**User Identification Testing : Report**

Rocket Elevators wants to go the “extra mile” and to launch a new features for its elevators : a “Speaker Recognition” function that will enable the users to sign in with their profile ID with their voice. If not already registered in the system with a profile ID, the user can sign up (also vocally) with a sequence of steps that will be detailed below.

The present report will detail the inscription and recognition sequences a user must follow to use the system. It will also describe the tests and steps that had to be realized in order for the user profiles to be saved in the system and recognized.

The Microsoft Cognitive Services were used for that purpose. The system has to be taught which voice profiles to recognize as well as the vocal signatures related to it. The profiles have to be created following the cognitive services documentation (detailed on <https://westus.dev.cognitive.microsoft.com/docs/services/563309b6778daf02acc0a508/operations/5645c068e597ed22ec38f42e>).The instructions that were followed were in the “Identification Profile” and “Speaker Recognition” sections of the documentation.

**Methodology**

The main objectives of this part of the project are : creating three user profiles and be able to recognize them in various contexts (mainly conversations). The user profiles were created using the Microsoft Cognitive Services documentation methodology with the help of the Postman app allowing us to make the required API calls to create the user profiles and then verify their actual integrity (for example, are the right users correctly identified ?). All the vocal samples used for the profile creations were English or French discourses where the subjects recited song lyrics on a conversational tone for a duration of about 40 to 60 seconds. The ten conversations used for the speaker recognition part of this project were “elevator-related business conversations” in French and in English. Every test detailed below will be analyzed with the following criteria :

1. Main objective
2. Choice of implementation and parameters
3. Result for every file
4. Conclusions and recommendations if the experience was to be repeated

Every step was completed using the Microsoft Cognitive Services documentation that can be found here : <https://westus.dev.cognitive.microsoft.com/docs/services/563309b6778daf02acc0a508/operations/5645c068e597ed22ec38f42e>

**User profile creation**

1. *Main objective* : create three different user profiles, two English and one French. Raphaël Vallée et Maxime Patry are the English-speaking users. Rémi Lemay-Dupont is the French-speaking user.
2. *Choice of implementation and parameters* : we created the user profiles using the Postman app to call the following GET url with a POST function : https://westus.api.cognitive.microsoft.com/spid/v1.0/identificationProfiles

The following parameters had to be entered in the Postman application :  
-in the “Headers” section, the following key/values pairings : Content-type / application/json, Accept / application/json, Ocp-Apim-Subscription-Key / a9e3e21c91124fc3bb63a9e8ec9813a4 (related to the Azure Cloud account)  
-in the “Body” section, enter the following text : { “locale” : “en-us” }, select “raw” and “JSON” as a type (the “fr-fr” parameters were entered for user 3, a french-speaking user).

1. *Result for every file* : after pressing the “Send” button, the response gave back an “identificationProfileId” in the following form : { “identificationProfileId” : “49a36324-fc4b-4387-aa06-090cfbf0064f” } which will be needed for the subsequent steps.
2. *Conclusions and recommendations* : this being the first step (and a very simple one), and the documentation being crystal clear on this, it wasn’t hard to complete.

**Enrollment creation**

1. *Main objective* : associate each created profile with a voice do the speaker can later be identified. A 30 to 45 seconds voice sample was recorded to be sent and processed in the instructions below.
2. *Choice of implementation and parameters* : we enrolled the different user profile IDs using the following url with a POST function : https://westus.api.cognitive.microsoft.com/spid/v1.0/identificationProfiles/1bb58f2d-7db2-4b5d-8d58-7a5ea7aece58/enroll/

Here, the “1bb58f2d-7db2-4b5d-8d58-7a5ea7aece58” in the url was a profile ID created in the earlier step. The following parameters had to be entered in the Postman application :   
-in the “Headers” section, the following key/values pairings : Content-type / multipart/form-data, Ocp-Apim-Subscription-Key / a9e3e21c91124fc3bb63a9e8ec9813a4.  
-in the “Body” section, the recorded voice sample that we wanted to associate the profile ID to was sent as a binary file that we had to send along with the POST function. The voice samples were recorded using a Blue Yeti microphone as well as the Audacity software. The samples were song lyrics recited by the users on a conversational tone.

1. *Result for every file* : we ran into a few issues while transmitting the audio files. We firstly tried to send a too short (28 seconds) sample to the url, with an 404 error message returned to us telling us the sample was too short. Another try gave us another error message because the file had to be in mono. Another try returned us a 404 message saying the sample had to be in 16K format. We retried the B) steps with a 16K, mono and longer audio sample, which returned a 202 OK response. Therefore, we only ran into issues for user 1’s enrollment, because we recorded longer and appropriately formatted audio samples for the others (about 40 seconds long).
2. *Conclusions and recommendations* : audio files should be at least 35 seconds long to avoid recording it twice. The enrollment result will be verified with in the subsequent section : “Profile fetching”.

**Profile fetching**

1. *Main objective* : see if the profile creation and enrollment procedures now allow us to “GET” a profile and see its properties.
2. *Choice of implementation and parameters* : we fetched the user profiles using the following url with a GET function : https://westus.api.cognitive.microsoft.com/spid/v1.0/identificationProfiles/1bb58f2d-7db2-4b5d-8d58-7a5ea7aece58

Where the “1bb58f2d-7db2-4b5d-8d58-7a5ea7aece58” in the url was the profile ID we wanted to fetch, to “GET”. The following key/value parameters had to be entered in the “Headers” section of the Postman request : Ocp-Apim-Subscription-Key / a9e3e21c91124fc3bb63a9e8ec9813a4. The GET request could now be sent to the url.

1. *Result for every file* : All our profiles were rightly created, so our query returned the following in the “Body” section of Postman :

{

"identificationProfileId": "1bb58f2d-7db2-4b5d-8d58-7a5ea7aece58",

"locale": "en-us",

"enrollmentSpeechTime": 56.19,

"remainingEnrollmentSpeechTime": 0,

"createdDateTime": "2019-04-29T16:50:43.513Z",

"lastActionDateTime": "2019-04-29T18:16:34.600Z",

"enrollmentStatus": "Enrolled"

}

The results varied from user to user (because they have different “identificationProfileIds”, because they were not created at the same time and because user 3 has “fr-fr” instead of “en-us” in the “locale” section), but all returned 200 OK responses. They were all “Enrolled”, which means the enrollment did work.

1. *Conclusions and recommendations* : all profiles returned “Enrolled” the first time we tried to “GET” them, so no further recommendation will be suggested here. Very few and very simple steps have to be realized and are all crystal clear in the Microsoft Cognitive Services documentation.

**User verification**

1. *Main objective* : verify if a chosen user (with its profile ID) is present or not in a specific audio sample sent to a verification url. The Microsoft API will then analyze if the profile ID X can actually be heard in the sample and other data that we will describe to you below.
2. *Choice of implementation and parameters* :