A Web-App for Collecting Semantic Fluency Data

Danyi He

Master of Science
Computer Science
School of Informatics
University of Edinburgh
2023

Abstract

The semantic verbal fluency test is an efficient way to detect cognitive dysfunction. However, under current circumstances, due to social-distance requirements caused by COVID-19, the traditional method, a face-to-face interview, to conduct an SVF test is unlikely to carry out. This study created a web application supporting tele-assessment for clinical practice and data collection for academic purposes. This application integrates recording, volume adjustment, noise filtering and other functions and is deployed in and uses Google cloud platform services to ensure system performance and data transmission security.

Acknowledgements

During this project, I received a great deal of support and assistance.

I want to thank my supervisor Professor Maria Wolters. Without her guidance, timely and insightful feedback, and encouragement, I would not finish this work so smoothly.

I also want to thank Yasa Kostas Rabia, who gave me advice and help in the past five months. She was always there when I had troubles. Her patience and warm heart impressed me, and her academic attitude inspired me to work hard.

I want to acknowledge my family. With their support, I had the opportunity to see the outside world. I had the chance to be more independent. They concerned me and gave me the best so that I could handle all the problems with more strength and confidence.

In addition, I want to thank my friends and my partner. My solitary life got enriched with their company so that I felt not so lonely in the lockdown.

Loneliness, academic pressure and the cancellation of air tickets, again and again, tortured my body and mind, but I thank myself for not being defeated at that time. I am proud that I went through all the bad things.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Danyi He*)

Table of Contents

Chapter 1

Introduction

1.1 Motivation

In the neuropsychology field, the semantic verbal fluency (SVF) test is widely accepted as an efficient approach to analyse cognitive functions in low technology environment[1][2]. The test requires participants to name words in several categories in a limited time. The most common way to do the SVF test is an in-person interview. Nevertheless, this kind of method is time-consuming, human-intensive, and requires an office visit[3]. Besides, the coronavirus pandemic outbreak meets us a remote working under the lockdown rules, which forces scientists to use techniques to collect data. However, there are three types of state-of-the-art recording applications but not suitable for SVF tests. One type aims for music production, for example, FL Studio[4] and Adobe Audition[5]. With complicated interfaces, this kind of application requires professional knowledge to operate. Another type focuses on note-taking, such as built-in Voice Memos applications, which have simple interfaces but fail to limit the recording time. The last type is phone call recording applications, such as Google Voice[6]. These three types can only save the audio files locally. As a result, users need to send the files to the researchers manually through other applications. In this process, confidential problems like personal information and file leakage exist.

Therefore, a web application for collecting SVF test data is in need. It can also be used in telehealth, which supports clinic practice, reduces human force consumption, and allows parallel testing[7].For test-takers, they can do the test anytime and anywhere. The whole process is self-controlled. They can decide when to start the test, and during the test, they can take a break after each subtest without the time limitation. For researchers who work in the neuropsychological field or have a massive demand for SVF data, by utilising this application, they can quickly obtain data from users regardless of the geological and other factors. For example, during an interview conducted on April 8, 2021, with Yasa Kostas Rabia, a PhD student studying at the University of Edinburgh, she stated this application would help her study that aims to automated analysis of SVF data from different parts of Turkey and the Turkish diaspora. Besides, doctors who diagnose patients' brain condition rather than make a face-to-face appointment can acquire patients' information immediately, even when they are far away.

1.2 Objectives

This project aims to create a web application for remote SVF tests. This application should meet the following requirements:

- Basic recording functions. Functions like starting and stopping recording, and transferring the signal to WAV format should be implemented in this application.
- Recording time control. In the traditional SVF test, speaking is restricted to 60 seconds generally, and sometimes 90 seconds, so this application's recording period should be limited.
- User-friendly interfaces. The target user is not expected to be high-educated and familiar with computers, so the interfaces should be easy for those with basic computer skills to use.
- Automatic uploading. Once a test finishes, the audio files should be automatically and directly to the storage specified by the researchers.
- Background noise reduction. Data quality is essential in the research. If the noise is too loud, some information might be covered. However, a filter is optional in this project because the signal quality also reduces when filtering the noise.

1.3 Thesis Structure

The structure of the thesis is:

- **Chapter 2:** This chapter introduces telehealth, the definition of semantic verbal fluency test, and the data collection processes.

- **Chapter 3:** This chapter discusses the methodology used in this project. It first describes the design principles and the system design. Subsequently, it illustrates the frameworks utilised in client- and server-side programming. Then, it introduces the web audio APIs and Python modules applied to implement recording functions. It also explains how noise filter works and how to achieve it. Furthermore, it gives information of what platforms are used, and acknowledges the read of the principles of design a web application and how to design and evaluate the application.
- **Chapter 4-6:** These chapters depict the functions achieved and the evaluation processes in three iterations, respectively.
- **Chapter 7:** This chapter gives some ideas for future improvements of the system, including the improvement in noise filtering and the connection with the Qualtrics survey.
- **Chapter 8:** In this chapter, I conclude the whole process of the project, analyse the results.

Chapter 2

Background

2.1 Benefits and concerns on telehealth

Under the COVID-19 crisis, the benefits of telehealth reveal. It can reduce the use of medical institutions, improve access to care, and avoid infection from person to person[8]. In response to travel restrictions, quarantines, and limited medical resources, many cognitive assessments used to rely on interpersonal procedures have changed into a remote way[9].

However, in [10], the authors asserted that there are concerns for both patients and therapists. For patients, they might think telemedical is the second-best compared to face-to-face therapy. It might be hard for therapists to find a trustful software platform that works like a regular pen and paper method.

2.2 Semantic Verbal Fluency Test

Verbal fluency test is frequently used in neuropsychological assessments. Past studies have shown that when people think of a word, other related or semantically similar words will be activated[11]. Cognitive flexibility, information classification, and inhibitory capability will be triggered in this process[12]. There are two types of the verbal fluency test: phonemic and semantic[13]. In a phonemic test, test-takers must name all of the words that begin with that particular letter. In an SVF test, the test-taker can speak as many words as possible in a given semantic category, such as animals or fruits, and in a limited time, typically 60 seconds. For this dissertation, I only investigate a solution for collecting semantic verbal fluency (SVF) data.

2.3 Data collection in SVF tests

The standard approach to conduct an SVF test is a person-to-person interview. In this situation, pen and paper is the preferred way to collect data. Some researchers count the number of words during the interview. For instance, in [2], the researchers recorded scale scores, which means the number of correct words meeting the requirements when participants did two one-minute SVF tests. While in some studies, the examiner wrote down the words generated by the test takers[14][15].

However, many automated analysis methods rely on data such as the speech containing more information than word count, like pausing, speaking time, and error rate. To collect sound data, [16] tape-recorded the responses, and [17] used a digital voice recorder during the interviews.

In emergencies or when environmental or biological hazards occur, such as lockdown resulting from the COVID-19 pandemic, telehealth replaces in-person meetings to conduct medical practice[7]. For academic purposes, many research had used remote solutions like phone calls and computer-based techniques to collect data before COVID-19. It was shown in [18] that the authors used a telephone to conduct SVF tests and proved the effectiveness of telephone-based methods in dementia automatic front-line screening based on telephone-sampled speech. Another approach is utilising computer-based methods to collect audio data. The authors of [19] validated the feasibility to use an iPad application to administer SVF tests and audio-record responses. Later, as mentioned in [1], the researchers recorded the test by an automated recording application, which is provided by the University of Toronto, Canada and the Winterlight Labs. Based on the recordings, they obtained word count and investigated the automatic evaluation method. Another example of using speech samples to develop analysis methods is that, with 98 audio recordings, the authors of [20] found a multi-feature automatic way to analyse the verbal fluency test, including count-based and time-based features.

Chapter 3

Methodology

3.1 Design principles

3.1.1 Human-centered design

Human-centred design[21] is a practical approach adopted in the design development process to achieve a usable system.

In [22], it mentions a method to support human-centred design is iterative design. As presented in figure 3.1, in every iteration, the developer redesigns the system based on the feedback of the previous system version from end-users.

Based on the human-centered design, this project consists of three iterations. The first iteration is the minimal viable version which has basic functions like:

1. Recording functions: starting recording, stopping recording both manually and automatically, playback, formatting data into a WAV file.
2. Visualization of audio signal.
3. Uploading files to the Cloud Storage.
4. Updating the database.

In the second version, according to the evaluation of the previous version, some functions were updated and some new functions were added:

1. Another visualization of the sonic signal, a bar.
2. Filter implemented by Web APIs.
3. Volume Control.

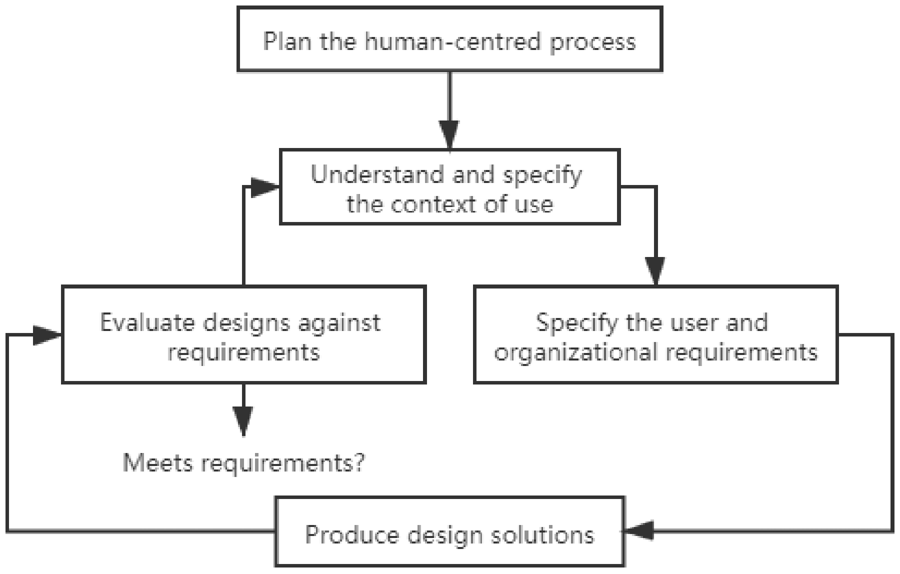

Plan the human-centred process
Understand and specify the context of use
Specify the user and organizational requirements
Produce design solutions
Evaluate designs against requirements
Meets requirements?

Figure 3.1: The cycle of human-centred design. Figure source: [22]

Subsequently, after the assessment of the second version, the modification in the final iteration focuses on the filter design:

1. Filter implemented by SciPy signal module.
2. Saving both original and filtered files.

3.1.2 Heuristic rules

A heuristics evaluation is judging the interfaces based on some heuristic rules. In this project, during the whole process, the design followed Jakob Nielsen's ten general principles for interaction design[23]:

1. Visibility of system status. The interfaces should always inform users of the status of the system. For example, give them timely feedback, and show progress. In this system, the progress bar on the top of interfaces (figure 4.3, 4.4, 5.3, 4.5, 4.6, 5.4, 5.5, 4.7, **??**) indicates what stage the user is, and the text in the break page (figure 5.6) shows the progress of the test.
2. Match between system and the real world. The content in the interfaces, such as words, icons, and buttons, should be uniform with the use in real-world con-

ventions. In this system, the buttons and icons are used traditionally and hiding time while recording matches the fact that users are not allowed to know the remaining speak time in face-to-face tests.

3. User control and freedom. Users might click some buttons or enter an unwanted page by mistake, so it is necessary to have an emergency exit when this situation happens. So there is an exit button in the upper right corner to let users quit at any time.

4. Consistency and standards. The representations should not be vague and follow industry standards. Specifically, the descriptions in the interfaces are similar to the sentences used in the traditional SVF tests in this system.

5. Error prevention. Before the user makes a mistake, an error message hints them the consequence of this action and asks them to confirm their action. So the mistake can be eliminated in the first place.

6. Recognition rather than recall. The design of interfaces should not ask users to interact with one interface based on the memory of another interface, so the instruction in each interface is always clear to make them aware of the target.

7. Flexibility and efficiency of use. For both novice and experienced users, the interfaces meet their needs.

8. Aesthetic and minimalist design. The interfaces only contain valuable and essential components.

9. Help users recognise, diagnose, and recover from errors. Once an error occurs, a message explains what the error is and how to solve it.

10. Help and documentation. There should be instructions that help users to complete the tasks. Especially in this application, the target users are new to this system and might not be familiar with the advanced computer operations. Therefore, it is essential to provide clear documents.

3.2 System structure

Figure 3.2 is an overview of the system design. The system is built on Django framework (section 3.3.1), and the frontend is based on Vue.js framework (section 3.3.2).

In the frontend, an AudioContext (section 3.4.1) obtains the user microphone media source by a MediaStreamSourceNode (section 3.4.2) and then passes the signal to the GainNode (section 3.4.3) to change the amplitude, later on, an optional node, BiquadFilterNode (section 3.4.4), cleans the data. Finally, after ScriptProcessorNode (section 3.4.5) tranforms the format of data, the processed data is transfered to the destination of the AudioContext. Before sending an HTTP request to the server, the data is encoded into WAV format and encapsulated into a Blob object.

In the backend, URLs match the request interface with the corresponding view. Subsequently, the view receives the HTTP request, extracts the data, deals with the data, stores the WAV files in Google Cloud Storage (section 3.7.1), and inserts records into the MySQL database on Google Cloud SQL (section 3.7.3).

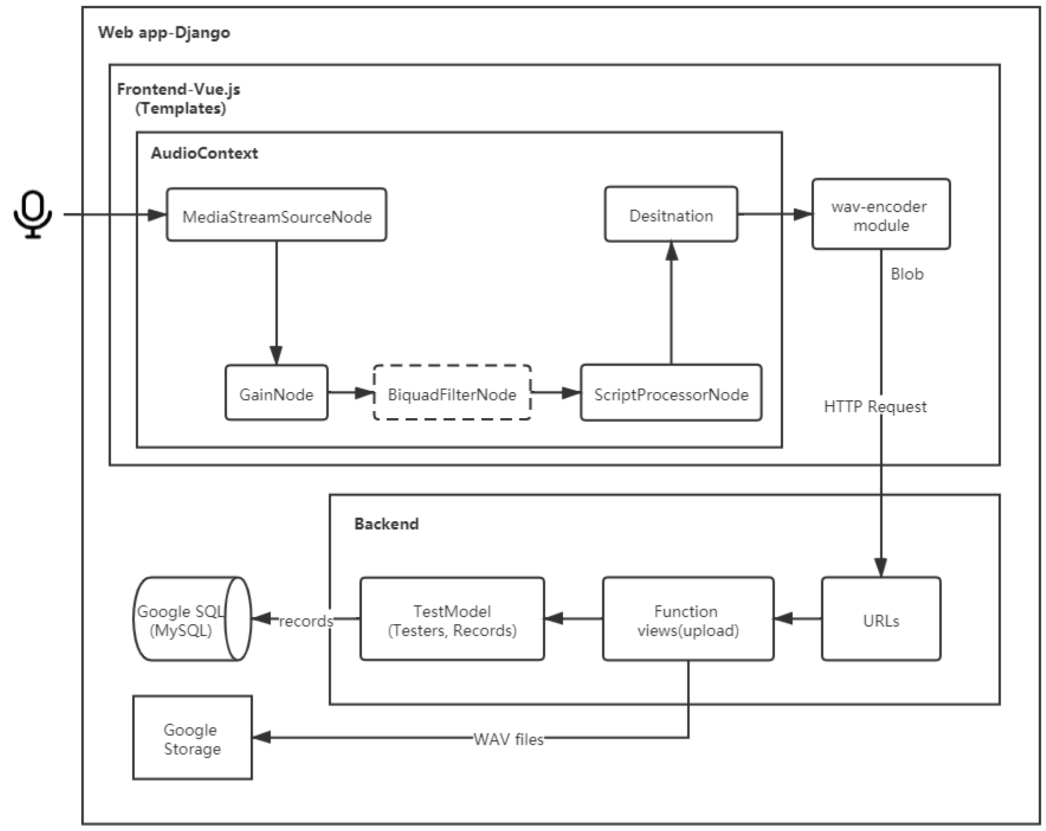


Figure 3.2: The structure of the system.

3.3 Frameworks

3.3.1 Django

In this project, Django[24] is the web framework. It is written in Python and applies the model–template–view(MTV) design model. As shown in figure 3.3, the model layer is responsible for object-relational mapping (ORM)[25], which means establishing the relation between the object model of programming language and the relational model of the database. So the database operations like adding, updating, deleting and searching records can be done using the programming language. The template layer encapsulates the logic responsible for processing user requests and returning responses. As for the view layer, it deals with business logic and calls the models and templates at the appropriate time. Such a low coupling model in which each of its components has little or no knowledge of the definitions of other separate components allows the separate implementation of the frontend and backend[26]. Furthermore, Django allows developers to build a website quickly because it integrates a wealth of standard components for WEB development, such as user authentication, paging, middleware, cache, session.

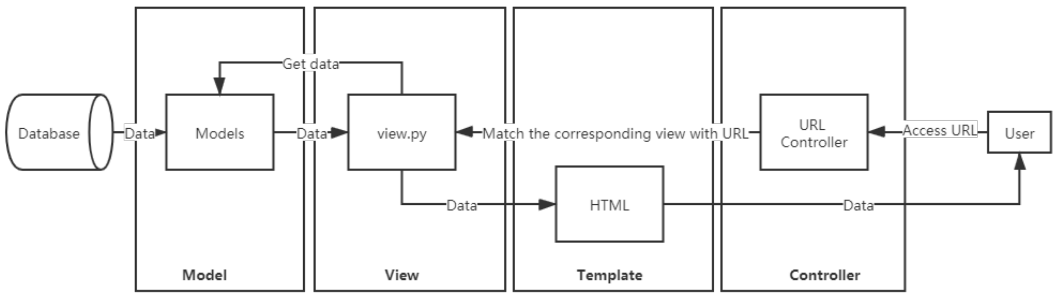


Figure 3.3: The workflow of MTV design model in Django. Figure source: [27]

3.3.2 Vue.js

In the frontend implementation, the framework is Vue.js[28]. Vue.js is a progressive JavaScript framework for building user interfaces. Three main characteristics of this framework are:

- **Declarative, reactive data binding.** Vue.js can automatically respond to data changes, and it modifies all the data and views that are bound together according to the declared binding relationship. This is a two-way binding that enables the change of data and view to happen simultaneously.

- **Componentized development.** The module encapsulation can split various modules into separate components, making implementation easier by allowing the reuse of components. On the parent page, these components can be utilized by announcing and passing parameters.
- **Virtual DOM.** Virtual DOM is a tree structure based on JavaScript objects (VN-odes) to simulate the DOM structure. This tree structure contains information about the entire DOM structure. Using virtual DOM, DOM comparison operations can be placed on the JS layer to improve efficiency and rendering performance.

3.4 Web audio APIs

3.4.1 AudioContext

The AudioContext interface[29] represents an audio processing graph constructed by linked AudioNode. These nodes are created by the AudioContext object and work sequentially to achieve the execution of audio processing. Therefore, announcing an AudioContext object is the first thing before any other operations when decoding a signal.

3.4.2 MediaStreamAudioSourceNode

The MediaStreamAudioSourceNode interface [30] is a kind of AudioNode that only has one output and no input. In the audio processing graph, it operates as the media source. The media, in this project, is only the user microphone and can be obtained using the WebRTC APIs[31], and its output connects to the input of GainNode.

3.4.3 GainNode

GainNode[32] controls the volume. It only has one property, gain, which changes the amplitude of the signal. The change of the value of gain immediately affects the input signal, resulting in a strange sound. Therefore, the volume control operation should only be done before recording. GainNode can connect to either BiquadFilterNode or ScriptProcessorNode.

3.4.4 BiquadFilterNode

BiquadFilterNode[33] is an AudioNode performing as a noise filter. The noise filter is chosen to be a band-pass filter to keep the signal from a specific frequency band designated by the frequency and quality factor parameters. The frequency controls the centre of the frequency band, and the quality factor describes the bandwidth. The higher the quality factor is, the frequency band is narrower[34].

3.4.5 ScriptProcessorNode

The ScriptProcessorNode interface[35] is used to do custom events. Once the input buffer has new data, it operates and outputs the processed one until the output buffer is full. In this application, the event in this node is to get the input audio data in the buffer, convert it into a 32-bit floating-point, and push it into an array. Also, it obtains the maximum signal power in each sample, which contributes to the visualization of the voice level.

3.5 Python modules

3.5.1 SciPy signal module

Scipy module[36] belongs to the Scipy package, which is used for signal processing. It has functions like convolution, peak finding, spectral analysis, filtering, and B-splines. This system benefits from the filtering functions, butter[37] and sosfilt[38].

In this application, a 10-order Butterworth band-pass filter[39] is created with the use of the Scipy.signal.butter function. Unlike the filter function mentioned in section 3.4.4, the frequency passband is defined by one parameter *Wn* that is a length-2 array, which is more straightforward to control than with BiquadFilterNode. Moreover, the filter can be an analog filter or a digital filter, and in this design, it is selected as digital. The output type is the second-order sections format used for general purposes to avoid numerical error with the transfer function format. Another function Scipy.signal.sosfilt filters data along one dimension by applying the second-order sections format output of Scipy.signal.butter.

3.5.2 uuid module

The system assigns users a unique label that contains no biographic information to different them. In this case, it is suitable to use a universally unique identifier(UUID) that has a fixed size and guarantees uniqueness in space and time through MAC address, timestamp, namespace, random number, pseudo-random number.

Python package provides a module named uuid[40] supporting four kinds of methods to generate UUID:

- uuid1(). This method is based on the current timestamp and MAC address, so it might intentionally expose the MAC address, which is unwanted in this application.
- uuid3(). By calculating the MD5 hash value of the namespace and the name to generate the UUID, this function ensures the uniqueness of different names in the same namespace and the uniqueness of different namespaces. However, the UUID is the same when the name and namespace are the same.
- uuid4(). In this approach, UUID is created by the random value.
- uuid5(). Like uuid3(), this method uses namespace and the name, but it applies another algorithm, SHA-1.

In this application, the uuid4() function is in use, which is the best way to hide user personal information.

3.6 Noise filter

Noise is an unwanted signal in audio recording, usually contains the background sound. It reduces data quality and even hinders information from data which makes the recording not useful for data analysis. So filtering noise is necessary for data collection.

In this project, two filtering techniques were tested. One is adding a BiquadFilterNode, as mentioned in section 3.4.4, in the recording process chain. Another is implemented in the backend using Python module SciPy.signal that is discussed in section 3.5.1. According to [41], most speech perception research only makes use of human voice with a frequency no more than 5KHz. So, both approaches use a band-pass filter to remove signals with very low frequencies and very high frequencies. Consequently, the human sound will remain, and most noise will be erased.

3.7 Google Cloud Platform

The deployment of this system is based on services provided by Google Cloud Platform (GCP)[42]. These services, including Cloud Storage, App Engine, and Cloud SQL, support computing, storing data, and deploying applications.

3.7.1 Google Cloud Storage

Cloud Storage[43] is an object storage system, which means its records keep both the name and the structure of data. In a typical storage system, the content of the file is rendered as a string of digits[44]. However, object storage can hold entire organised databases, raw video streams, or matrices for machine learning models [45]. Therefore, it is suitable for store audio data and allows users to listen to the audio file on the website. Each App engine instance allocates two storage buckets for free in the standard environment—one for application data and one for other demand. In this project, the static files and the audio data from participants are stored in these buckets.

3.7.2 Google App Engine

App Engine[46] enables developers to build applications remotely. In addition to the container being created on the same platform where it will be deployed, GAE provides the just-in-time compilers and interpreters required to run programs written in Python, JavaScript and other programming languages. Therefore, developers no longer need to maintain the server; they upload the application and immediately serve users. GAE has two environment settings: standard and flexible. The standard environment is like a managed virtual machine where the configuration is defined, while the flexible environment is a custom one[47]. So the standard environment is easier to deploy an application, and it is also the choice in this project.

3.7.3 Google Cloud SQL

Cloud SQL[48] is a relational MySQL database that manages and stores data. Google is responsible for database management, patch management, and replication to guarantee availability and performance. Moreover, it can be accessed by applications in GAE quickly.

Chapter 4

First iteration of the design

In the first iteration, based on the workflow of a face-to-face SVF test, a minimally viable version application was implemented as depicted in figure 4.1. This version achieves fundamental recording functions, supports a single test, and can automatically upload the file. Users enter the system, check the microphone by recording for a few seconds, and start the test. There would be 10 seconds to let them read the test category and speak for the next 70 seconds. In the end, the system would upload the audio file to Google Cloud Storage and insert records into Google SQL automatically.

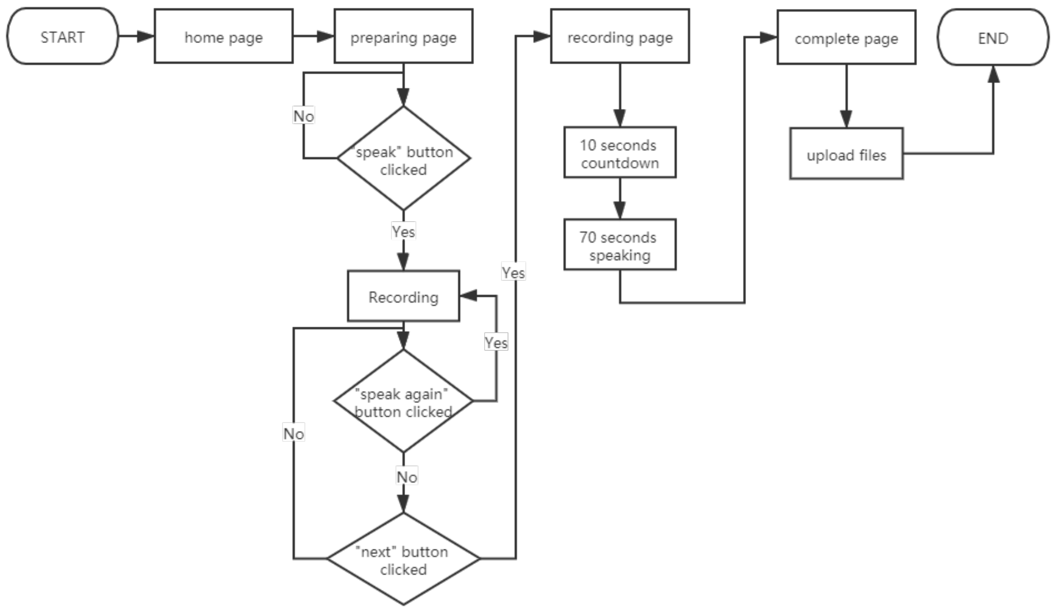


Figure 4.1: The workflow of the minimal viable version.

4.1 Implementation

4.1.1 Database design

There are two tables, testers and records. The table testers, as depicted in table 4.1, stores the information of the user, including user ID, age, gender. User ID is generated by the method described in section 3.5.2 which is the primary key. As for age and gender, they are information that might be helpful to future work, but currently, they are default values 22 and 1 (male). In the records table (table 4.2), the attributes are id, audio, test_time, and test_id. Attribute id is created in order by the database, and audio means the audio file name helping locate the file in the storage. Test_time indicates when the file is stored. There is a foreign key of this table, test_id, which is the primary key of the records table.

Attribute Name	Type	Length	Key
uid	varchar	64	True
age	int	11	False
gender	tinyint	1	False

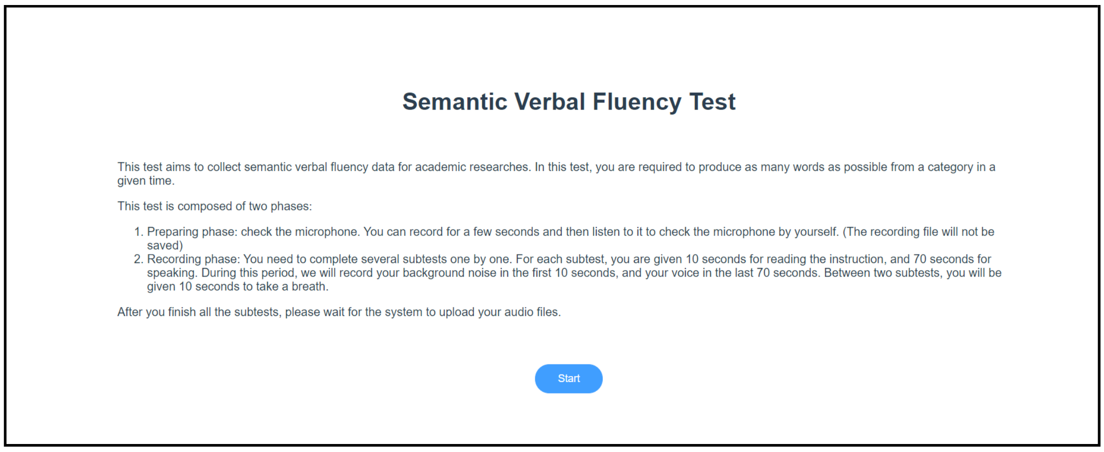
Table 4.1: The description of table testers

Attribute Name	Type	Length	Key
id	bigint	20	True
audio	varchar	128	False
test_time	datetime	6	False
tester_id	varchar	64	False

Table 4.2: The description of table records in the minimum viable version

4.1.2 Home page

This page (figure 4.2) is the first page of the system. It provides information on the test, including the purpose, the workflow of the system, and a brief description of each step. On the bottom, there is a button to enter the next page.



Semantic Verbal Fluency Test

This test aims to collect semantic verbal fluency data for academic researches. In this test, you are required to produce as many words as possible from a category in a given time.

This test is composed of two phases:

1. Preparing phase: check the microphone. You can record for a few seconds and then listen to it to check the microphone by yourself. (The recording file will not be saved)
2. Recording phase: You need to complete several subtests one by one. For each subtest, you are given 10 seconds for reading the instruction, and 70 seconds for speaking. During this period, we will record your background noise in the first 10 seconds, and your voice in the last 70 seconds. Between two subtests, you will be given 10 seconds to take a breath.

After you finish all the subtests, please wait for the system to upload your audio files.

Figure 4.2: The home page of the minimum viable version.

4.1.3 Preparing page

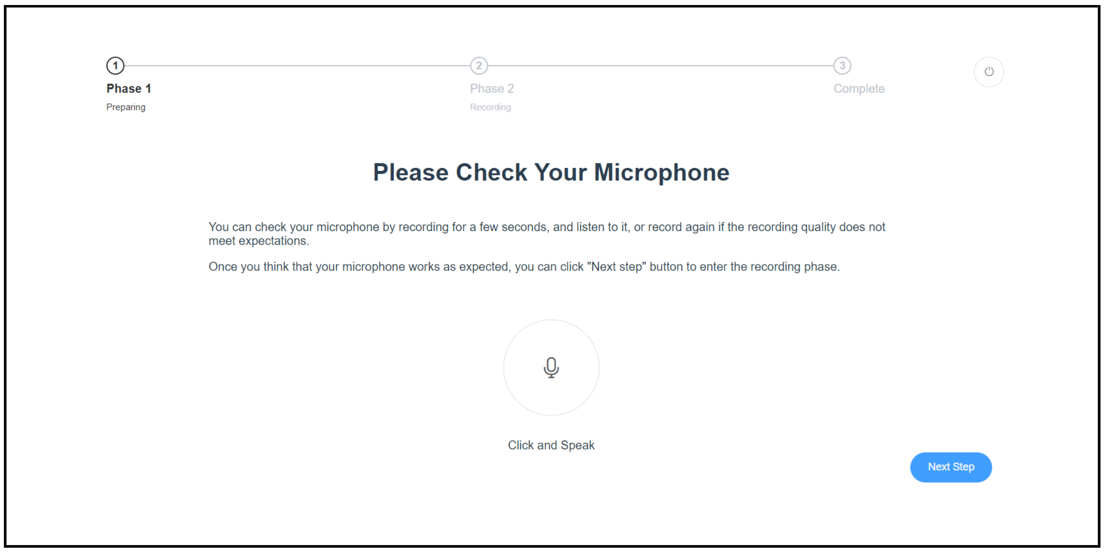
In the minimum viable version, checking the microphone is by listening to the recording, as shown in figure 4.3 and 4.4. Once the user is confident with their recording quality, they can enter the test part. Otherwise, tune the microphone and record again.

The recording function in this page relies on web audio APIs talked in section 3.4 and HTML DOM audio object, which supports playback in the interface. Once the user clicks the circle button with the microphone icon, the recording starts. After the recording stops, the signal will transform into WAV format and be stored in the temporary storage, then load into the audio object so the user can play it.

The process of transforming sample data into a WAV file is based on a Binary Large Object (Blob)[49], a container to store raw data. In this application, it is constructed by an array containing the sample data and information, including RIFF chunk length[50], format chunk length, sample rate, channel count, block align (channel count * bytes per sample), byte rate(sample rate * block align), sample format, format chunk length, bits per sample, and data chunk length. The MIME type[51] of the data will be stored in the Blob is *audio/wav*.

4.1.4 Recording page

When the user is redirected to the recording page, firstly, there would be 10 seconds for them to read the instruction (figure 4.5), and following, 70 seconds for them to speak (figure 4.6). The reason to record for 70 seconds rather than 60 seconds is that Yasa Kostas Rabia stated in the interview (personal communication, April 16, 2021) that



Please Check Your Microphone

You can check your microphone by recording for a few seconds, and listen to it, or record again if the recording quality does not meet expectations.

Once you think that your microphone works as expected, you can click "Next step" button to enter the recording phase.

Figure 4.3: The preparing page of the minimum viable version.

someone who is unfamiliar with the system might be too early or waste few time to speak. Therefore, if recording for precisely one minute, some parts might be missed. Moreover, there is no countdown during speaking because participants cannot know how long they have spoken in a real test.

For visualising real-time audio signals, a canvas is used. First, build a two-dimensional rendering context of the canvas. Then create an analyser of the 2D context and apply it to copy the 32-bit floating-point array containing PCM channel data into a Uint8Array (unsigned byte array). This processed array can then be plotted on the canvas, which presents as a wave line.

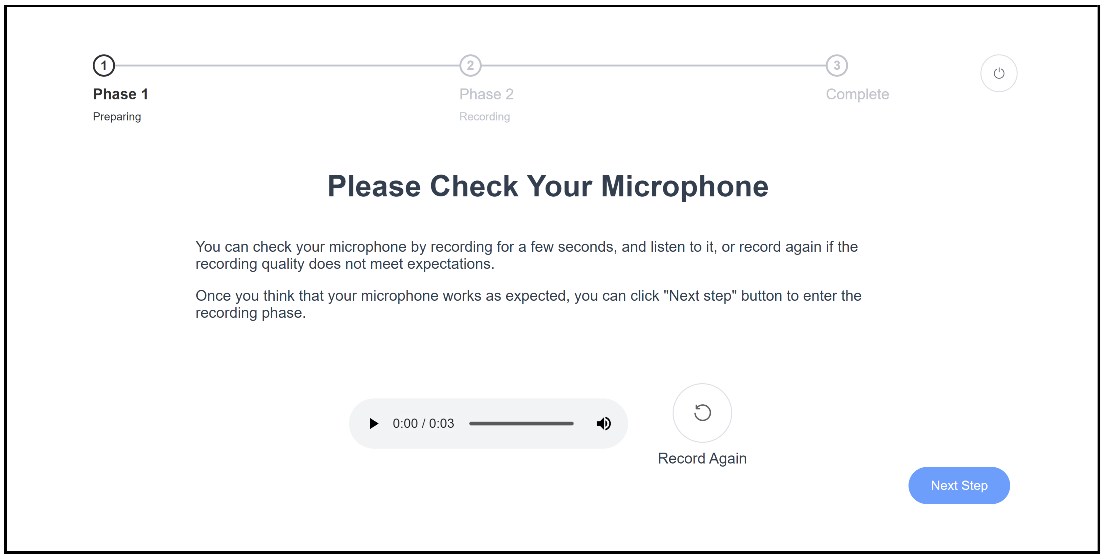
4.1.5 Complete page

The file is uploaded on this page. As the file has been saved successfully and safely, this page shows to inform users of the end of the test and thank them for their participation.

4.2 Evaluation

4.2.1 Aims

Evaluation at this stage, on the one hand, confirms that the software has done as expected, and on the other hand, confirms that the workflow and interfaces of this system

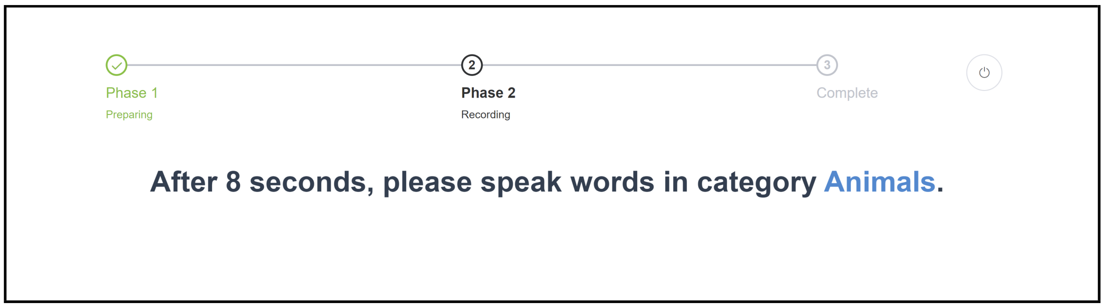


Please Check Your Microphone

You can check your microphone by recording for a few seconds, and listen to it, or record again if the recording quality does not meet expectations.

Once you think that your microphone works as expected, you can click "Next step" button to enter the recording phase.

Figure 4.4: The preparing page during checking the microphone in the minimum viable version.



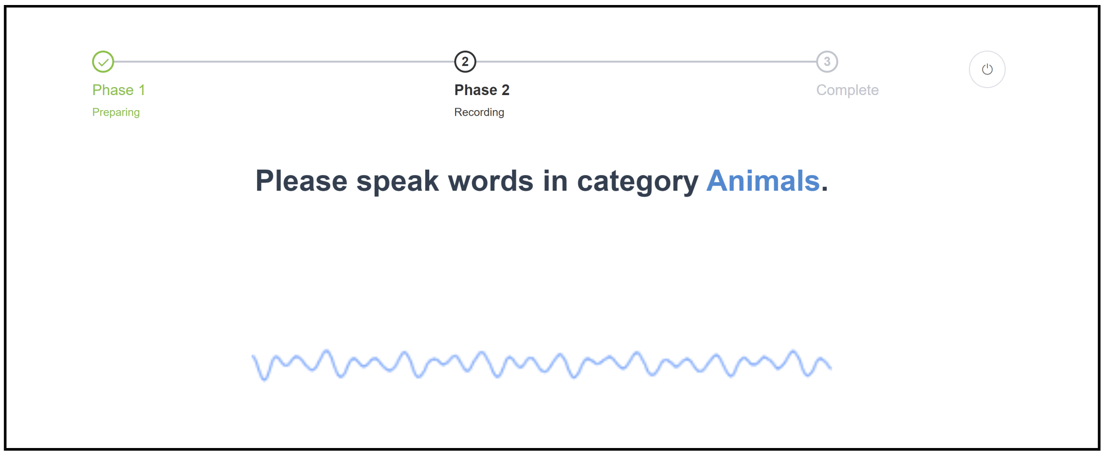
After 8 seconds, please speak words in category Animals.

Figure 4.5: The recording page of the minimum viable version.

fit the real semantic verbal fluency test. Additionally, by doing user tests, feedback from professional people like those who are now studying PhD in the psychology field or informatics field and who have already done the degree can be collected to improve the system.

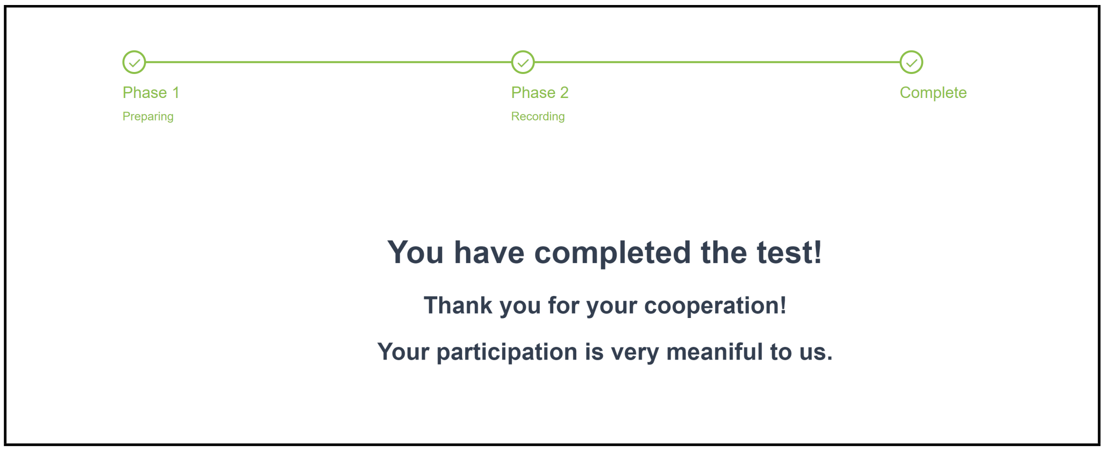
4.2.2 Procedure

In this pilot test (R. Yasa Kostas, personal communication, June 4, 2021), no personal information was stored. Participants entered the system and finished the test then gave some feedbacks.



Please speak words in category Animals.

Figure 4.6: The recording page of the minimum viable version.



You have completed the test!

Thank you for your cooperation!

Your participation is very meaniful to us.

Figure 4.7: The complete page of the minimum viable version.

4.2.3 Results

From the feedback from participants and the advice of Professor Maria Wolters expressed in interview (personal communication, June 9, 2021), the system has essential functions to complete an SVF test and would give a result close to what from a face-to-face interview. However, in the aspect of appearance, the interfaces need to be more uniform. For example, the sonic animation is different when the user records on the preparing page and recording page. There is no animation but a button to stop recording on the preparing page, while there is a blue wave line on the recording page, which might confuse the users when they name the words. About the system setting, most participants think 10 seconds before recording is enough to read instructions and prepare well, but some said they need more than 10 seconds. As for the instructions, Maria said it needs to be more precise, and some participants recommended adding some tips to teach the user how to lower the background noise:

- Close the window and door,
- Move to a quiet room,
- Be sure no other sounds in the place/environment,
- Keep the microphone 20-30 cm away from your mouth, and not too close or far.

Chapter 5

Second iteration of the design

The second version supports multiple tests and changes the way to check the microphone and the real-time input display. The workflow of the system turns to be as drawn in figure 5.1. The user can make sure the microphone works by checking the input level bar, which reflects the signal power of the microphone input. After each subtest, the system checks whether there is any unfinished subtest. If yes, redirect the user to the break page. Else, jump to the complete page.

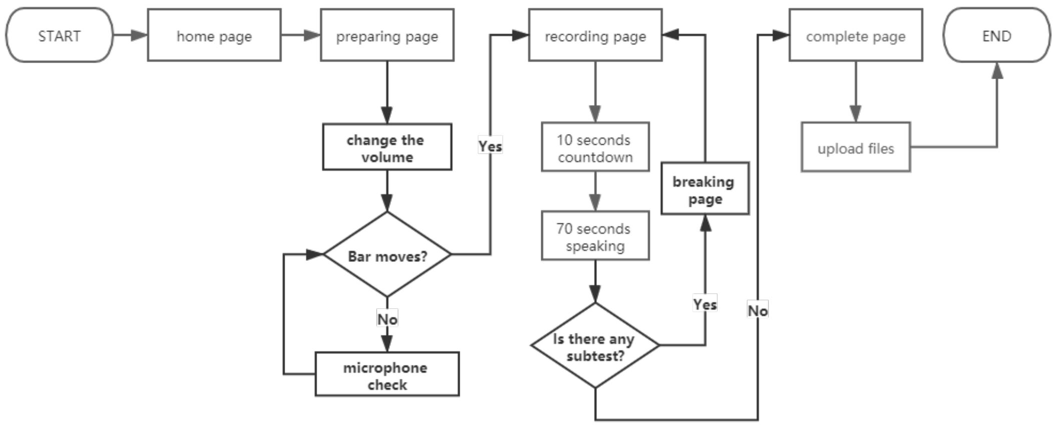


Figure 5.1: The workflow of the second version.

5.1 Implementation

5.1.1 Database design

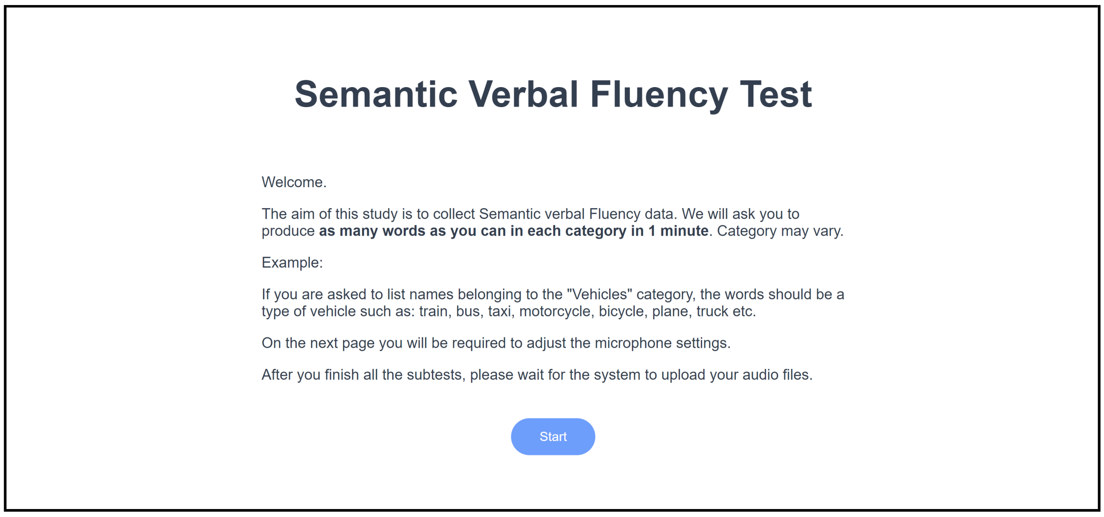
Due the support for multiple category test, another attribute, category, was added in the records table.

Attribute Name	Type	Length	Key
category	varchar	64	False

Table 5.1: The description of category attribute in records table.

5.1.2 Home page

In this version, according to the feedback of the first evaluation, the text on this page, as illustrated in figure 5.2, is more precise. It only gives a simple view of the semantic verbal fluency test and shortens the description of each page.



Semantic Verbal Fluency Test

Welcome.

The aim of this study is to collect Semantic verbal Fluency data. We will ask you to produce **as many words as you can in each category in 1 minute**. Category may vary.

Example:

If you are asked to list names belonging to the "Vehicles" category, the words should be a type of vehicle such as: train, bus, taxi, motorcycle, bicycle, plane, truck etc.

On the next page you will be required to adjust the microphone settings.

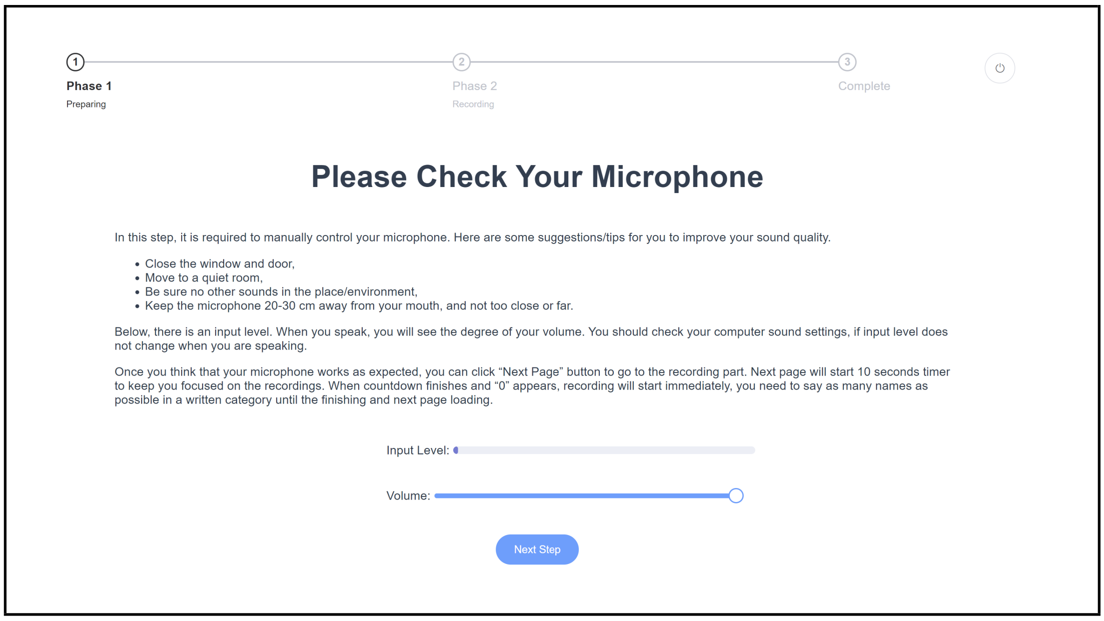
After you finish all the subtests, please wait for the system to upload your audio files.

Figure 5.2: The home page of the second version.

5.1.3 Preparing page

In this page (figure 5.3), users can see the input level of their microphone through a bar, the color gets red when the voice is too loud and becomes blue when the volume is too low. By checking the input level and using the slider, they can vary the volume of their recording.

Changing the microphone check into this way is more intuitive and straightforward for people unfamiliar with computer operations. No need to record themselves to avoid wasting too much time in this process.



Please Check Your Microphone

In this step, it is required to manually control your microphone. Here are some suggestions/tips for you to improve your sound quality.

- Close the window and door,
- Move to a quiet room,
- Be sure no other sounds in the place/environment,
- Keep the microphone 20-30 cm away from your mouth, and not too close or far.

Below, there is an input level. When you speak, you will see the degree of your volume. You should check your computer sound settings, if input level does not change when you are speaking.

Once you think that your microphone works as expected, you can click “Next Page” button to go to the recording part. Next page will start 10 seconds timer to keep you focused on the recordings. When countdown finishes and “0” appears, recording will start immediately, you need to say as many names as possible in a written category until the finishing and next page loading.

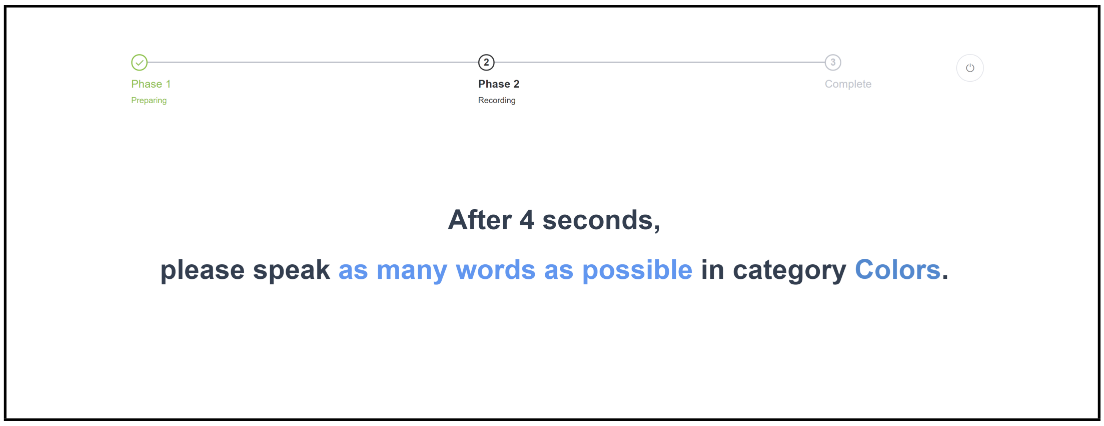
Figure 5.3: The preparing page of the second version.

5.1.4 Recording page

In this version, as illustrated in figure 5.4, following the feedback of the first evaluation, the instruction highlights ”as much as possible”, which can constantly remind them to try their best to speak. Besides, the recording volume is passed as a parameter to control the recording setting of this page.

In figure 5.5, the sonic animation during recording changed into the way as preparing page (section 5.1.3). This makes the interfaces becomes more consistent, which can reduce the memory load of users.

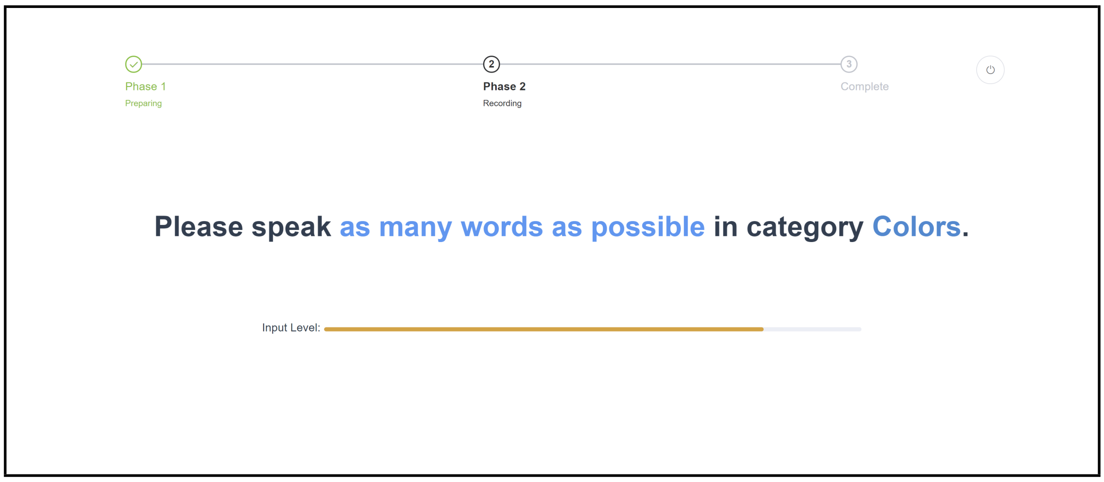
Due to multiple tests, only when the system judges that this is the first test will an id be generated. Besides, The file uploading process moves into this page. Once the recording stops, the WAV files of this test will be uploaded directly before jumping to another page.



After 4 seconds,

please speak as many words as possible in category Colors.

Figure 5.4: The recording page before start recording in the second version.



Please speak as many words as possible in category Colors.

Figure 5.5: The recording page during recording in the second version.

5.1.5 Break page

There is a break page between two subtests to let the participant take a breath and inform them how many subtests have finished and how many remains. When the participants get ready, they can click the button to continue the test.

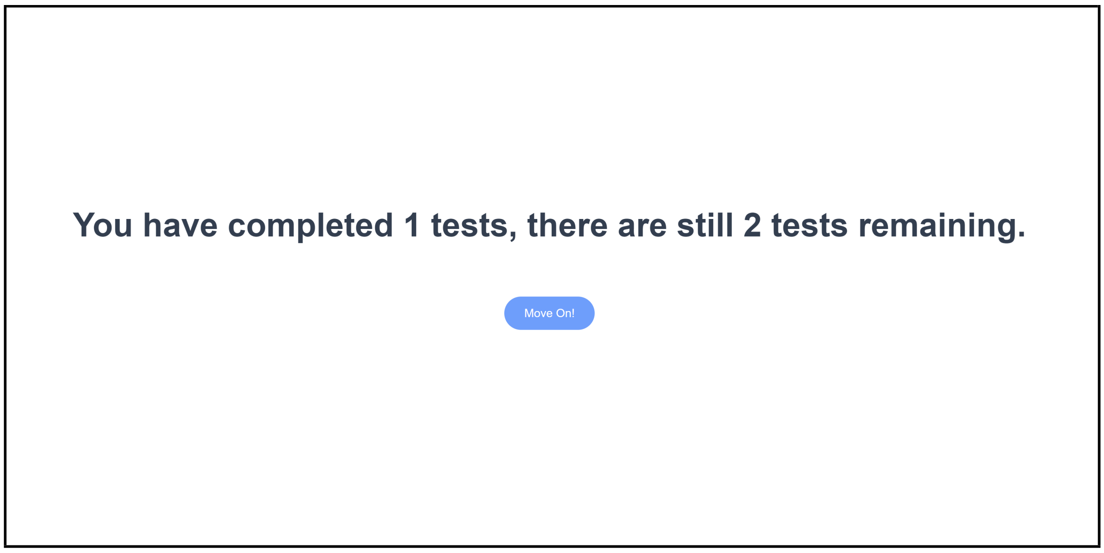

You have completed 1 tests, there are still 2 tests remaining.

Figure 5.6: The break page of the second version.

5.2 Evaluation

5.2.1 Aims

The user test in this stage was a concurrent test. In practice, there might be situations when multiple users enter the system at the same time or in a very short interval, so it is essential to ensure no blockage occurs.

5.2.2 Procedure

As discussed with Yasa Kostas Rabia (personal communication, July 2, 2021), there would not be so many people using the system simultaneously in the actual data collection, so three participants are enough. The participants used different devices to enter the system and started the three-subtests SVF tests concurrently. After the files were all submitted successfully, the test finished.

5.2.3 Results

Three tests all processed smoothly, and nine files were saved, which indicates the system can support at least three people to take the test simultaneously.

Chapter 6

Third iteration of the design

This version design is based on the suggestions from Professor Maria Wolters during the interviews (personal communication, July 7, 2021). Because this filtering method, to some extent, removes human voice, keeping the original file is an efficient way to avoid losing vital information. Filtering is an option if they need to remove noise, but there is no need for some research, so this version provides both files. The video to show how the system works is in the appendix A.

6.1 Implementation

The filter is implemented in the backend with the use of the SciPy signal module (section 3.5.1). In the front end, the audio context deletes the BiquadFilterNode, so the GainNode connects to the context destination directly, as shown in figure 6.1.

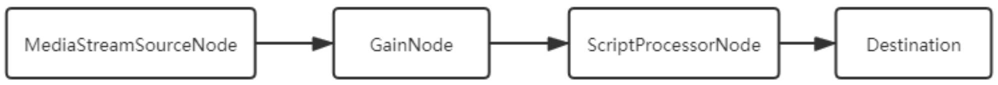


Figure 6.1: The graph for audio processing.

After generating a Blob containing sample data, the browser sends a POST HTTP request to the backend with the Blob object, the category name, and user ID as body parameters. Then, in the interface dealing with this request, firstly, the original data is filtered by the bandpass filter with passband 60-2000Hz, and then both filtered and original files are saved in Google Cloud Storage. The reason to use this range of frequency is that it is clear from the spectra of samples that most power is concentrated in this frequency band. So I removed frequencies with minor energy and listen to the filtered file to get the best frequency band that keeps human voice and removes most of the noise.

6.1.1 Database design

Because both filtered and original audio files are stored, a new attribute, filtered, was applied to help identify these two files. It is a Boolean value, 1 represents filtered file and 0 for another.

Attribute Name	Type	Length	Key
filtered	tinyint	1	False

Table 6.1: The description of filtered attribute in records table.

6.2 Evaluation

6.2.1 Aims

This evaluation assessed the final version of system performance on three aspects: instruction clarity, system error, countdown suitability, and further improvements.

6.2.2 Procedure

The system has two subtests: animals and fruits. Participants took tests, reported erratic performance, told how they felt about the instruction and the countdown, and suggested other improvements.

6.2.3 Results

The results reflect that no system error occurred during the test, but about the instructions, some participants thought the recording page and the break page lacked clear descriptions. Because there was no reminder on the recording page that recording ends automatically, some participants thought they had to stop recording themselves. In the aspect of the countdown suitability, most people thought the time is enough to read the instruction and get prepared to speak, but a small number of people thought it is inadequate and 30 seconds might be a better setting.

Chapter 7

Future works

7.1 Other filtering methods

In this project, the noise filter is a bandpass filter, but it results in a reduction in signal energy of both human voice and noise. Moreover, the background noise contains both stationary and nonstationary noise. The fixed passband filter can only remove stationary noise that is related to a narrow frequency band. However, this kind of filter fails to delete nonstationary noise because of its continually changed frequency[52]. In recent years, many speech enhancement studies show that deep learning technologies such as deep multi-layer perceptron, convolution neural networks, and deep neural networks are robust enough to segregate human sound from background noise[53]. Therefore, using deep learning technologies to remove noise might be a solution in future work.

7.2 Qualtrics connection

The demographic information is going to be stored in another platform, Qualtrics[54]. Google also provides survey services. If the recording files and demographic information are saved separately, in the worst situation when data stored on a platform is leaked, user information can still be partially protected.

In practice, users need to fill the form from Qualtrics before entering this system. However, so far, the connection between this system and Qualtrics is still unsolved. One way to deal with this problem is using the unique ID created by Qualtrics. Once the user fills a survey, Qualtrics will generate an ID for it. If the system can get the ID and use it directly, the demographic information in the Qualtrics can then be mapped on audio recordings. Another solution is the match between unique IDs created by these two systems because both Quatrics and this system generate a unique ID for one user, which is a connection between these two systems.

Chapter 8

Conclusions

With the constraints of the COVID-19 pandemic, traditional SVF tests need telehealth techniques to conduct as usual. In this work, I implemented a web application for data collection in semantic verbal fluency tests. Before starting web design, I looked through papers related to semantic verbal fluency tests and telemedical. It allows users to take SVF tests remotely, which helps telehealth and defuses the obstacles caused by COVID-19 and other emergencies. The design consisted of three iterations, and in each iteration, there was an evaluation to assess the performance of the system and give the direction of the next stage. The overall response was positive, and some people who work in the neuropsychology area believed this application would help data collection in their studies shortly.

Nevertheless, still, some places can be improved next. For example, apply deep learning techniques to create a more effective noise filter and connect Qualtrics and this system to match demographic information with speech.

Bibliography

[1] Alexandra König, Nicklas Linz, Johannes Tröger, Maria Wolters, Jan Alexandersson, and Phillipe Robert. Fully automatic speech-based analysis of the semantic verbal fluency task. *Dementia and Geriatric Cognitive Disorders*, 45(3-4), 2018.

[2] Esteban Vaucheret Paz, Celeste Puga, Christy Ekonen, Paula Pintos, Isabel Lascombes, Soledad De Vita, Mariana Leist, Mariela Corleto, and María José García Basalo. Verbal Fluency Test in Children with Neurodevelopmental Disorders. *Journal of Neurosciences in Rural Practice*, 11(1), 2020.

[3] Holly Jimison, Misha Pavel, and Thai Le. Home-based cognitive monitoring using embedded measures of verbal fluency in a computer word game. In *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS'08 - "Personalized Healthcare through Technology"*, 2008.

[4] Fl studio. `https://www.image-line.com/`. Online; accessed 09 May, 2021.

[5] Audio recording and editing software: Adobe audition. `https://www.adobe.com/uk/products/audition.html`. Online; accessed 09 May, 2021.

[6] Google voice. `https://voice.google.com/u/0/about`. Online; accessed 09 May, 2021.

[7] Anthony C. Smith, Emma Thomas, Centaine L. Snoswell, Helen Haydon, Ateev Mehrotra, Jane Clemensen, and Liam J. Caffery. Telehealth for global emergencies: Implications for coronavirus disease 2019 (COVID-19). *Journal of Telemedicine and Telecare*, 26(5), 2020.

[8] Elham Monaghesh and Alireza Hajizadeh. The role of telehealth during COVID-19 outbreak: A systematic review based on current evidence, 2020.

[9] A. Wright, J. Mihura, H. Pade, and D. McCord. Guidance on psychological tele-assessment during the COVID-19 crisis. *American Psychological Association*, 2020.

[10] Glenn Waller, Matthew Pugh, Sandra Mulkens, Elana Moore, Victoria A. Mountford, Jacqueline Carter, Amy Wicksteed, Aryel Maharaj, Tracey D. Wade, Lucene Wisniewski, Nicholas R. Farrell, Bronwyn Raykos, Susanne Jorgensen, Jane Evans, Jennifer J. Thomas, Ivana Osenk, Carolyn Paddock, Brittany Bohrer, Kristen Anderson, Hannah Turner, Tom Hildebrandt, Nikos Xanidis, and Vera Smit. Cognitive-behavioral therapy in the time of coronavirus: Clinician tips for working with eating disorders via telehealth when face-to-face meetings are not possible. *International Journal of Eating Disorders*, 53(7), 2020.

[11] Muriel D Lezak, D B Howieson, D W Loring, J H Hannay, and J S Fischer. Neuropsychological Assessment. Oxford University Press. *New York*, 2004.

[12] L. Olabarrieta-Landa, D. Rivera, L. Lara, S. Rute-Pérez, A. Rodríguez-Lorenzana, J. Galarza-Del-Angel, A. I. Peñalver Guia, R. Ferrer-Cascales, J. Velázquez-Cardoso, A. I. Campos Varillas, D. Ramos-Usuga, B. Chino-Vilca, M. A. Aguilar Uriarte, P. Martín-Lobo, C. García De La Cadena, B. Postigo-Alonso, I. Romero-García, B. V. Rabago Barajas, M. J. Irías Escher, and J. C. Arango-Lasprilla. Verbal fluency tests: Normative data for Spanish-speaking pediatric population. *NeuroRehabilitation*, 41(3), 2017.

[13] Angela K. Troyer, Morris Moscovitch, and Gordon Winocur. Clustering and switching as two components of verbal fluency: Evidence from younger and older healthy adults. *Neuropsychology*, 11(1), 1997.

[14] Claire Hughes, Marie Hélène Plumet, and Marion Leboyer. Towards a cognitive phenotype for autism: Increased prevalence of executive dysfunction and superior spatial span amongst siblings of children with autism. *Journal of Child Psychology and Psychiatry and Allied Disciplines*, 40(5), 1999.

[15] Ignacio Obeso, Enrique Casabona, Maria Luisa Bringas, Lázaro Álvarez, and Marjan Jahanshahi. Semantic and phonemic verbal fluency in Parkinson's disease: Influence of clinical and demographic variables. *Behavioural Neurology*, 25(2), 2012.

[16] Rowena G. Gomez and Desirée A. White. Using verbal fluency to detect very mild dementia of the Alzheimer type. *Archives of Clinical Neuropsychology*, 21(8), 2006.

[17] Abhijeet Patra, Arpita Bose, and Theodoros Marinis. Lexical and cognitive underpinnings of verbal fluency: Evidence from bengali-english bilingual Aphasia. *Behavioral Sciences*, 10(10), 2020.

[18] Johannes Tröger, Nicklas Linz, Alexandra König, Philippe Robert, and Jan Alexandersson. Telephone-based dementia screening i: Automated semantic verbal fluency assessment. In *ACM International Conference Proceeding Series*, 2018.

[19] Serguei V.S. Pakhomov, Wrenda Teeple, Anne M. Mills, and Michael Kotlyar. Use of an automated mobile application to assess effects of nicotine withdrawal on verbal fluency: A pilot study. *Experimental and Clinical Psychopharmacology*, 24(5), 2016.

[20] Liu Chen, Meysam Asgari, Robert Gale, Katherine Wild, Hiroko Dodge, and Jeffrey Kaye. Improving the Assessment of Mild Cognitive Impairment in Advanced Age With a Novel Multi-Feature Automated Speech and Language Analysis of Verbal Fluency. *Frontiers in Psychology*, 11, 2020.

[21] International Standard. Ergonomics of human–system interaction — Part 210: Human-centred design for interactive systems. Technical Report 1, 1994.

[22] Martin Maguire. Methods to support human-centred design. *International Journal of Human Computer Studies*, 55(4), 2001.

[23] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Conference on Human Factors in Computing Systems - Proceedings*, 1990.

[24] Why django? `https://www.djangoproject.com/`. Online; accessed 29 April, 2021.

[25] Object–relational mapping. `https://en.wikipedia.org/wiki/Object%E2%80%93relational_mapping`. Online; accessed 16 August, 2021.

[26] Loose coupling. `https://en.wikipedia.org/wiki/Loose_coupling`. Online; accessed 16 August, 2021.

[27] Introduction to django. `https://www.runoob.com/django/django-intro.html`. Online; accessed 16 August, 2021.

[28] What is vue.js? `https://v3.vuejs.org/guide/introduction.html`. Online; accessed 29 April, 2021.

[29] Audiocontext. `https://developer.mozilla.org/en-US/docs/Web/API/AudioContext`. Online; accessed 09 May, 2021.

[30] Mediastreamaudiosourcenode. `https://developer.mozilla.org/en-US/docs/Web/API/MediaStreamAudioSourceNode`. Online; accessed 16 August, 2021.

[31] Webrtc api. `https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API`. Online; accessed 16 August, 2021.

[32] Gainnode. `https://developer.mozilla.org/en-US/docs/Web/API/GainNode`. Online; accessed 16 August, 2021.

[33] Biquadfilternode. `https://developer.mozilla.org/en-US/docs/Web/API/BiquadFilterNode`. Online; accessed 16 August, 2021.

[34] Q factor. `https://en.wikipedia.org/wiki/Q_factor`. Online; accessed 16 August, 2021.

[35] Scriptprocessornode. `https://developer.mozilla.org/en-US/docs/Web/API/ScriptProcessorNode`. Online; accessed 16 August, 2021.

[36] Signal processing (scipy.signal). `https://docs.scipy.org/doc/scipy/reference/signal.html`. Online; accessed 16 August, 2021.

[37] scipy.signal.butter. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.butter.html#scipy.signal.butter`. Online; accessed 16 August, 2021.

[38] scipy.signal.sosfilt. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.sosfilt.html#scipy.signal.sosfilt`. Online; accessed 16 August, 2021.

[39] Stephen Butterworth. On the theory of filter amplifiers, 1930.

[40] uuid — uuid objects according to rfc 4122. https://docs.python.org/3/library/uuid.html. Online; accessed 16 August, 2021.

[41] Brian B. Monson, Eric J. Hunter, Andrew J. Lotto, and Brad H. Story. The perceptual significance of high-frequency energy in the human voice. *Frontiers in Psychology*, 5(JUN), 2014.

[42] Google cloud platform (gcp) - get started on google cloud. https://cloud.google.com/. Online; accessed 16 August, 2021.

[43] Cloud storage. https://cloud.google.com/storage/. Online; accessed 16 August, 2021.

[44] Object storage. https://en.wikipedia.org/wiki/Object_storage. Online; accessed 19 August, 2021.

[45] Scott Fulton III. What is google cloud is and why would you choose it? https://www.zdnet.com/article/what-is-google-cloud-is-and-why-would-you-choose-it/. Online; accessed 16 August, 2021.

[46] App engine. https://cloud.google.com/appengine. Online; accessed 16 August, 2021.

[47] Google app enginecomparison —standard vs flexible environment. https://medium.com/@venkat86.careers/how-to-choose-app-engine-environment-standard-flexible-9f4c26a723b0. Online; accessed 19 August, 2021.

[48] Cloud sql. https://cloud.google.com/sql/. Online; accessed 16 August, 2021.

[49] Blob. https://developer.mozilla.org/en-US/docs/Web/API/Blob. Online; accessed 16 August, 2021.

[50] Riff. https://en.wikipedia.org/wiki/Riff. Online; accessed 16 August, 2021.

[51] Mime types (iana media types). https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types. Online; accessed 16 August, 2021.

[52] Joseph Bamidele Awotunde, Roseline Oluwaseun Ogundokun, Femi Emmanuel Ayo, and Opeyemi Emmanuel Matiluko. Speech segregation in background noise based on deep learning. *IEEE Access*, 8, 2020.

[53] Soha A. Nossier, Julie Wall, Mansour Moniri, Cornelius Glackin, and Nigel Cannings. An experimental analysis of deep learning architectures for supervised speech enhancement. *Electronics (Switzerland)*, 10(1), 2021.

[54] Qualtrics xm: The leading experience management software. `https://www.qualtrics.com/`. Online; accessed 09 May, 2021.

Appendix A

Final version presentation video

https://drive.google.com/file/d/147Go4MjfowdkgysBpxxlnssxqjfzpBP-/view?usp=sharing