About this project

This project is built using the Spring Web MVC framework. The audio that the web application captures is analyzed by CMUSphinx, an open source speech recognition library.

To get the project up and running on your local machine, read the section in this document on “Setting up the development environment”.

Before starting to work on this project, it is highly recommended that you read the Spring MVC documentation, particularly parts II and III, in order to understand how the underlying framework functions.

- https://docs.spring.io/spring-boot/docs/current/reference/html/index.html
- https://docs.spring.io/spring-boot/docs/current/reference/pdf/ (pdf)

Capturing the audio data and displaying it as a graph is done with Mozilla’s Web Dictaphone example, available on Github and explained in detail here.

The web app uses WebSockets to transmit the audio data to the server. The following link explains how to use WebSockets in SpringMVC:

- https://spring.io/guides/gs/messaging-stomp-websocket/

The audio must be in the following format for analysis by CMUSphinx: RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, mono 16000 Hz. Documentation on CMUSphinx and the data it gives back can be found here:

- https://cmusphinx.github.io/wiki/

We use Let’s Encrypt as our HTTPS Certificate Authority. You must control the domain that you register the certificate for – AWS addresses will not work, as they are ephemeral and easily regenerated. You will have to register and use your own domain name. Information on using the Let’s Encrypt Certbot to establish and automatically generate certificates can be found here:

- https://letsencrypt.org/getting-started/

Database Information

Host: ec2-54-225-112-61.compute-1.amazonaws.com

Database: ds0m2g3nacuvc

User: huzjrznhoilodv

Port: 5432

Password: e3670ef2720dba4d5b47c5298a34e481fc0765298256e29faf0a6237d4726983

URI: postgres://huzjrznhoilodv:e3670ef2720dba4d5b47c5298a34e481fc0765298256e29faf0a6237d4726983@ec2-54-225-112-61.compute-1.amazonaws.com:5432/ds0m2g3nacuvc

Heroku CLI: heroku pg:psql postgresql-opaque-96001 --app micronophones

WARNING: Heroku says these credentials aren't permanent, so if you are having trouble logging in, log into our micronophones heroku account navigate to "heroku postgress" -> "settings" -> "database credentials"

Amazon: Host: micronophonesdb.cx3jzyuelefs.us-west-2.rds.amazonaws.com

Database: micronophonesdb

User: micronophones

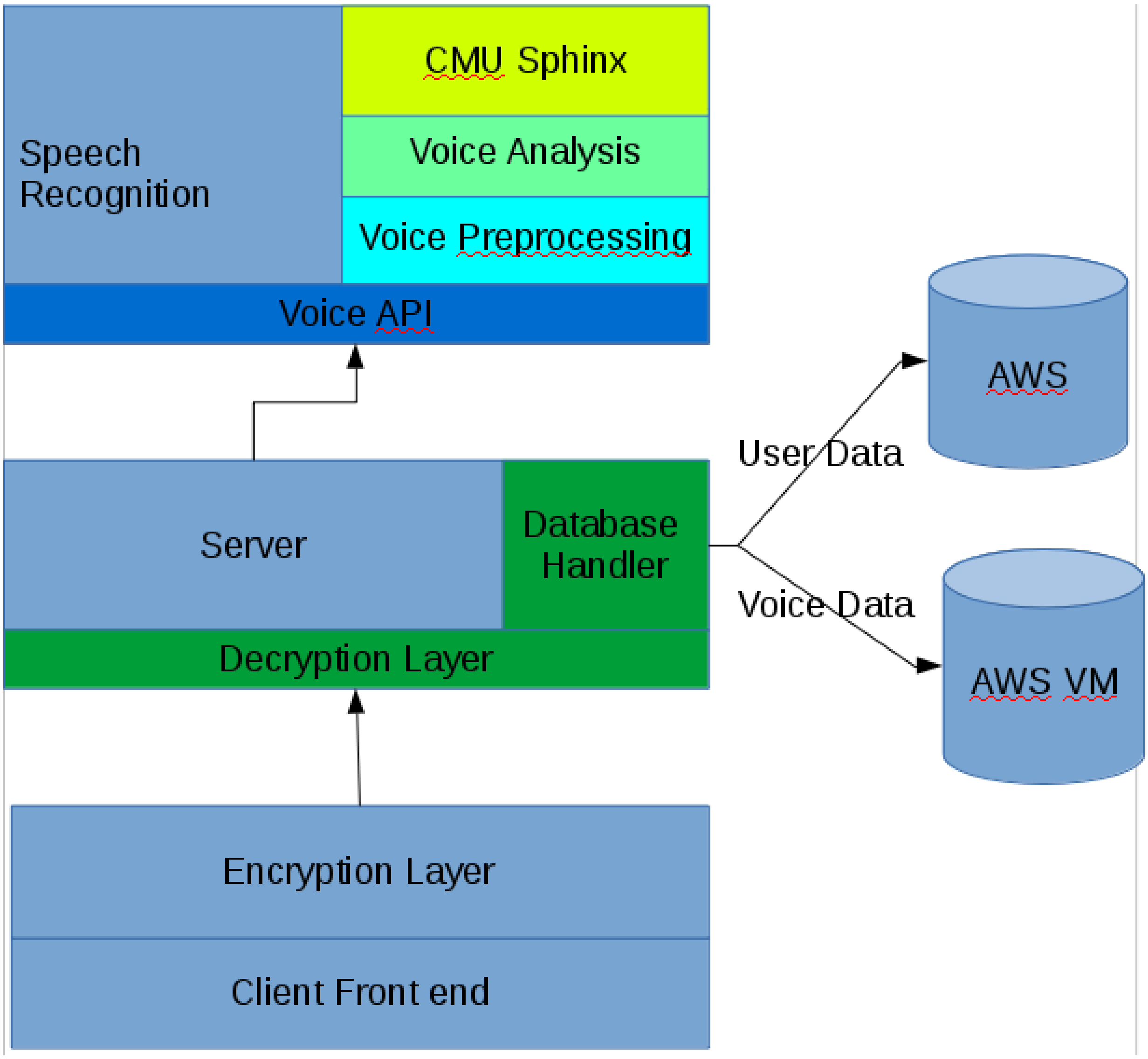
Port: 5432

Password: Micr0n0ph0nes

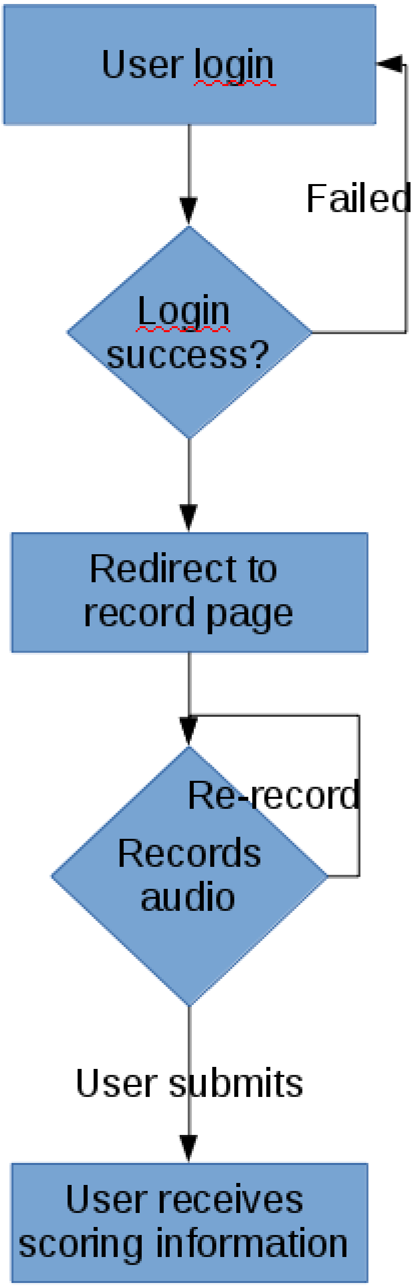
Master Password: DouglasDolphinFish42

Framework

High Level Framework Diagram



User Happy Path



Git Branches

On our host we will always be hosting a working version of our code. The branch will be created at the start of each sprint and the branch name will be sprint#, where # is the current sprint we are on.

So we will always be pushing/pulling from master so be cautious when committing your code to master, make sure the code can actually be built using maven locally and also be sure to test the code on the website itself once built.

If you don't want to work on master you can always create branches for the the task you are working on and then merge the branch onto master once it is working and done.

To create a branch and automatically go to it, type the following command:

```
$ git checkout -b [branch-name]
```

You can now code on this branch without affecting the master branch. You can add and commit as normal on this branch. Once you are ready to merge the branch back onto master be sure to push the branch with the following command:

```
$ git push --set-upstream origin [branch-name]
```

If you have multiple commits on your branch, you should squash your commits to 1 commit with the following command.

```
$ git rebase -i master
```

This will pull up a page with all of the commits you have made. The first commit will be left the same. The other commits will need to be changed from "pick" to "fixup". Then save and exit the document. This is also where any conflicts will be resolved if any. You can also change the commit message itself with the following command:

```
$ git commit --amend
```

This will pull up a prompt where you can change the commit message.

Once all of this has been done, force push the changes with the following command:

```
$ git push -f
```

Now you are ready to merge this branch back onto master. To do this first checkout the master branch:

```
$ git checkout master
```

Then type the following command, to merge the branch onto master:

```
$ git merge [branch-name]
```

Setting up the development environment

Download IntelliJ Community Edition from here https://www.jetbrains.com/idea/download/ and install it.

Check out the project from version control either from the first welcome screen (`Check out from Version Control > GitHub`) or from the main window (`File > New > Project from Version Control > GitHub`). You will get a window where you will need to log in to GitHub. Click `Create API token` and enter your credentials. The next window will prompt you for a Git repository URL - copy/paste `https://github.com/BoiseState/CS471-F17-Micronophones.git`, and click `Clone`.

Once the repository has finished downloading, there should be a popup at the bottom of the window (or a clickable line of text at the bottom of the window if you missed the popup) that says `Non-managed pom.xml file found`. Click on it and then click `Add as Maven Project`. Depending on your internet speed, it may take some time to resolve all the dependencies and finish all background tasks.

[setting up project SDK ?]

Once finished, in the explorer panel on the left, navigate to `src > main > java`. Right-click on `Main` and select `Run 'Main.main()'`. (At some point you may get a window that wants to add a bunch of dependency-related files to Git. Either click `Cancel` or uncheck everything and click `Ok`.)

If the project successfully compiled without errors, you should now be able to navigate to `localhost:5000` in your browser and view the website locally.

Users

Researcher:

As a researcher, I want several things out of the system. First, I want to be able to create and login to my online account where I can view information about speech data in the database. I need to be able to create and manage API keys in my account so that I can ping the server for speech data. I also need sample code on my login screen that shows me how to use the API keys to get data from the server.

Since I am gaining access to personal data, I expect that there will be a verification process when I try to create my account.

Therapist:

As a therapist, I also want several things out of the system. First, I want to be able to create and login to my online account. On my online account, I expect to see a list of patients that I have, and I want a quick display of the patient's progress. I should be able to click into my patient's profiles and see more detailed information about how they are performing and improving. For each patient, I expect to see at a minimum the following data: recent activity (last time logged in, duration, etc.), scores, and even voice snippets (so I can hear and further help them improve their speech), and the patients therapy plan.

The most important thing is that for every patient I can view, modify, and create their personalized treatment plan. At a minimum for the treatment plan I should be able to specify what my patients are reading, i.e. I should be able to specifically set the "game" they are working with. For each "game", I would like to hear the systems text-to-voice translation, and if I find it will not be satisfactory to help my patient, I would like to be able to supply my own audio to play for the patient when they click the audio button for that "game". At an advanced level, I would like to set the types of sentences, phrases, text, etc. that my patient is getting, but not necessarily give each patient specific sentences, phrases, text, etc.

For each patient, the treatment plan will consist of "games", I have already mentioned some of the games above (sentences, phrases, text, etc.), but they might also include single sounds, non-dictionary words, or other things. I would like to be able to provide these things to my patient.

Finally, as a therapist, I want to be able to add new patients. This consists of me linking my profile with theirs. I will need to be able to send a request to a user, and once they accept they will be my patient.

Patient:

As a patient, I also want several things out of the system. First, I want to be able to create a login to my online account. On my online account, I expect to see several things: a games page, a treatments plan page, and a scores page.

The games page should load the next game in my treatment plan. Once the game loads up I want to be able press a button and have the system speak the “game” to me through the speakers. I also want to be able to press a button and begin recording myself speaking the “game”. After I have a recording of myself speaking the “game” I want to be able to listen to what I just said, and try again if I didn’t like it. Once I feel good about my recording, I want to press submit, and have the system come back and score me on how I did. I also want it to tell me how to improve specifically, for example if I was pronouncing a certain syllable wrong I want it to tell me. Once I am satisfied with my score I also want to be able to go onto the next “game” in my treatment plan. If there is nothing currently in my treatment plan I expect some random “game” to come up so that I can practice on that. If I’m getting a random game it would also be nice if I could select the difficulties ranging from easy, medium, and hard.

On the treatment plan page, I want to be able to see progress in my treatment plan, as well as get a sneak peek as to what is coming up. Ideally this would display as some sort of nice chart or graph, this can could also serve as a nice way to show me progress I’ve made. I expect to see things like percentage done with treatment plan, improvement statistics, etc.

On the scores page, I want a more detailed summary of what I have been doing. For each game, I want to be able to see the scores. As I earn more points through my “games” I would also like to earn badges, these badges should show up on the scores page.