

Peer Testing Report II



Produced by:
The Speaking
Portal Project
- Team A

Edouard Eltherington
Matt Kuelker
Logan Parker
Veronica Jack



System Description

The Speaking Portal Project is an animation-generating add-on for the Kukarella text-to-speech (TTS) system. Our system is built as an API which is able to generate and return an MP4 animation based on the Kukarella-provided text and audio files. For this round of peer testing, our system had its full set of features implemented, so our program:

- ☑ Receives the Kukarella TTS audio file, text file, language selection (compatible with all languages available on the Kukarella website), and avatar selection as input
- ☑ Animates the selected avatar to have lip and mouth movements synchronized to the audio and text (compatible with all available languages on the Kukarella website)
- ☑ Animates the selected avatar to blink naturally and change body poses regularly
- ☑ Animates the selected avatar using the assets provided by Kukarella's artist
- ☑ Generates an MP4 video with the avatar animated at 24-fps, and then return it within a reasonable amount of time. See below for the current processing time.
 - on Localhost
 - a 30-second English audio file is processed within 22 seconds
 - a 30-second non-English audio file is processed within 10 seconds
 - on an AWS server instance
 - a 30-second English audio file is processed within 27 seconds
 - a 30-second non-English audio file is processed within 25 seconds
- ☑ Is packaged as an API so that it is portable and can be easily integrated into multiple operating systems (Linux, Windows, and MacOS)
- ☑ (Bonus Feature) animates the selected avatar to be compatible and look natural with the available voice effects on the Kukarella web application, including pitch, pause, speed, say as, emphasis, volume, mute, breath, and whisper

Participants

A total of 9 participants completed peer evaluations for the Speaking Portal Project. Three of the evaluations were completed in remote sessions with Veronica Jack and the rest were conducted in person. Please note that all evaluations were `User Feedback` scenarios. Without a user interface, we believed that User Feedback Evaluations would be most appropriate for our project.

Participant Name	Administrator	Type of Evaluation	Status
Yash Atreya	Edouard	User Feedback	Complete
Ryder	Veronica	User Feedback	Completed remotely
Brenden Trieu	Logan	User Feedback	Complete
Abhiek	Veronica	User Feedback	Completed remotely
Aidan	Matt	User Feedback	Complete
Thom	Matt	User Feedback	Complete
Ivan	Logan	User Feedback	Complete
Mohammed	Veronica	User Feedback	Completed remotely
Akshat	Edouard	User Feedback	Complete

User Groups and Task Lists

User Group 1: User

A user is a member of the general public who is using the Kukarella web application; they are someone who would like to have a more immersive text-to-speech experience where an animated avatar reads their text aloud. In their pursuit of having an animated text-to-speech experience, they use a premium Kukarella account to have access to all Kukarella features.

User Task 1: English Baseline

This first task helps the user get a feel for the system by asking them to use the Kukarella web application. The user is asked to enter text in English, select a voice, convert the input to a .wav and .txt file, and inform the administrator which avatar they'd like to use: Barb or Boy. The administrator sends the input to the API to simulate an integrated API, and then the user watches the video and provides feedback. The feedback is related to the time it took for the video to be generated and the quality of the realism features, including the lip synchronization, the blinking, and the rate of change for body poses. See the quantitative score section below for more information regarding the feedback request.

User Task 2: Language Test

The second task asks the user to enter text in a language other than English. It is preferable for the user to be familiar with the language to ensure the lip synchronization is accurate for non-English languages; however, it is not necessary for them to be familiar with the language, and the user may use an online translation service to assist with this task. Again, the user is asked to select a voice, convert the input to a .wav and .txt file, and inform the administrator which avatar they'd like to use: Barb or Boy. The administrator sends the input to the API, and then the user watches the video and provides similar feedback to task 1.

User Task 3: Voice Test

The goal of the third task is for the user to select another voice from the Kukarella application to ensure that the avatar's animation looks natural no matter which voice is selected. The user is asked to enter text in a language they're familiar with, select a different voice from previous tasks, convert the input, and inform the examiner of their avatar selection. After the examiner sends the input to the API, the user watches the video and provides similar feedback to task 1.

User Task 4: Voice Effects

The Kukarella system has voice effects available for premium users, so this task is a bonus feature to see if the API works appropriately with the available voice effects, including pitch, pause, speed, emphasis, volume, breath, and whisper. In this case, the user is asked to enter text in a language they're familiar with, and then select a voice that supports voice effects (some voices only support a few effects, and some do not support any). Again, the user converts the input, informs the examiner of their avatar selection, and provides feedback after watching the video.

User Task 5: Mix and Match

This final user task allows the user to try each available option in the Kukarella web application to see if there are any combinations that result in an unsatisfactory animation. The user is asked to enter text in any language, select any voice, add any voice effects, and then convert the input for the administrator to send to the API. The user then watches the video and provides feedback.

User Group 2: Developer

The developer is a Kukarella employee who is responsible for incorporating the API into the Kukarella system and adding more avatars in the future.




Developer Task 1: API Integration











The developer is asked to read the README to ensure that the API integration instructions are clear. The developer is asked about the parameters sent to the API to ensure they understand which inputs are needed even if the API is not integrated.

Developer Task 2: Character Assets

The developer is asked to look at the source code directories to identify where the character image assets are saved. The peer testing administrator then asks about the image file naming scheme to see if the developer understands the purpose of the naming scheme and if they could add a new character if images were provided.

Issues Discovered

The following is a list of issues identified in order of priority from  high to  medium to  low, along with their statuses.

Priority	Issue	Status
	Blinking Probability allows for the character to not blink during shorter clips	In Progress
	Avatar selection error handling allowed for invalid character selection	Completed
	Documentation for server-side AWS implementation	In Progress
	Additional blinking (higher probability) to make it appear more natural (not just for short clips)	In progress
	Documentation: links to additional tooling, links to error codes	In progress
	Documentation: instructions to increase the frames per second, to add transition frames for mouth movements, for the image assets' naming scheme	In progress
	The body poses were described as “awkward” and “strange.” This is a low priority because the official poses were not yet provided by the artist and the current poses are temporary to show that pose changes are possible.	Backlog
	Pose changes were “sudden” and participants requested transition poses. This is a lower-priority issue because the transition poses require the official art from the artist so that the transitions are appropriate.	Backlog
	Documentation: image asset directory location	Backlog
	The whisper voice effect resulted in no lip synchronization in one instance. This is considered a low priority because compatibility with voice effects is a bonus feature.	Backlog

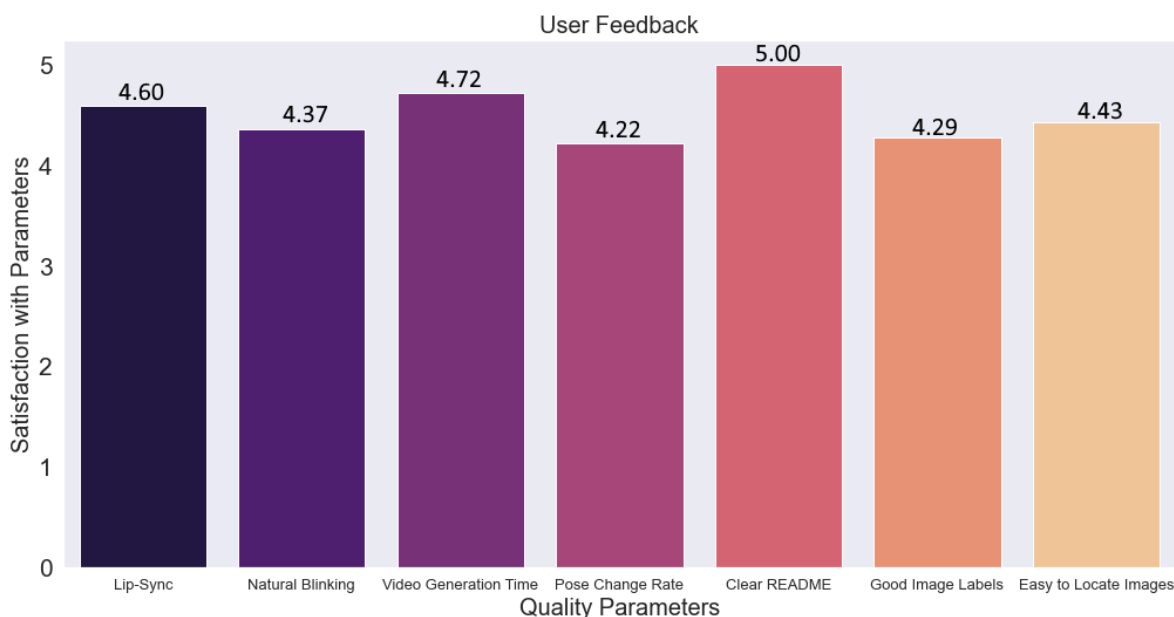
Quantitative Scores

Due to the nature of our project, the majority of the usability heuristics listed in the course do not apply to our API, as the project lacks a user interface. Therefore, we chose alternative quality parameters that were valued by our client. Before explaining our user feedback and graph, we will first address the 10 heuristics:

1. **Visibility of System Status:** The system is built as an API, informs the developer when it is listening for requests, and returns all appropriate HTTP response codes.
2. **Match between system and the real world:** Represented as animation realism through lip-synchronization, natural blinking, and pose change rate.
3. **User Control and Freedom:** Users can choose avatar types and languages. This is modelled under the Lip-Sync quality parameter as any Kuckerella-generated audio will work with the API.
4. **Consistency and Standards:** Represented by lip-sync, video generation time, and clear readme
5. **Error Prevention:** The system returns HTTP error codes as well as embedded HTML responses notifying users if parameters are not specified or incorrect.
6. **Recognition rather than recall:** Not Applicable
7. **Flexibility and efficiency of use:** Not applicable, but is somewhat related to the Video Generation Time, as this API takes only four parameters.
8. **Aesthetic and minimalist design:** Not Applicable
9. **Help users recognize, diagnose, and recover from errors:** HTTP requests and error HTML will be returned, however since this program is built as an API, troubleshooting will be done by developers instead of users, so this is not easily applicable.
10. **Help and Documentation:** Modeled as Clear Readme

The scores we received as feedback during the peer testing session are on a 5-point scale and participants were asked to indicate their level of agreeableness (strongly disagree to strongly agree) to the following:

1. The avatar's lips are synchronized to the words being said. In other words, the mouth movements appear natural for the sounds being produced.
2. The avatar's blinking appears natural.
3. The video was generated within a reasonable amount of time.
4. The avatar changes body poses at a reasonable rate.
5. The README is clear and the parameters sent to the API are evident.
6. The avatar asset naming scheme makes sense and provides adequate information about the images.
7. The avatar assets are easily found in the source code directories.



The first score is 4.60 out of 5 for lip synchronization quality. Positive feedback included that “English syncs quite well” and “sync was pretty realistic to how irl people would speak”. However, there were some criticisms, including that the “word cool at the end didn’t seem as natural” and that it “seemed a little robotic but it’s hard to tell.” We are satisfied with this result, but we know that the quality could be improved if we increase the frames per second and if we add transition mouth images as well. The implementation would require additional art from Kukarella’s artist, and we were recently informed that we would no longer be provided with art for this project. Therefore, to address this issue, we can provide instructions in the documentation so that developers will understand where and how to implement the additional art.

The second score is 4.37 out of 5 for the natural or realistic quality of the blinking. We received compliments including, “I did not notice the blinking, it must have been natural” and “she blinked and that alone made it more realistic.” One participant simply commented, “It blinks!” The recommendations for improvement mentioned having the character blink more often, as the comments included “Maybe 1 more blink but it’s great” and “Animation is good, timing could be improved.” There was also one participant who had a short video and indicated that the character did not blink at all. Therefore, we created this as a high-priority issue to guarantee at least one blink for short videos (under 10 seconds) and to slightly increase the blinking probability to ensure a more natural appearance.

The third score is 4.72 out of 5 for the time it took the API to generate and return the video file. We are very satisfied with this score, as we have been putting a lot of effort

into ensuring the timing performance of the API. We will continue to look into how to improve the timing, as the client indicated that it is a high priority for them; however, we did not receive any comments or recommendations for improvement from our participants.

The fourth score is 4.22 out of 5 for the rate at which the avatar changes poses. We found that many participants reacted and rated the quality of the poses rather than the rate at which the avatar changes poses. We know that we have awkward temporary poses due to not having received the official poses from the Kukarella artist. Unfortunately, we were recently informed that we would no longer be provided with art for this project; therefore, we will try to make our temporary poses look better for the final presentation to ensure people are not distracted by the odd poses. As for the participants who responded to the rate, their comments included, “looks good, timing seems fluid” and “smaller changes in posture at current rate seems natural but big changes in posture might not”. We found one comment that summarized how we felt about this rating, which was, “the body poses are a bit strange and could definitely be smoother, but looks good overall!”

We identified two low-priority issues for avatar pose changes, which are to address the poses themselves and transition poses to improve realism. For the “awkward” and “strange” poses, we will leave this issue in our backlog in case the client does provide us with more art. For the transition poses, the issue will be to provide instructions in the documentation to indicate how and where the transition poses should be implemented in terms of the image assets and in the code.

The fifth score is 5.00 out of 5 for the clarity of the README. We are very proud of this score, as we spent a lot of time writing the README to ensure it is easy to understand and addresses all aspects of the project. We, unfortunately, forgot to link to our additional documentation related to our tooling, but the participants’ comments were all positive: “Easy to read, simple” and “Great README. Very clear and well written.” Even though we had a perfect score, participants had excellent suggestions, including the addition of a “link to common error code list” and “maybe add a few steps on deployment to a server.” These suggestions were included in our list of issues.

The sixth score is 4.29 out of 5 for the intelligibility of the image asset naming scheme. Although this score is still high (above 4 out of 5), it is one of our lowest scores. We knew that we were taking a risk by not including an explanation of the naming scheme in our documentation. After this round of peer testing, we can see that an explanation is advisable. One comment specifically asked us to “add possible description regarding different asset options” so that it’s clearer how to add new image assets in the future. Otherwise, the feedback was positive, indicating that they understood the purpose of the image file names. When asked what information they perceived from the naming scheme,

participants replied with “combinations of various poses, mouth and blinks” and “they tell me which frame and body movement - whether it’s blinking or anything else, and the alphabet order makes sense.” This last comment was a little troubling because this tells us that it’s not clear to the developer that the alphabet is, in fact, a code for mouth positions and not a mouth position synchronized to that letter. Altogether, this issue is a medium priority.

The seventh score is 4.43 out of 5 for the ease at which participants found the avatar image assets. Most participants commented that the directory was “very easy to find” and that they “found it in a second.” There was one participant that didn’t realize that the animation was produced from a pool of images, so they were looking for the animation factory, which was mentioned in the README. We may add to the README to clarify where the images are located if we have time, but the rest of our participants found the directory easily.