Requirements Report

Speaking Portal Project – Team B

Team Members:

- Sarvagya Pandey – QA Lead
- Mawanli Cui – Tech Lead
- Cooper Smith – Client Liaison
- Yash Atreya – Project Manager

1. Software Description

The Speaking Portal project acts as an add-on to Kukarella's existing web platform to pair with the company's text-to-speech (TTS) software solution. It involves the generation of a 2D animated video with the help of user-input text, the speech file generated by Kukarella's TTS, and an avatar selected from a list of available models. It also provides the user an video embed code/link and the option to download the generated video file. The focus of the project is mostly on the back-end; as of now, the client does not require a front-end to be developed.

2. Target User Groups

- **Casual Users/Enthusiasts -** Users who are interested in the animated video generation feature and want to try it out.
- **Professional Users/Companies** - Users working in various fields such as education and marketing, who wish to integrate the animated video into their lectures, meetings, presentations, etc. in order to enhance their audience's experience.
- **Kukarella Developer Team** - Responsible for integrating the software with their existing application and making any desired changes

3. Software Requirements

3.1. Functional Requirements

- Receive user-input text through existing Kukarella TTS implementation via API request.
- Receive 2D avatar image chosen by the user from a list of options via API request.
- Receive speech file generated by Kukarella TTS implementation via API request.
- Create animated 2D video based on user-selected avatar, text and speech file.
- Store animated video in database.
- Provide an embed link for the generated video.
- Allow user to download the generated video.

3.2. Non-Functional Requirements

- **Performance** - Low buffer times, fast generation of animated video.
- **Code Quality** - Code must be professionally formatted, linted and maintained.
- **Robustness** - Ensuring that there are no failures through testing.
- **Efficiency** - Maximum utilization of resources for fastest response times.
- **Accuracy** - Accurate/realistic lip-syncing of animated video avatar to TTS audio.

3.3. User Requirements

Through the Kukarella App:

- Users are able to enter the text that they want to convert either through typing or by uploading a text file.
- Users select a 2D avatar (from a list of available avatars) that they want the animation to be created with.
- Users can view a list containing all available languages for the audio and select their desired language.
- Users can view a list containing all available voices for the selected langauge audio.
- Users can play a test/sample for each voice and make a selection based on their preferences.
- Users can request the video to be generated by clicking on a confirmation button.
- Users can view the generated video.
- Users can copy the embed code/link for the generated video.
- Users can download the generated video.

4. Constraints and Risks

- **Inexperience with 2D animation** - lack of knowledge with regards to animation process or libraries or frameworks that help with animation. This may lead to animation quality not meeting client's standards.
- **No support for custom avatars** - Animation capabilities limited to existing avatars provided by client.
- **Project time constraints** - Limited time to complete the project, team members need to focus on other courses every week.

5. Choice of Tech Stack

5.1. Back End:

Languages

- **Javascript** - Used to implement performant web servers and API endpoints.
- **Python** - Used to write scripts for video generation from the still image, text and audio files.

Frameworks

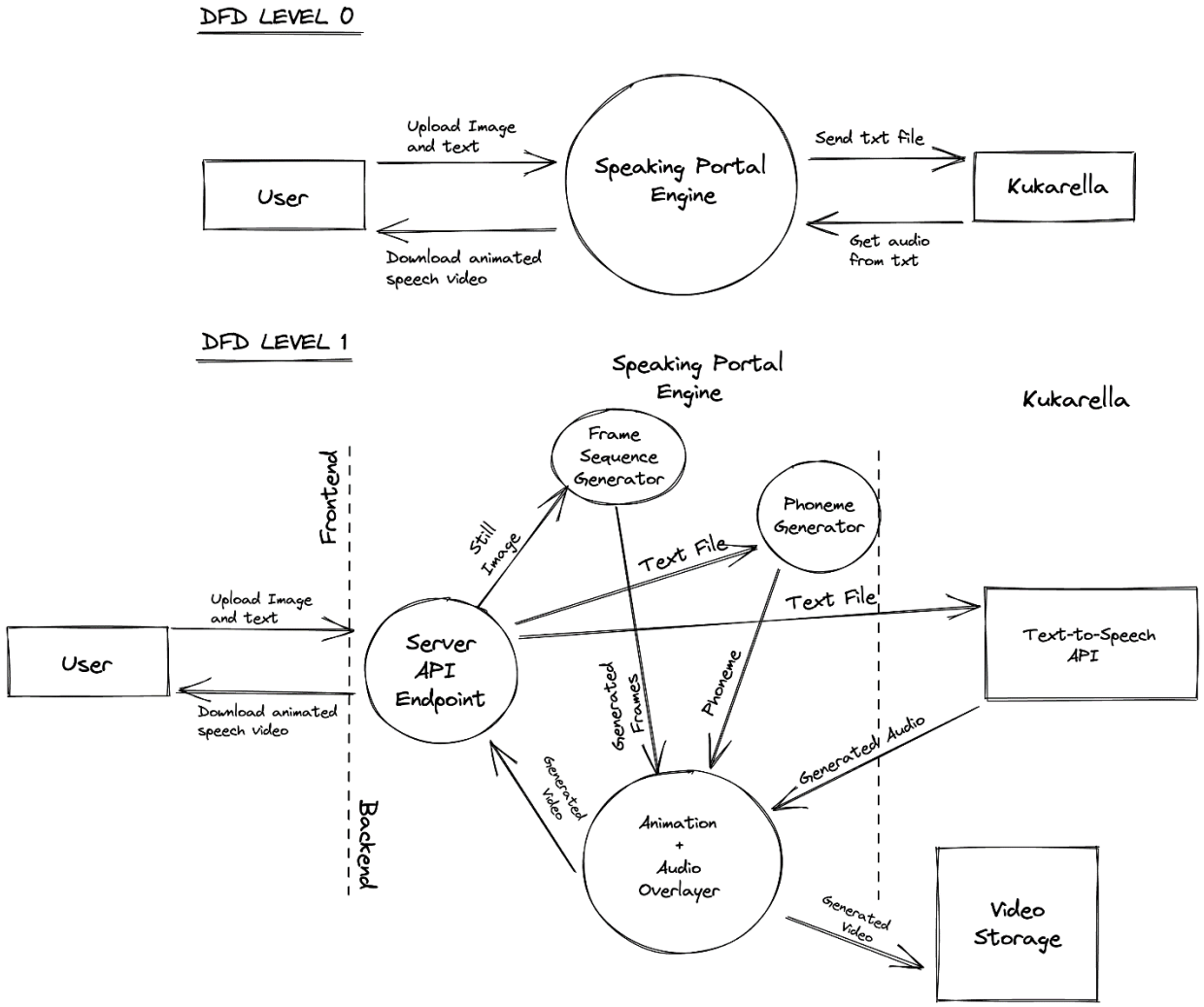
- **NodeJS and Express** - Runtime environment and framework for creating API endpoints using Javascript. Kukarella stack also uses NodeJS and Express which enables seamless integration.

- **Compute Engine** - Compute instances from Google cloud used to run heavy operations of animating still images, create frame, overlay audio to create a video. If necessary, replacement for this is EC2 instances from AWS.

Tools

- **Google Cloud Functions** - Used to create serverless functions for the backend. It is a serverless framework which enables us to create functions without having to worry about the underlying infrastructure. It enables infinite scalability and 99.9% uptime. If necessary, replacement for this is AWS Lambda.
- **Google Cloud Storage** - Used to store images and videos. It is a scalable and secure storage solution from Google Cloud. If necessary, replacement for this is AWS S3.

6. Data-Flow Diagrams



7. Project Milestones

- **Milestone 1 (Requirements Report)** - Functional, non-functional and user requirements; Data-flow diagrams (level 0 and 1); Choice of tech stack; Testing strategies (Due Oct. 21)
- **Milestone 2 (Peer Testing I)** - Program receives selected avatar, text and speech file input through Kukarella API; Program generates phoneme sequence and resulting viseme sequence; Program aligns viseme sequence with audio file; Program combines viseme sequence to generate video; Program allows for download of generated video; Program has simple interface/GUI that allows for peer testing (Due Nov. 25)
- **Milestone 3 (Peer Testing II)** - Incorporate feedback from Peer Testing I; API endpoints to access program's features; Support for custom/user-uploaded avatar animation if possible (Due Mar. 3)
- **Milestone 4 (Final Product)** - Incorporate feedback from Peer Testing II; Implementation of bonus features such as subtitles, additional facial animation, etc.; Finish documentation (Due Apr. 21)

8. Testing Strategy

- **Unit Testing** - individual features will be tested using PyTest.
- **Regression Testing** - Running all tests after a Pull Request to the main branch is created to ensure all previously written tests are passed even with the new code being implemented. This will be achieved with the help of GitHub Actions workflows.
- **Integration Testing** - Ensuring all components of the software work together by making calls to the API and comparing results with expected outputs.

- **Usability Testing** - Involves target user groups' feedback on product prototypes to ensure software is working as intended and to gather their impressions on features such as satisfaction with animation quality, video generation speed, etc.

9. Group Evaluation Questions

Q: Do professional users require a login?

A: A professional account on Kukarella will be required to use the video generation feature.

Q: Is the generated video going to go to a template page for viewing then deleted a certain amount of time after the generation of the page?

A: The video is going to be stored in a database, but we are currently unsure of how long it will be stored for. We will clarify with the client in our future meetings!

Q: Will you need help from professional artists/animators in order to create the code for animations? If yes, how do you plan on contacting such people?

A: We will require 2D assets previously made by Kukarella's animators. Should the need arise, we will be requesting the animators to create additional assets such as different mouth shapes for all the phonemes, etc.

Q: What metrics will you use to check the accuracy of the syncing? Do you have a definition for "seamlessly flowing audio"?

A: If phonemes have an 80% rate of being recognized successfully, this is within the margin

of error for seamless audio. We will also make sure that the testers of our application find the animation smooth and “in-sync” during the Usability Testing phase sometime during Term 2.

Q: How do you plan on implementing the download function for users? Will you use a video hosting service?

A: Generated videos will be stored in object storage on AWS or Google Compute, and we’ll expose a download URL from there.

Q: What practices will you be following in order to provide professionally formatted code?

A: All of us use VSCode to write code, which has extremely good linters like Flake8, etc that will help us check for simple logical as well as stylistic errors.. Other than that, we will be following the practice of decorators and comments to explain the code we are writing. All code will need to be reviewed by at least 2 other team members before it is merged to the main branch of the GitHub repository. This will help ensure that the quality is acceptable by all members of the team.

Q: Are you concerned with Python's relatively poor performance relative to other languages?

A: Python has poor performance relative to Javascript only for web servers, for which we will use NodeJS. Python is better at processing the tech for phoneme detection and generating the resulting animated video.

Q: How do you plan on testing components working together?

A: The best way to test components working together is to use a combination of unit and integration tests. Unit tests can be used to test the basic functionality of each component in

isolation. Integration tests can be used to test how different components work together to produce the desired output.

Q: How do you plan to ‘maximize efficiency’ for fast response times?

A: For the front-end, we will be using NextJS which has server-side rendering one of the best load times in the industry. Our web-server will be deployed using Google Cloud functions which are almost infinitely scalable low latency endpoints. Our compute for video generation we will be using cloud compute with GPUs.

Q: Are you concerned about the computing speed python provides (given its not exactly known for being performant)?

A: Python has poor performance relative to Javascript only for web servers, for which we will use NodeJS. Python is better at processing video and various ML models.

Q: What is the definition of "fast generation?" Why did you choose this timeframe?

A: This is a poor choice of words. A better timeline would be time/data = gen time

Q: Are you animating this yourself? The presentation made it seem like you were.

A: We are going to get some assistance from our clients in terms of borrowing already existing 2D assets that they have on hand, but we will be animating by ourselves.

Q: Do you have an idea of what python libraries you’ll be using?

A: We will be using Gentle for phoneme generation. We plan to use ffmpeg to generate the video.

Q: The current Kukarella product supports a very wide range of languages and accents, will your final product also be able to support all the currently available options or will it be limited?

A: Our primary focus would be to ensure that most popular languages (eg. - English, French, etc.) are supported. We are hoping to eventually support all languages that Kukarella does. Support relies on forced-alignment tools that may or may not work with niche languages and mouth sounds.

Q: In your presentation, you mentioned that your product will have fast response times, how do you plan to ensure that? What is the criteria that defines it as “fast”?

A: Fast response times is a poor choice of words. A better criteria would be a certain amount of data processed per unit of time.

Q: It was mentioned that the user can access the generated video via a link, does the embed code/link have an expiration? Meaning, can the user access the video via the link even a year after the video was generated?

A: Link will have a period of expiration. This ensures resources are used efficiently.

Q: How do you plan to use NLP with Python for your project?

A: The NLP part is used for text-to-speech generation, which is handled by Kukarella.

Q: You guys did well by making further comments on some of the points to justify/explain the reasoning behind it.

A: Thanks for the feedback!

Q: Where will the available voice audios be obtained from?

A: The audio files are taken straight from Kukarella's output from text.

Q: Are there any format requirements for the image to be uploaded or will all image types work?

A: TBD, but we'll stick with JPG and PNG mostly.

Q: Could talk more efficiently as the presentation went overtime greatly.

A: Will keep that in mind

Q: Slides were beautiful and materials were explained clearly

A: Thanks

Q: You mentioned that your project would be supporting multiple languages. Which languages will be supported and will that make your workload beyond the scope?

A: The languages that we'll use are Javacript for frontend and web-server, Python for video generation.

Q: What technology constraints are you encountering or you might encounter in the future?

A: Forced alignment of audio files in unsupported languages may or may not result in poor lip sync. There might be framerate-dependant TBD

Q: With the front-end, back-end mentioned, your team has decided to use python as the main language, but will there be other toolkits/frameworks used on top?

A: Yes, for frontend we will be using NextJS, and any CSS framework such as Tailwind or Bootstrap. NodeJS for our webserver and exposing our API endpoints.

Q: How are different error cases handled by the backend? Is there a detection methodology?

A: We haven't decided on a specific methodology but we'll write good error messages so as to not create any ambiguity. We also use service like Sentry and cloud logs to trace errors and what made them occur.

Q: What size of audio file is supported?

A: No hard limits yet, but we'll discuss this with the client. Keeping it less than 1GB is ideal as it will be handled smoothly by the web server and won't be too much for the users internet connection.

Q: What modeling software will you be using?

A: Since we are not making 3D animations, we will not be using any modeling software.

Q: It's great to list down the constraints and risks, but it would be better to have a backup plan for the risks.

A: Good point. Noted

Q: DFD level 0: Better to simplify level 0 DFD to reduce tech nouns in it.

A: Noted

Q: How do you define "good performance"? Any specific number?

A: Good performance is a poor choice of words. A better criteria would be a certain amount of data processed per unit of time.

Q: TypeScript is a new language and always comes up with new updates. Are you intending to update the language in the middle of development? If so, how do you maintain version controls?

A: We are not planning on using TypeScript for this project.

Q: How are you planning on creating an embedded link? Is the video expected to be hosted after generation on the Kukarella database?

A: TBD. Video will be retrieved in a format in which the client approves of.

Q: As an add-on, is the list of available languages and voices for you to create, or is that already in the Kukarella web app?

A: The list of available languages and voices will be provided by our client.

Q: How do you plan to manage implementation of both 2d and 3d at once?

A: We have decided to only stick to 2D animation to make the project more manageable.

Q: How do you plan to animate both uploaded images and the avatars by Kukarella?

A: At the time, it did not seem feasible to generate an animation from an uploaded image. We will focus on animating existing 2D avatars provided by our client first, and if time permits, we will work on animating user-uploaded images as a bonus feature.

Q: Nice work on the functional requirements, good job outlining the details.

A: Thanks for the comment.

Q: Target user groups should be more specific.

A: Additional target groups have been added.

Q: It would be nice to see some more design for the slides.

A: Noted

Q: How does the animation get made, who is making it?

A: We will be making the animation by generating a sequence of phonemes with timestamps, and then sync this sequence with the audio output we receive from Kukarella. We will then assign an image to each item of this sequence and merge them together to make an animated video.

Q: Does your tech stack have any requirements or external costs?

A: Mostly cloud costs for running compute and storage.

Q: Tech stack should probably have more than just the language.

A: Yes, we have made the changes in our report

Q: Do you plan on any future features?

A: Time permitting we can add subtitles, additional animation features such as full body animation, possibly program in the form of a google extension. Emotion or infliction on certain words that appear in animation.

Q: In relation to team communication, how do you plan to resolve or work around that issue?

A: We have resolved the team communication problems. We both agree that every one of our team members should show up to class.

Q: How will the workload be divided up in the case that the absent member is removed from the group, or refuses to contribute?

A: The group member that was absent has started contributing and actively participating in discussions. The work will be divided equally amongst the rest of the members if he is removed or leaves the group.