vScription Transcribe Pro Backend Recognition Specifications

Last Update: 11DEC2020

## Summary:

We want to add backend voice recognition to vScription Transcribe Pro using the rev.ai cloud speech to text engine. This engine does not support dictation (ie: Letters etc… where you would use things like New Line, Punctuation etc…). it is for meeting notes, interviews etc…. An application for this would be Zoom or MS Teams meetings

I have done some testing with the rev.ai meetings and sent some Zoom meetings and MS Teams meetings through the engine and the recognition rate is really good and is a viable solutions for meetings, notes interviews etc..

## Value Adds:

There are services out there where you can upload audio and get a transcript back but the advantages we will have will be as follows;

* Be able to review and edit the transcript with the built in web player as well as use a foot control to stop and start playback
* Job management. You can do basic document management
* Ability to search for specific words and jump to that position in the audio to review what was said or to cherry pick information as needed
* Follow text during playback to know where you are in the document
* Easy to use
* Flexible billing options. Per transcript or pre-pay (Need to think about this one)
* Competitive pricing
* Geographic servers (Canada and US)

## Basic Workflow Diagram:

Diagram

Description automatically generated

## Workflow Documentation:

### Enable backend speech by default per account

* If account is enabled for speech recognition, all uploaded jobs will go to the rev.ai engine and follow the speech rec workflow.
* On the Job Uploader, need to have an override option to prevent jobs from going for speech recognition. They might want this if it is a bad recording or other reason. This will be a per job/upload option
* If we do this we may need to add an sr\_override option to the vScription Upload application so that if the user is using the vScription Upload on a SR enabled account they can prevent from going to speech recognition
* Job Uploader will need some modifications we could use the session information to display certain information;
  + Speech Recognition Override (Checkbox)
  + Add estimated charges column when adding files. This is to be rounded up to the next 15 seconds as per rev.ai’s billing. (Length [in Mins.decimal ie: 1:30 would be 1.5, 1:45 would be 2, 1:13 would be 1.25 etc..] X SR\_Billrate from account table (New Column))
* For future flexibility, maybe we could automatically set override for specific worktypes? Not needed right now but something to think about

### Pros:

* Easiest to deploy and manage
* Is pretty flexible. If an account isn’t set for sr\_enabled, none of their jobs will go through the engine and they can’t use it (Good thing). If an account is set to sr\_enabled, they can still override if necessary.
* Since the document does leave our servers to get recognized (Even though it will get deleted immediately, most users with sensitive data won’t want to use this so an account that has sensitive data should have a separate account for non-sr if they want to transcribe meeting notes.

### Cons:

* Although the override is needed, technically an account could go to override and not send anything through the engine and technically use the platform for free. Maybe we should have a flat rate for override documents (That don’t’ go through SR)?

## Billing:

* We have a few options here.
  + Bill each time a report is run through the engine (Preferred)
  + Have client prepurchase recognition minutes
  + Bill client on a schedule (ie: weekly)
* I was looking into use the Zoho Books APIs but I think this can be done easier with the authorize.net suite (<https://developer.authorize.net/>)
* My thoughts are that after adding all of the files, there will be an list of what the charge will be. When the client clicks UPLOAD JOBS a payment window will popup and take the information. Once the payment is complete, it will be pre-authorized, not billed. Once the report is returned from rev.ai, the transaction will be captured.
* The invoice will need to have the following information;
  + Job Number (vScription Transcribe Job Number)
  + Amount billed
  + Taxes
  + Full Name
  + Full address (City, Province, Postal Code, Country)
  + Email address
  + Transaction Date
  + Transaction ID

I will need to look into this more to get more details on how the billing will actually work.

## Database Table Changes Needed:

### Account:

* sr\_rate (decimal 10,2)
* sr\_enabled (int)
* doc\_flat\_rate (decimal 10,2) – May be used if user overrides speech rec since an account like this will be completely transactional
* sr\_vocabulary\_words (mediumtext) – May be used in the future, When sending jobs to rev.ai, you can include words that may not be in their dictionary that it uses to increase recognition accuracy. (<https://www.rev.ai/docs/overview#section/Features/Custom-Vocabularies>)
* total\_sr\_minutes(decimal 10,2) – This will store the total minute that a user has sent for recognition

## Transcribe Changes Needed:

### Word Search Function:

* Add new search field to text area. This could be in the Shortcuts section or the Shortcuts Section could be split in half and add the search to the right. Here’s a crappy mockup;  
  Graphical user interface, application

  Description automatically generated
* After clicking the search icon or pressing enter, a popup will open that will display all of the occurrences of the search string and their approximate time. The items will be hyperlinks and if you click one, it will highlight the word in the text and move the audio to that position as well. Then the popup will close.
* The audio markers are going to be returned from rev.ai with the document so I imagine the captions file will need to be stored in the files table record.

Here’s a snippet from a .vtt file returned from rev.ai;

WEBVTT

1

00:00:00.000 --> 00:00:05.000

And then when I picked police service that we might clearly define between the

2

00:00:06.360 --> 00:00:11.190

two parties, exactly what it is each group looks to get out of,

3

00:00:11.430 --> 00:00:12.263

uh,

4

00:00:12.510 --> 00:00:17.510

out of this relationship and out of these officers being in schools.

5

00:00:18.280 --> 00:00:23.190

Uh, I appreciate the two reports that we already have on this.

So if a user searched for “relationship” for example, I’d like to see the results display something like this;

…of this relationship and out…

If this entry was clicked, it would move the audio to 00:00:12.510 (Or close to it), and select that word in the body text.

### Bouncing Ball Feature: (Text Follow Along):

If there is an associated captions file, it is assumed that the user will not be re-typing the document so what I want to happen is As the text plays, the body text gets highlighted.

So for the example above, when you are at the start (00:00:00), the first text section (And then when I picked police service that we might clearly define between the) gets highlighted until the audio reaches the next section (00:00:06) at which point the next set of text gets highlighted ( two parties, exactly what it is each group looks to get out of,) and is selected until 00:00:11 etc….

Note: Able player has some of this functionality built in so it may be easier to implement it that way versus redeveloping. There are sample files in the repo under the docs folder in the samples folder. All available downloadable formats from rev.ai are in there for reference.