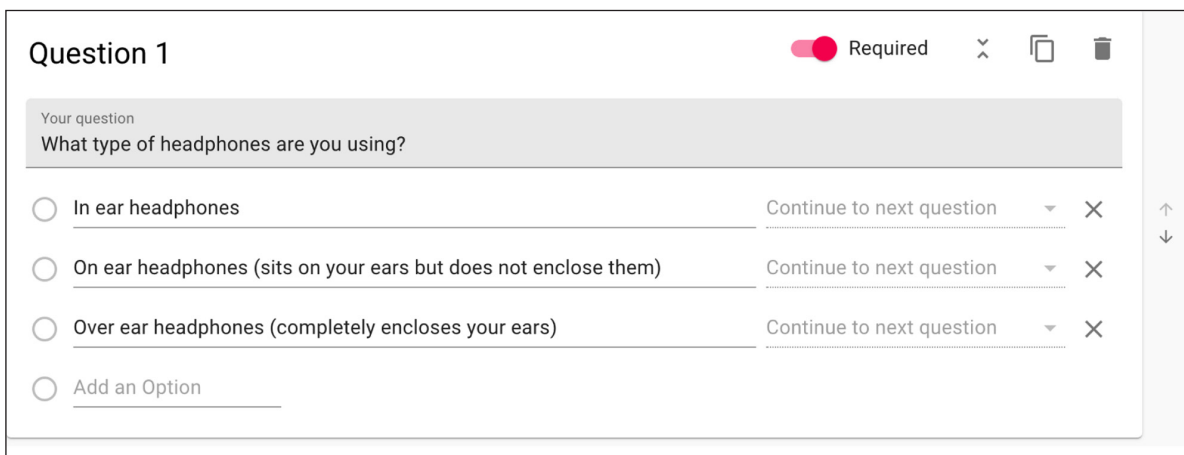


Figure 4 Singular Selection Question.

Figure 5 Text Input Question.

Figure 6 AB Audio Question.

question text can be edited as can the options presented to the respondent.

- **Audio Training Example** – An audio training example is a training step for the respondent. The test designer can provide instructions and an audio example without asking a question related to the example.

Audio Playback Settings

Each audio question has a settings menu which controls how the respondent will interact with audio playback as seen in [Figure 7](#). The audio formats supported are those described in the HTML5 Audio specification [12]. The options are as follows:

- Set the number of times audio will loop (range 1 to infinity)
- Require the respondent to listen to the audio in full
- Disallow the respondent to skip back and forth in time

- Allow the respondent to use looping controls

Previewing a Test

As you build the test in the test design screen, you may want to check what it will look like for the respondent. This can be achieved by tapping on the *preview icon* which can be found on the tool bar which appears at the top and bottom of the test design screen as seen in [Figure 2](#).

Global Test Settings

Every test has a set of global parameters which can be accessed by tapping on the *gear icon* in the tool bar at the top of the test design screen. Options include:

- A test can be shown in a single scrollable web page or each question can be shown on its own screen.
- Each question can also be timed in order to assess the cognitive load on the respondent.

Audio Playback Settings

Loop times

0

Set the number of times you wish the audio to loop. Setting this value to "0" will result in infinite looping.

- Require the user to listen to each example in full (at least once)
- Disable audio player time scrubber (user can't skip forward or back)
- Show section looping controls to user

CANCEL
SAVE

Figure 7 Audio Playback Settings.

Editing a Test

After inserting all questions in a test, you may need to edit or change some attributes of the test. The system contains some useful tools to make editing quick and easy. Tools include:

- Reorder questions using drag and drop or up/down arrows to the right of each question card.
- Make a question *required* in order to proceed through the test
- Skip subsequent questions based on the answer to the current question. See [Figure 4](#).
- Duplicate a question and its playback settings. This is achieved using the *copy* icon on the tool bar of each question card. See [Figure 4](#).

Audio Test Types

The system currently supports the following tests: *Multiple Stimuli with Hidden Reference and Anchor* (MUSHRA) [1], *Absolute Category Rating* (ACR) [2] and *A/B comparison testing* [2]. In all test types, audio switching is synchronous and seamless. This allows the respondent to switch audio conditions without hearing any discontinuities in the audio playback. A loop tool is optional in all test types.

A/B Comparison

An *A/B* test is used to compare two versions of the same piece of audio where each version has been subjected to a different version of the same process. When presented with the test, the respondent will see an interface similar to that in [Figure 8](#). The respondent may interact in the following ways:

- Play the audio
- Switch between a reference and condition A or B synchronously.
- Set loop start and end points for repetition.
- Change the playback position between the start and end of the clip.
- Select a preference from a radio group containing multiple options.

Absolute Category Rating (ACR)

An ACR test is used to assess the quality of a piece of audio on a labeled 5 point scale. When presented with the test, the respondent will see an interface similar to that in [Figure 9](#). The respondent may interact in the following ways:

- Play the audio.
- Set loop start and end points for repetition.
- Change the playback position between the start and end of the clip.
- Select perceived quality from a 5 point slider

Multiple Stimuli with Hidden Reference and Anchor (MUSHRA)

A MUSHRA test is used to assess the relative quality of multiple audio stimuli on a continuous scale ranging from 0 to 100. Within the stimuli there is a hidden reference and an anchor. The test is more flexible than the *ITU-R BS.1534-3* standard and allows the user to add or omit anchors. When presented with the test, the respondent will see an interface similar to that in [Figure 10](#). The respondent may interact in the following ways:



Figure 8 AB Participant View.

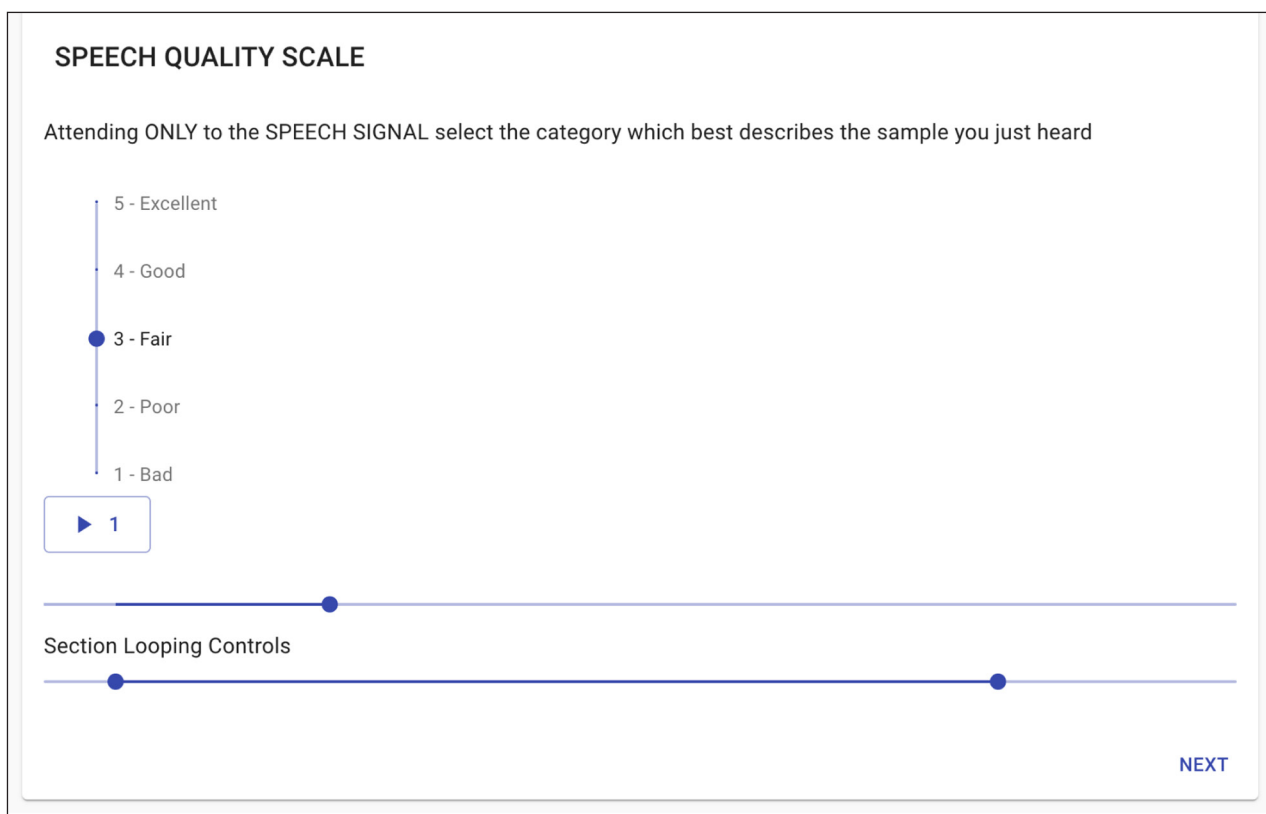


Figure 9 ACR Participant View.

- Play the audio.
- Switch between a reference and multiple stimuli synchronously.
- Set loop start and end points for repetition.
- Change the playback position between the start and end of the clip.
- Select perceived quality of each of the stimuli on a continuous slider in the range 1–100.

Randomisation

Test order randomisation is supported for all audio test types. This is achieved during survey creation by adding a *Group Divider* before and after a group of audio tests as shown in [Figure 11](#). Randomisation of stimuli within a single MUSHRA audio test is also supported and can be activated within the question settings menu of any MUSHRA audio example as seen in [Figure 12](#).

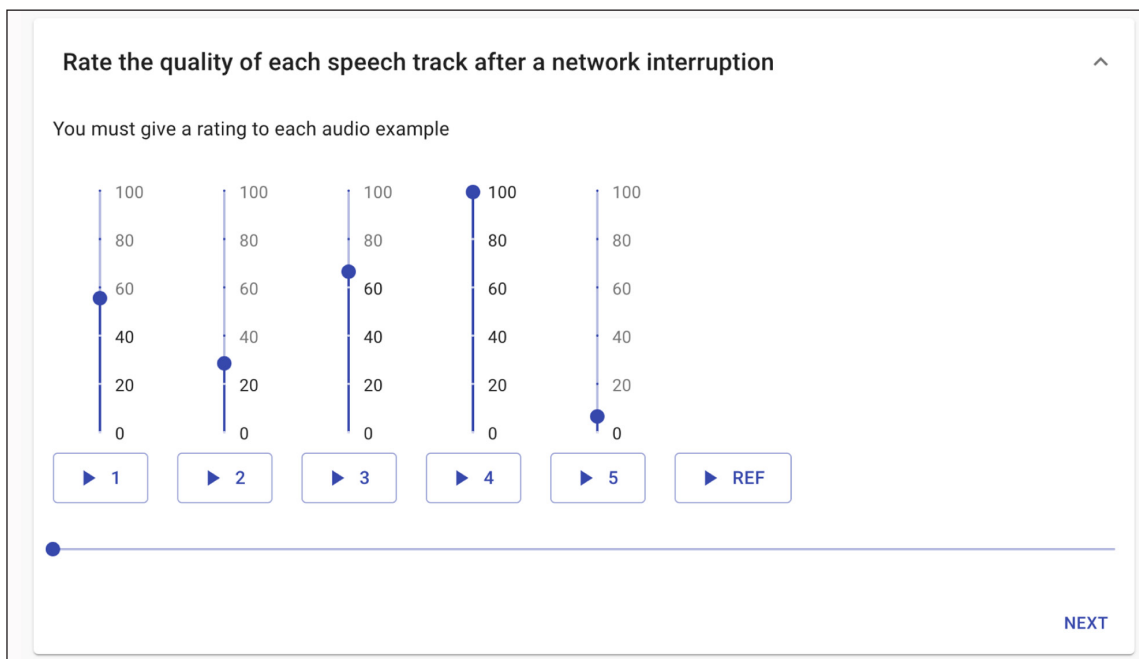


Figure 10 MUSHRA Test Interface.

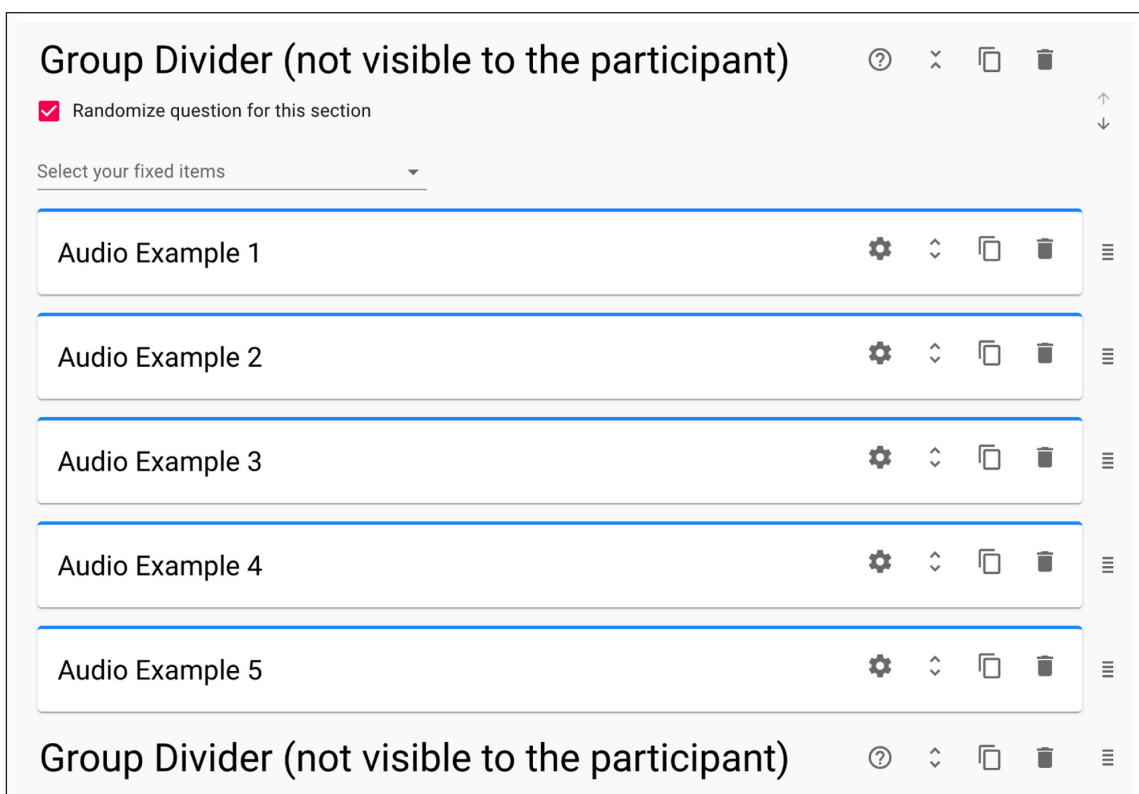


Figure 11 Test Order Randomisation.

Accessing Test Results

Once the link to a test has been shared, the system will collect responses from any respondent who completes the test. The test owner can check the responses at any time from the *Responses* tab at the top of the test design screen, [Figure 2](#).

The response view shown in [Figure 13](#) allows a user to:

- View and delete individual responses.
- Delete all responses.
- Download the results in CSV format.
- Download the results in JSON format.

Respondents also have the ability to delete their response using a unique link furnished to them at the end of a completed test. In this instance, the test owner will

Audio Playback Settings

Loop times
0

Set the number of times you wish the audio to loop. Setting this value to "0" will result in infinite looping.

Require the user to listen to each example in full (at least once)

Disable audio player seek bar (User cannot skip forward or back in time)

Show section looping slider bar

Randomize media files

CANCEL SAVE

Figure 12 Stimuli Randomisation within a MUSHRA test.

← AB Test Responses

QUESTIONS RESPONSES

| <input type="checkbox"/> | Test Name | Submitted At | |
|--------------------------|-------------------------|----------------------|---|
| <input type="checkbox"/> | Audio Preference Survey | 23/09/2020, 13:48:00 | 👁 |

Rows per page: 10 1-1 of 1 < >

<> DOWNLOAD IN JSON DOWNLOAD IN CSV

Figure 13 Results View.

receive a notification requesting that the updated results be downloaded.

Accounts

A test owner may update their password or delete their account from the *Accounts* screen (see [Figure 14](#)) which can be accessed from the navigation menu in the left pane, [Figure 1](#).

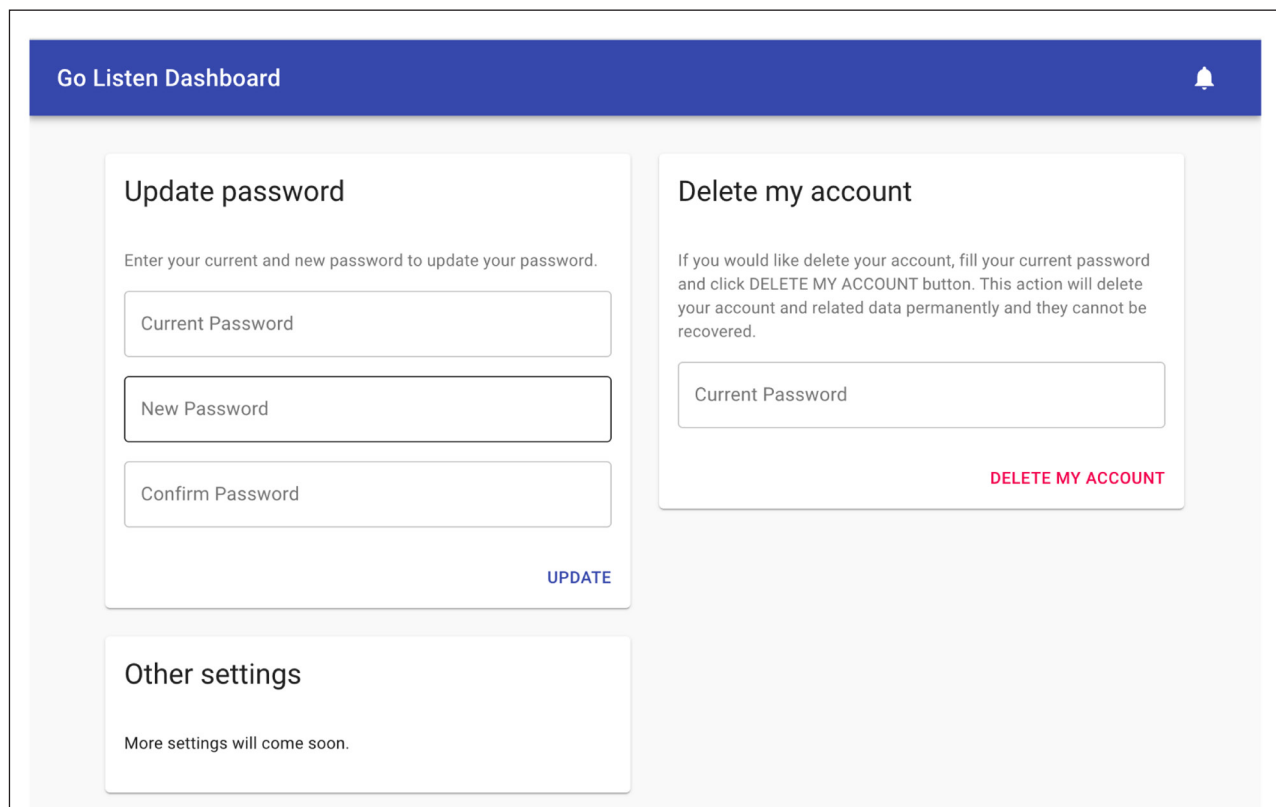
IMPLEMENTATION AND ARCHITECTURE

The software is separated into two independent parts: a front-end and a back-end which serves a database. These two parts are currently served on the same server with different processes but could easily be served

independently in a microservices style architecture. The front-end is implemented in React.js and requests data from the back-end which uses Python to make requests to the database which is implemented in MongoDB. The architecture is RESTful. In addition to source code, we have also provided the software both as a Docker install and an AWS AMI (Amazon Machine Image). Details can be found in the GitHub repository [13].

Docker

For the Docker install, there are 3 containers: frontend, backend and database, all working on the host network. This means each container must have some ports open



The screenshot shows the 'Go Listen Dashboard' with a blue header. Below the header, there are three main sections:

- Update password:** A section with the instruction 'Enter your current and new password to update your password.' It contains three input fields labeled 'Current Password', 'New Password', and 'Confirm Password'. A blue 'UPDATE' button is located at the bottom right of this section.
- Delete my account:** A section with the instruction 'If you would like delete your account, fill your current password and click DELETE MY ACCOUNT button. This action will delete your account and related data permanently and they cannot be recovered.' It contains one input field labeled 'Current Password' and a red 'DELETE MY ACCOUNT' button at the bottom right.
- Other settings:** A section with the text 'More settings will come soon.'

Figure 14 Account Management View.

and connected to local ports of the host. For example, the frontend has ports 80 and 443 opened, which are connected to ports 80 and 443 of the host.

Amazon Machine Image (AWS AMI)

AWS AMI is based on Ubuntu 18.04 with the latest Docker files installed. The configuration is the same as above.

QUALITY CONTROL

Some components contain unit testing within the code but in general, end-to-end testing was primarily used to ensure the quality of the software. Alpha testing was carried out by 5 researchers at QxLab in University College Dublin. Testers were instructed to create variants of each test type and to provide test responses in order to ensure that the data is collected and stored as expected. This phase of testing also revealed many useful user experience optimisations. Beta testing was carried out by conducting two real-world studies using the software. The software was then released to some invited universities and companies where more feedback was gathered.

Data and Security

The web application is built with privacy in mind. End to end SSL encryption (256-bit) is used by default between client and server. All passwords are stored in encrypted format (256-bit) so are not human readable in the event of any breach. All passwords require a minimum length with a combination of

upper/lower case letters, numbers and symbols. The researcher can delete all and any data at any point in time using the survey deletion and account deletion tools we provide within the app. The respondent is completely anonymous. No login is required and the app collects no data beyond the survey responses. The app does not store any IP addresses or track the respondents in any other way. The respondent can delete their response to a survey at any time using a unique link provided to them at the end of each survey. The privacy statement we use for the hosted web application can be found here <https://golisten.ucd.ie/GoListenPrivacyStatement.html>.

(2) AVAILABILITY

The system is available as a web service at <https://golisten.ucd.ie/> where a user can create a free account and begin using the service immediately. We have also made the source code available for use under the MIT License.

OPERATING SYSTEM

The software can be run on many operating systems. The following systems were used during testing.

- Ubuntu 18.04 and Ubuntu 16.04
- Mac OSX 10.15 (Mojave)
- Windows 10

PROGRAMMING LANGUAGES

- Python 3.8.0
- Node.js 12.16.1
- Node.js (React, NPM)
- NPM 6.13.4
- Typescript 3.9.5
- Python (Tornado, PYMongo)
- MongoDB
- Docker scripts and compose scripts (yaml)
- Nginx configuration file (conf)
- Supervisor configuration file (conf)
- Update delivery scripts (Python)

ADDITIONAL SYSTEM REQUIREMENTS

Minimum system requirements:

- 1GB memory
- 8GB disk space
- SSH access to the server which necessarily requires access to the Internet

DEPENDENCIES

- material-ui/core: 4.10.0
- material-ui/lab: 4.0.0-alpha.56
- testing-library/jest-dom: 4.2.4
- testing-library/react: 9.5.0
- testing-library/user-event: 7.2.1
- types/jest: 25.2.3
- types/node: 14.0.11
- types/react: 16.9.35
- types/react-dom: 16.9.8
- types/react-router: 5.1.7
- types/react-router-dom: 5.1.5
- axios: 0.19.2
- formik: 2.1.4
- mobx: 5.15.4
- mobx-react: 6.2.2
- mobx-utils: 5.6.1
- react: 16.13.1
- react-dom: 16.13.1
- react-router: 5.2.0
- react-router-dom: 5.2.0
- react-scripts: 3.4.3
- ts-md5: 1.2.7
- typescript: 3.9.5
- uuidv4: 6.1.1
- react-markdown: 5.0.3
- pymongo: 3.11.0
- tornado: 6.0.4

LIST OF CONTRIBUTORS

- Qijian Zhang: Lead Developer
- Dan Barry: Project Manager, UX/UI, Developer

- Pheobe Wenyi Sun: Research and Testing
- Alessandro Ragano: Research and Testing
- Andrew Hines: QxLab Director, Research and Testing

All contributors are members of QxLab at University College Dublin, Ireland.

SOFTWARE LOCATION

Archive

Name: Listening Test Platform

Persistent identifier: <https://github.com/QxLabIreland/listening-test>

Licence: MIT

Publisher: QxLab, University College Dublin

Version published: 0.1.0

Date published: 24/05/20

Code repository

Name: Listening Test Platform

Persistent identifier: <https://github.com/QxLabIreland/listening-test>

Licence: MIT

Date published: 24/05/20

(3) REUSE POTENTIAL

The software itself is a listening quality survey creation tool so the reuse potential is limitless. Furthermore, the software is designed to be an end-user system, meaning that it can be used by non-technical users with no coding ability if it is setup correctly once. For example, within a university setting, an audio research lab could install and run the system on a server. The system once running, handles all user account creation and management for other members of the lab. The survey creation tools within the system follow common design patterns allowing non-technical users to create and share very complex surveys without needing any knowledge of coding. As described throughout the article, the user can choose from many standard survey components along with the custom audio test components we provide (AB, ACR and MUSHRA). Combining these components allows a user to achieve a wide variety of listening quality test scenarios.

Since the software provides a full end-to-end survey platform, it could also be extended to other media types such as image and video giving rise to even greater reuse potential. The existing framework already handles account creation, user management, data storage and the survey creation tools which can be used as the common core for any other media quality assessment tasks one might think of. Even beyond media quality, many of the core survey features could be reused for any bespoke platform requiring online survey delivery and data collection.

ACKNOWLEDGEMENTS

We would like to acknowledge all those at QxLab who helped test the software.

FUNDING STATEMENT

Funded in part by the Science Foundation Ireland (SFI), and the European Regional Development Fund under Grant 12/RC/2289_P2 and Grant 13/RC/2077_P2.

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Dan Barry  orcid.org/0000-0002-8548-1549

Developer – School of Computer Science, University College Dublin, Ireland

Qijian Zhang  orcid.org/0000-0002-0120-7877

Developer – School of Computer Science, University College Dublin, Ireland

Pheobe Wenyi Sun  orcid.org/0000-0002-2858-0153

School of Computer Science, University College Dublin, Ireland

Andrew Hines  orcid.org/0000-0001-9636-2556

Developer – School of Computer Science, University College Dublin, Ireland

REFERENCES

1. **ITU-R**. “BS.1534-3: Method for the subjective assessment of intermediate quality level of audio systems.” *Int. Telecomm. Union, Tech. Rep.*, 2015.
2. **ITU-T**. “P.800: Methods for subjective determination of transmission quality.” *Int. Telecomm. Union, Tech. Rep.*, 1996.
3. **Cartwright M, Pardo B, Mysore GJ, Hoffman M**. “Fast and easy crowdsourced perceptual audio evaluation.” In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016; 619–623. DOI: <https://doi.org/10.1109/ICASSP.2016.7471749>
4. **Kraft S, Zölzer U**. “Beaqlajs: HTML5 and javascript based framework for the subjective evaluation of audio quality.” In *Linux Audio Conference*, Karlsruhe, DE, 2014.
5. **Jillings N, De Man B, Moffat D, Reiss JD**. “Web audio evaluation tool: A browser-based listening test environment.” In *12th Sound and Music Computing Conference*, 2015.
6. **Schoeffler M, Bartoschek S, Stöter F-R, Roess M, Westphal S, Edler B, Herre J**. “webMUSHRA – A Comprehensive Framework for Web-based Listening Tests.” *Journal of Open Research Software*. 2018; 6(1): 8. DOI: <https://doi.org/10.5334/jors.187>
7. **Guse D, Orefice HR, Reimers G, Hohlfeld O**. “Thefragebogen: A web browser-based questionnaire framework for scientific research.” In *11th International Conference on Quality of Multimedia Experience, QoMEX*, 2019. DOI: <https://doi.org/10.1109/QoMEX.2019.8743231>
8. **Naderi B, Cutler R**. “An Open Source Implementation of ITU-T Recommendation P.808 with Validation.” In *Proc. Interspeech 2020*. 2020; 2862–2866. [Online]. DOI: <https://doi.org/10.21437/Interspeech.2020-2665>
9. **MATLAB**. The Mathworks, Inc., Natick, Massachusetts, 2020.
10. **MAX**. Cycling74, 340 S. Lemon Avenue 4074 Walnut, CA 91789 USA, 2020.
11. **ITU-T**. “P.808: Subjective evaluation of speech quality with a crowdsourcing approach.” *Int. Telecomm. Union, Tech. Rep.*, 2018.
12. Mdn webdocs – media container formats. 2020 [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Containers>.
13. Go listen – github repository. 2020 [Online]. Available: <https://github.com/QxLabIreland/listening-test>.

TO CITE THIS ARTICLE:

Barry D, Zhang Q, Sun PW, Hines A 2021 Go Listen: An End-to-End Online Listening Test Platform. *Journal of Open Research Software*, 9: 20. DOI: <https://doi.org/10.5334/jors.361>

Submitted: 14 January 2021 Accepted: 07 July 2021 Published: 20 July 2021

COPYRIGHT:

© 2021 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

Journal of Open Research Software is a peer-reviewed open access journal published by Ubiquity Press.