

S2T e T2S systems and respective conceptual models

LEI - 2020

Voice recognition for education systems

Work Plan 01

ggeop/Python-voice-assistant

"(...) voice assistant service in [Python 3.5+](#) It can understand human speech, talk to user and execute basic commands"

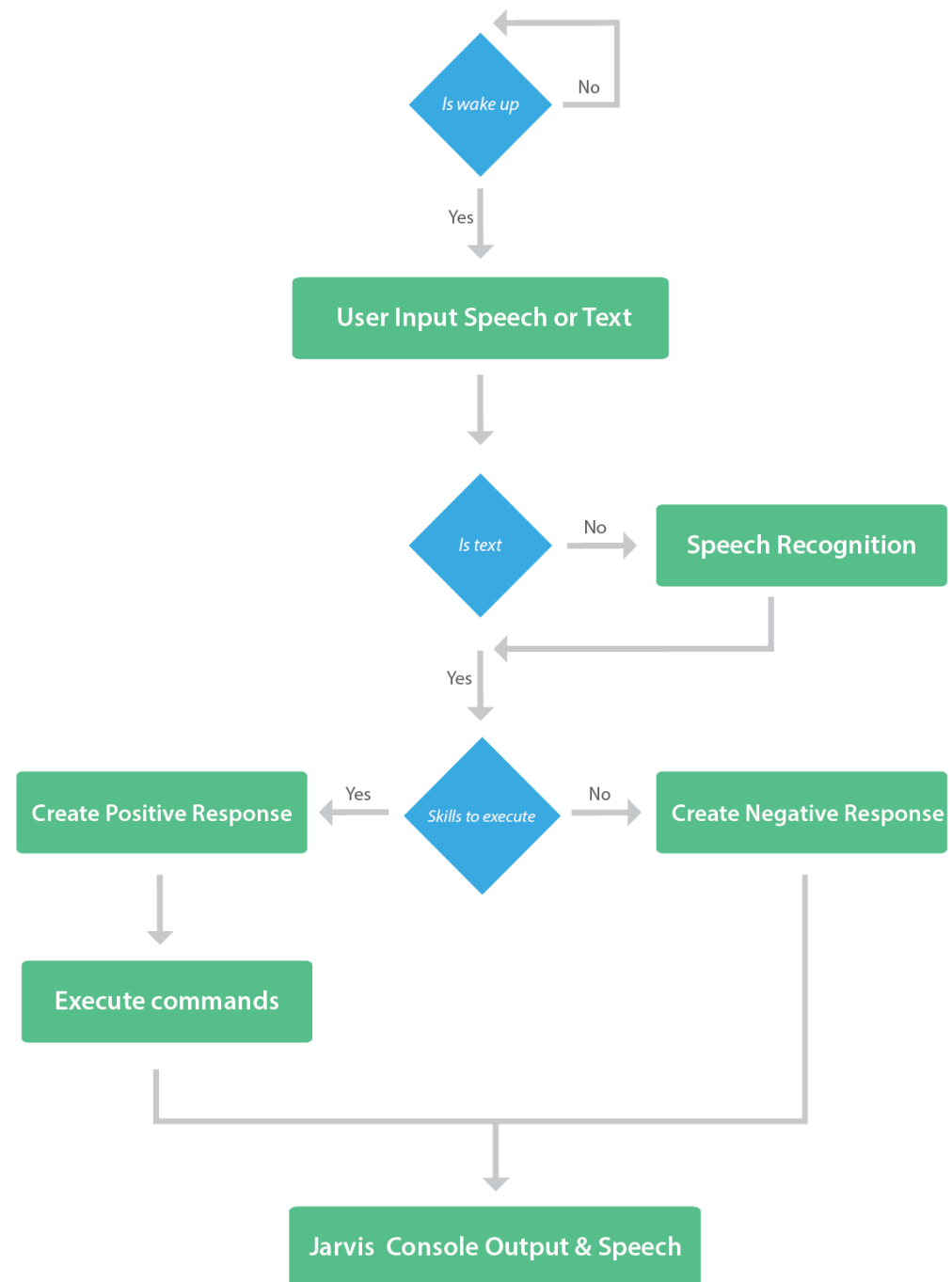
Features

- Asynchronous command execution & speech interruption
- Continues listening service
- Vocal or/and text response

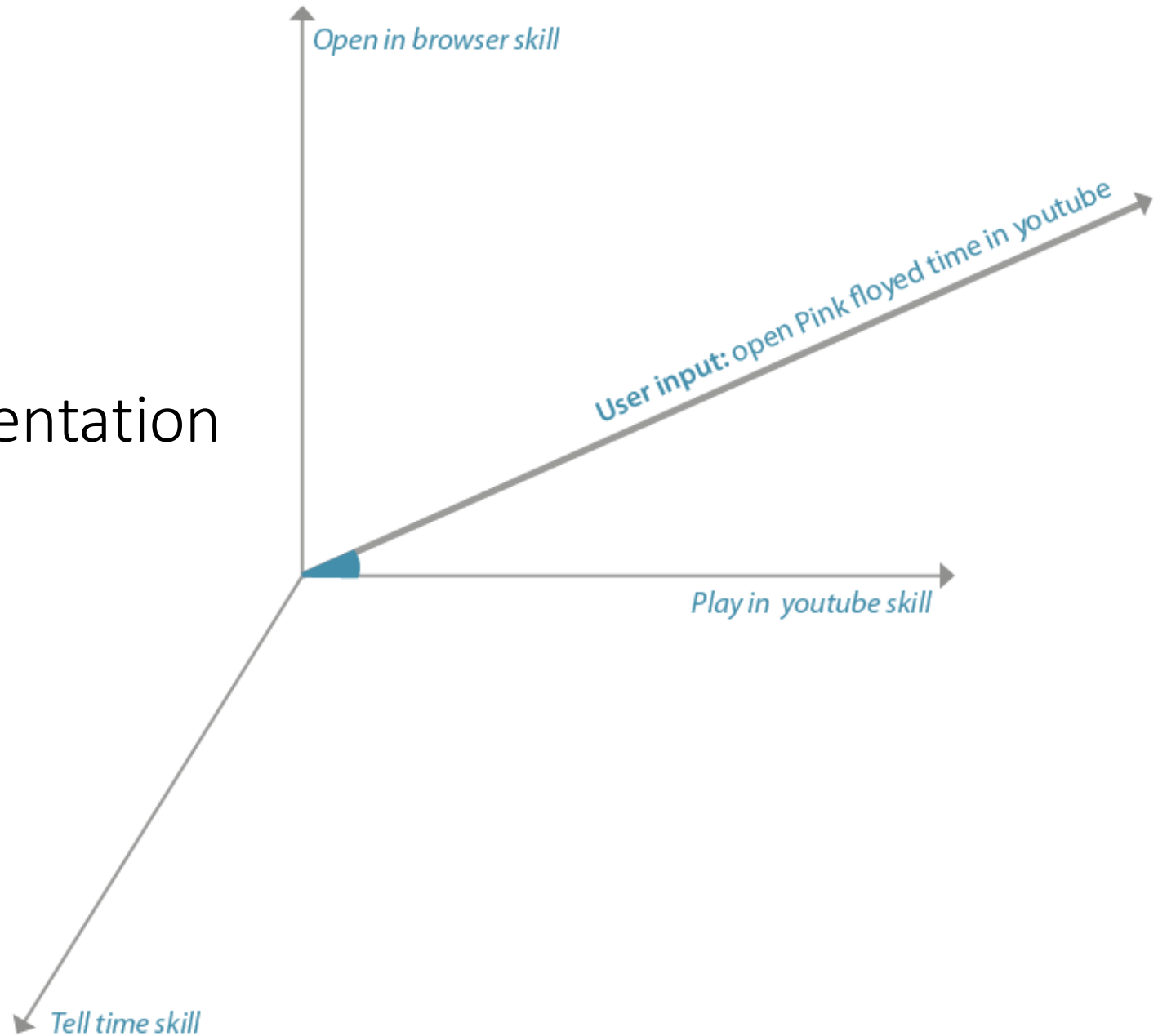
But why do we find this project interesting?

- Python?
- Skill based assistance?
- Code quality and popularity?
- Configurable third party APIs for speech recognition?

Decision Modeling



Skill extraction implementation
in a matrix of TF-IDF
features for each skill



[sklearn.feature_extraction.text](#).TfidfVectorizer



Treino do modelo de seleção de Skill

```
SKILLS = {  
    'tell_daily_news': {  
        'enable': True, 'skill': BrowserSkills.tell_me_today_news,  
        'tags': {'news', 'today news'},  
        'description': 'Ask me to tell the daily news e.x "Tell me the news  
today"  
    },  
    ...  
}  
tags = ['n e w s', 't o d a y n e w s']  
  
vectorizer.fit_transform(tags)
```

```
def extract(self, user_transcript):  
    train_tdm = train_model()  
    test_tdm = vectorizer.transform([user_transcript])  
    similarities = calculate_similarities(train_tdm, test_tdm)  
    most_similar_skill_index = similarities.argsort(axis=None)[-1]  
  
    if similarities[skill_index] > min_match:  
        skill_key = [skill for skill in enumerate(self.skills) if skill[0] == skill_index][0][1]  
        return self.skills[skill_key]  
    else:  
        return None
```


[leon-ai/leon](#)



6.6k

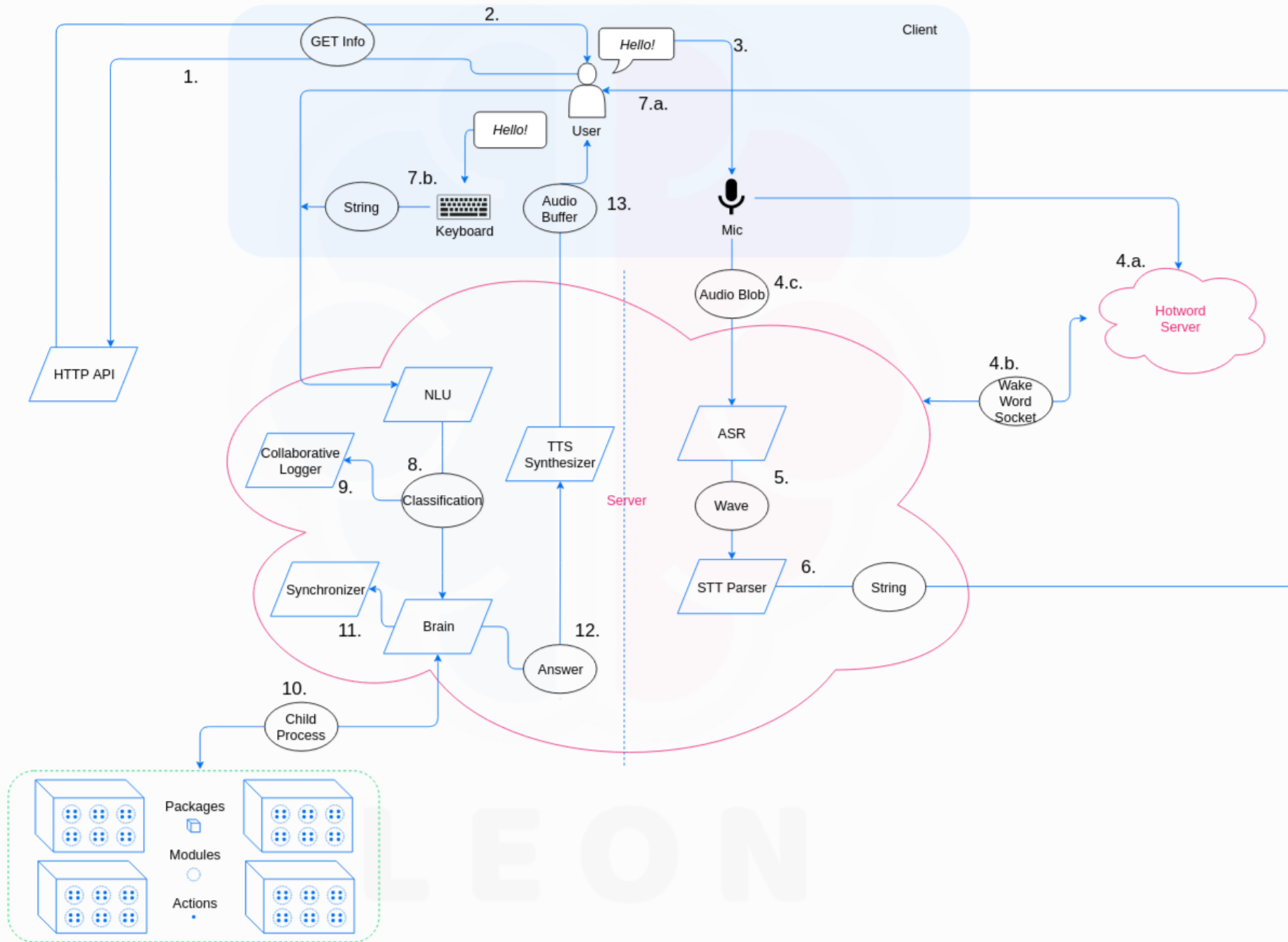
"Leon is an **open-source personal assistant** who can live on your server.

He does stuff when you ask him for.

You can talk to him and he can talk to you. You can also text him and he can also text you. If you want to, Leon can communicate with you by **being offline** to protect your privacy.

You can think of him as a second brain."

High-Level Architecture



1. User talks with their microphone, client transforms the audio input to an audio blob.
2. ASR transforms audio blob to a wave file.
3. STT parser transforms wave file to string.
4. NLU classifies string and pick up classification.
5. Brain creates a child process and executes the chosen module.
6. Brain creates an answer and forwards it to TTS synthesizer.
7. TTS synthesizer transforms text answer (and send it to user as text) to audio buffer which is played by client.

S2T and T2S providers

- Amazon

- Google

- IBM Watson

1. Create or select a project on the GCP (Google Cloud Platform).
2. Enable the S2T API and/or the T2S API.
3. Create a service account key.
4. Download the JSON file and copy its content to the: `server/src/config/voice/google-cloud.json` Leon's file.

but also offline...

DeepSpeech (S2T):

For the offline STT, Leon uses **DeepSpeech** which is a TensorFlow implementation of Baidu's DeepSpeech architecture. Everything is ready, you just need to run a command to download and setup the **pre-trained model (~ 2 GB)**. If you wish to only install the offline STT, run the following command:

```
`npm run setup:offline-stt`
```

Flite (T2S):

For the offline TTS, Leon uses **CMU Flite** which is a small, fast runtime open-source text-to-speech synthesis engine **developed at CMU** (*Carnegie Mellon University*).

Here also, everything is ready, you just need to run a command to download the binary and configure. If you wish to only install the offline TTS, run the following command:

```
`npm run setup:offline-tts`
```

Web App

Hello

Good morning, I hope your day will be full of joy and productivity!

Who are you?

I'm your daily personal assistant. I have been created by Louis.
I'm very happy to serve you everyday.

Give me a random number


5

Check if github.com and twitter.com are up


I'm checking Github state.

Github is working correctly.

I am now requesting Twitter.



Use ↑ ↓ to browse history; ↵ to submit; alt + t to listen.


v1.0.0-beta.0
Collaborative logger enabled, thank you.

Modules

1. Create config file
2. Setup Expressions and train module
3. Create Answers for possible outcomes
4. When Leon understands what you told him, he:
 - Triggers a module action.
 - Do the job.
 - Returns you the output of that execution.

config.sample.json:

```
{
  "youtube": {
    "api_key": "YOUR_GOOGLE_API_KEY",
    "playlist_id": "PLAYLIST_ID",
    "options": { "synchronization": { "enabled": true, "method": "direct", "email": "" } }
  }
}
```

expressions/en.json:

```
{
  "youtube": {
    "run": { "expressions": [ "Download new videos from YouTube" ] }
  }
}
```

answers/en.json:

```
{
  "youtube": {
    "success": [ "All of the videos have been downloaded.", "I finished to download the videos." ],
    "downloading": [ "I'm downloading%video_title%",
    ...
  }
}
```


Voice assistants in education



AI-powered service for learning foreign languages



Intelligent tutoring system designed and optimized for educational conversation

Cognii

What are the structures of a typical neuron?

Student

Neurons are the basic building blocks of the nervous system. A neuron consists of dendrites and a cell body called soma.

Cognii

Very close! Would you like to explain the neural transmitter?

S2T e T2S systems and respective conceptual models

LEI - 2020

Voice recognition for education systems

Work Plan 01

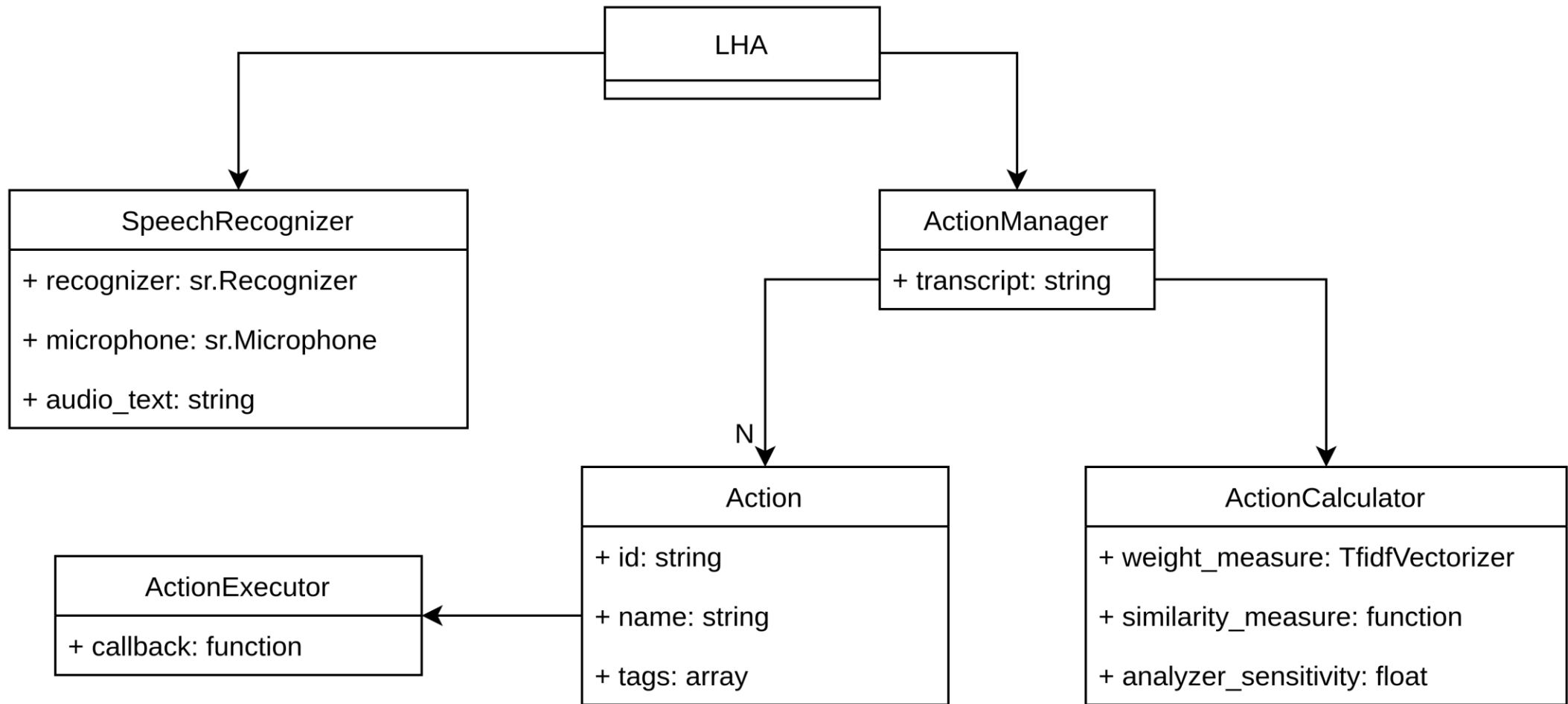
System Analysis & Design

LEI - 2020

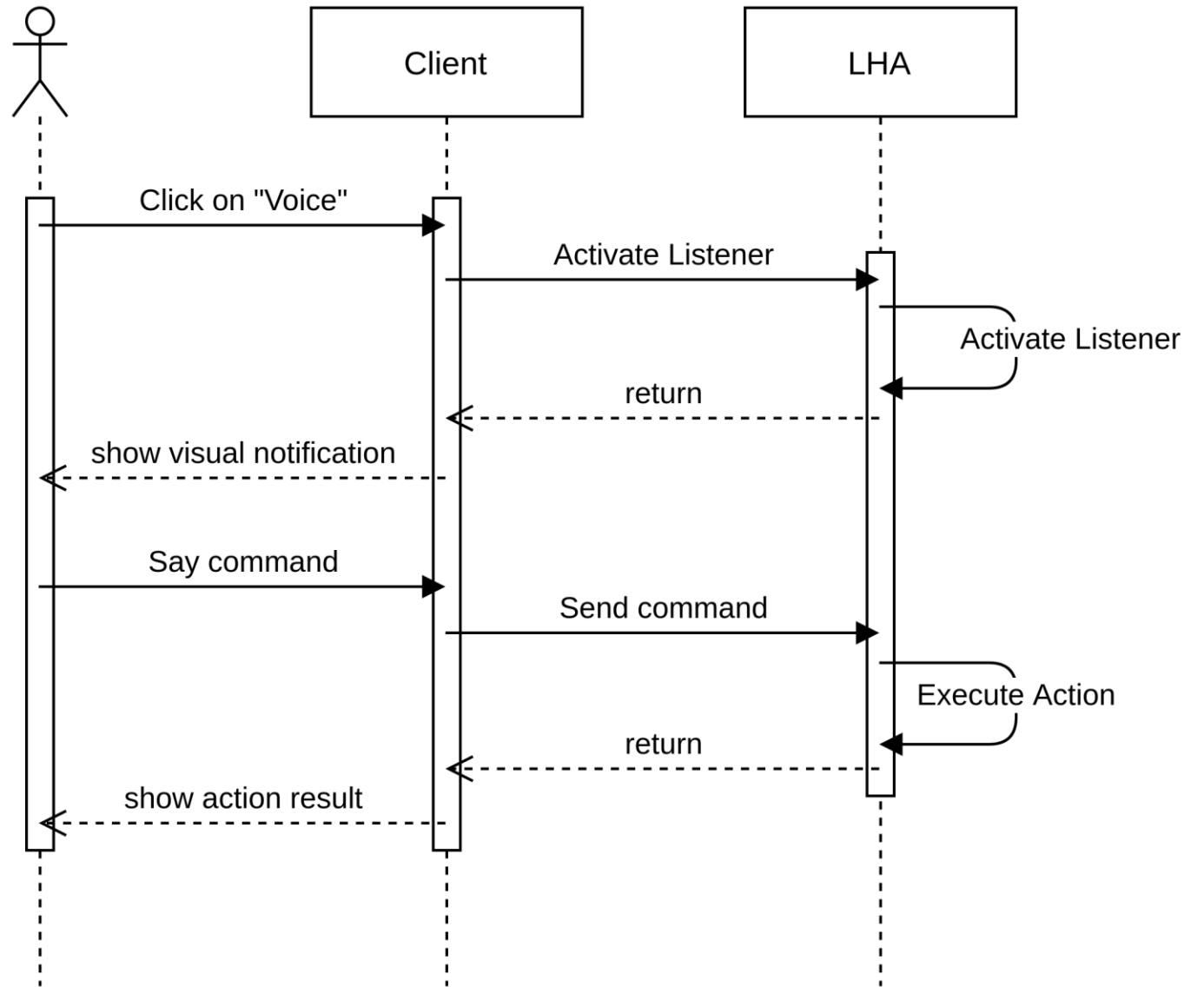
Voice recognition for education systems

Work Plan 02

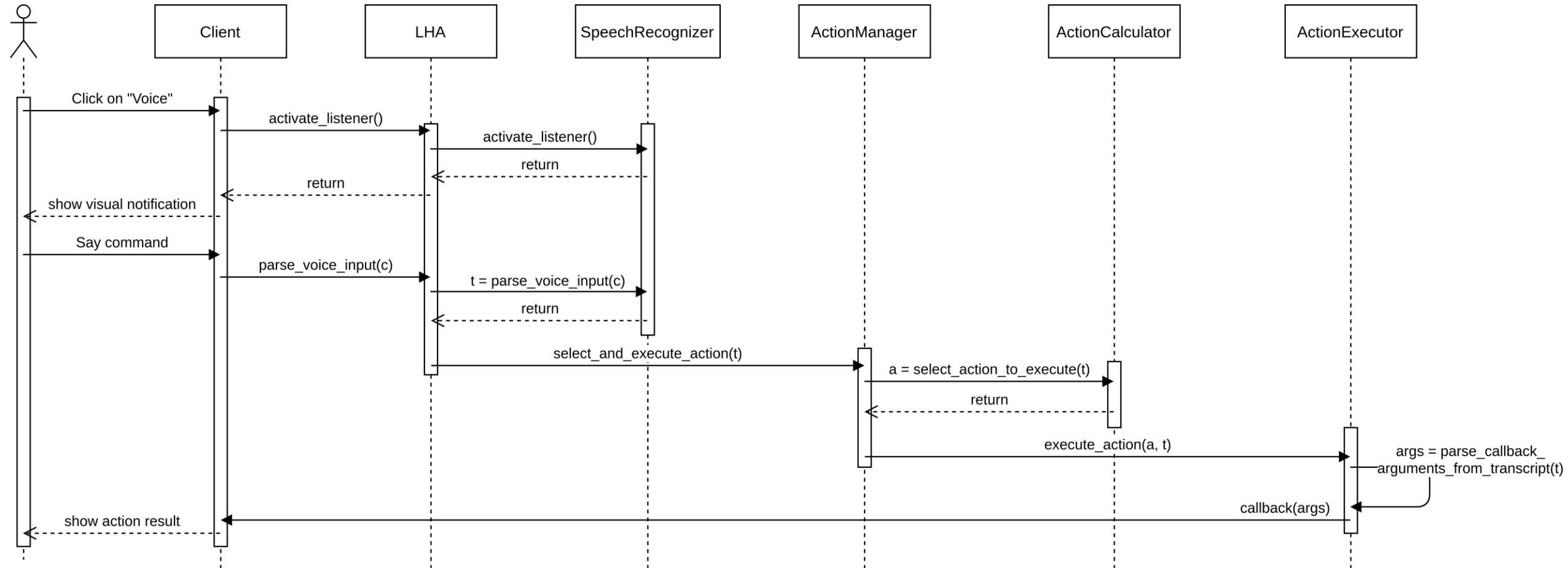
Classes Diagram



System Diagram



Subsystem Diagram



Action Example

```
class Action:  
    action_id: int  
    name: str  
    tags: list  
    callback: Callable
```

```
---
```

```
def create_action() -> Action:  
    action: Action = Action(0, "Selector", ["select", "answer"], callback)  
    return action
```

```
def callback(answer: str, certainty: str):  
    print("Selected {} with {} certainty.".format(answer, certainty))
```


System Analysis & Design

LEI - 2020

Voice recognition for education systems

Work Plan 02

Implementation

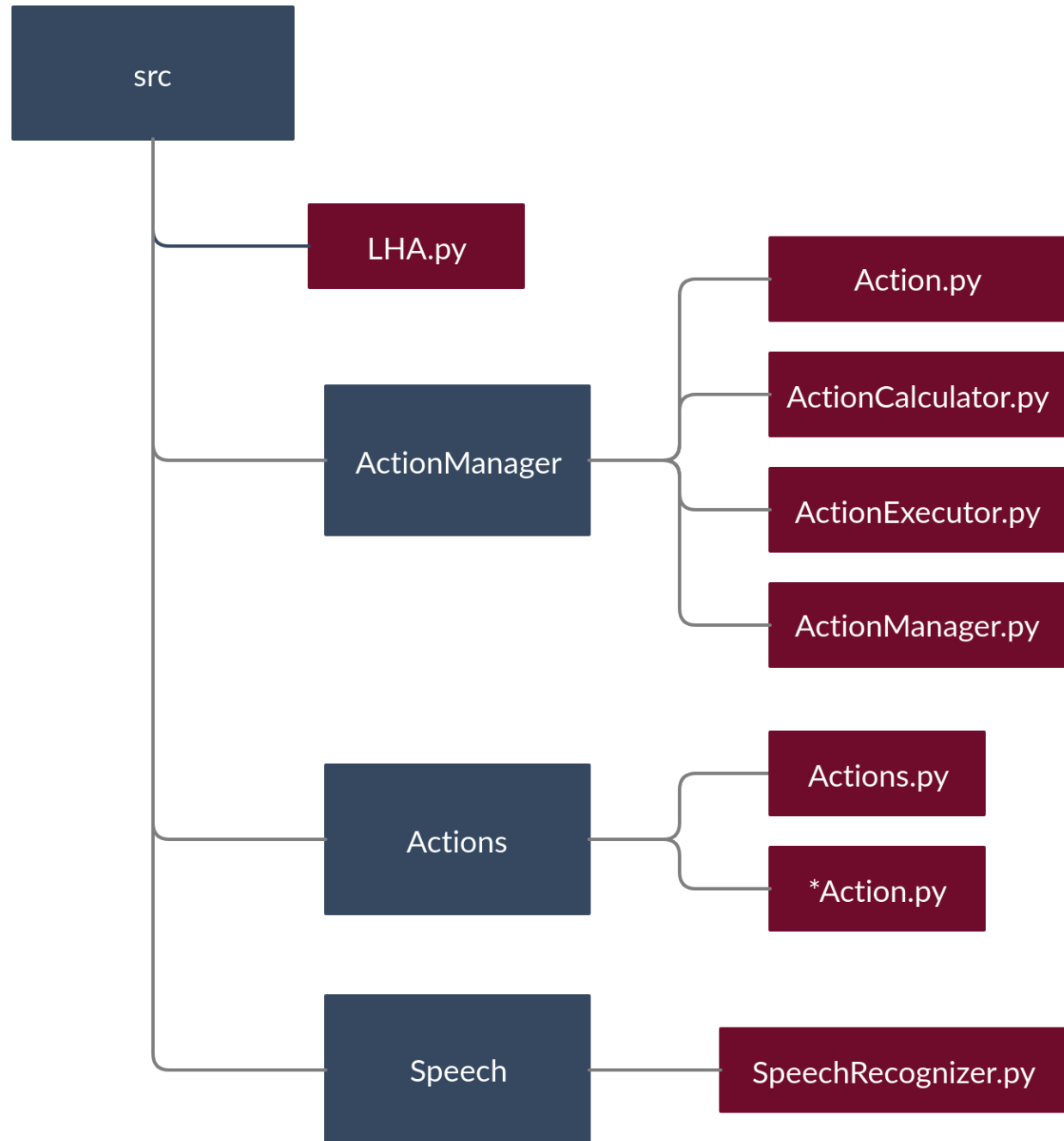
LEI - 2020

Voice recognition for education systems

Work Plan 03



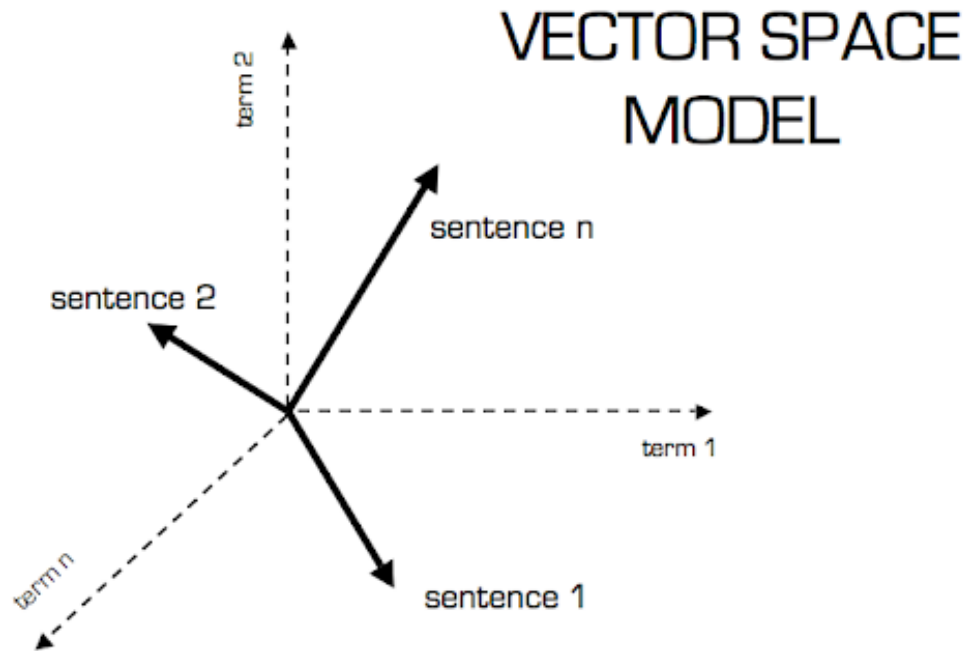
File structure



Simplified speech recognition algorithm

```
with speech_recognition.Microphone(  
    device_index=self.device_index,  
    sample_rate=self.sample_rate,  
    chunk_size=self.chunk_size  
) as source:  
    self.sr.adjust_for_ambient_noise(source)  
  
    audio = self.sr.listen(source)  
  
    transcript = self.sr.recognize_google(audio, language='pt-PT')  
  
    return transcript
```

ActionCalculator - TfidfVectorizer with cosine_similarity





Frase: "Fixa opção um"

Modelo: [
['selecionar','escolher','opção'],
['próxima','questão'],
['anterior','questão'],
['fixa','questão','opção']
]

Teste: [
['selecionar','escolher','o
pção'],
['próxima','questão'],
['anterior','questão'],
['fixa','questão','opção']
]

Creating Actions for possible options in the interface

 Leonardo



Studying

Question nº 1 | Level 2 Time left: 25

Consider the logical scheme presented in the figure. Which of the following queries allows you to indicate payments over '5' made by customers '1' and '5'?

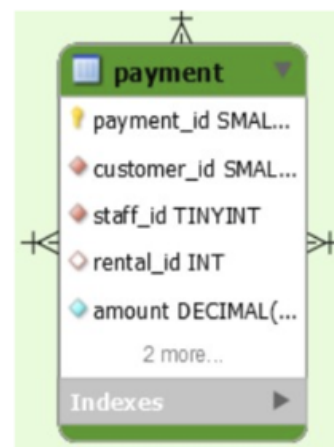
☐ SELECT Customer_Id, Amount FROM Payment WHERE Customer_Id = 1 OR Id=5 AND Amount >= 5;

☐ SELECT Customer_Id, Amount FROM Payment WHERE Customer_Id IN (1,5) AND Amount > 5;

☒ SELECT DISTINCT Customer_Id, Amount FROM Payment WHERE Customer_Id IN (1,5) AND Amount > 5;

☐ SELECT Customer_Id, Amount FROM Payment WHERE Customer_Id IN (1,5) AND NOT <= 5;

☐ None of the above.



Resposta

Anterior

Seguinte

Terminar

Porquê

Explica

Como

Opinião

Estatísticas

Voz

Leonardo

Regras

Ajuda

Acerca

Actions.py

SelectOptionAction.py

NextQuestionAction.py

PreviousQuestionAction.py

...

```
def create_action() -> Action:
```

```
    action: Action = Action(  
        "SelectOption",  
        ["escolhe", "escolher", "seleciona", "selecionar", "bloqueia", "bloquear", "opção"],  
        callback,  
        parse_callback_arguments_from_transcript  
    )  
    return action
```

```
def callback(answer: str):
```

```
    ...
```

```
def parse_callback_arguments_from_transcript(transcript: str):
```

```
    ...
```


Implementation

LEI - 2020

Voice recognition for education systems

Work Plan 03

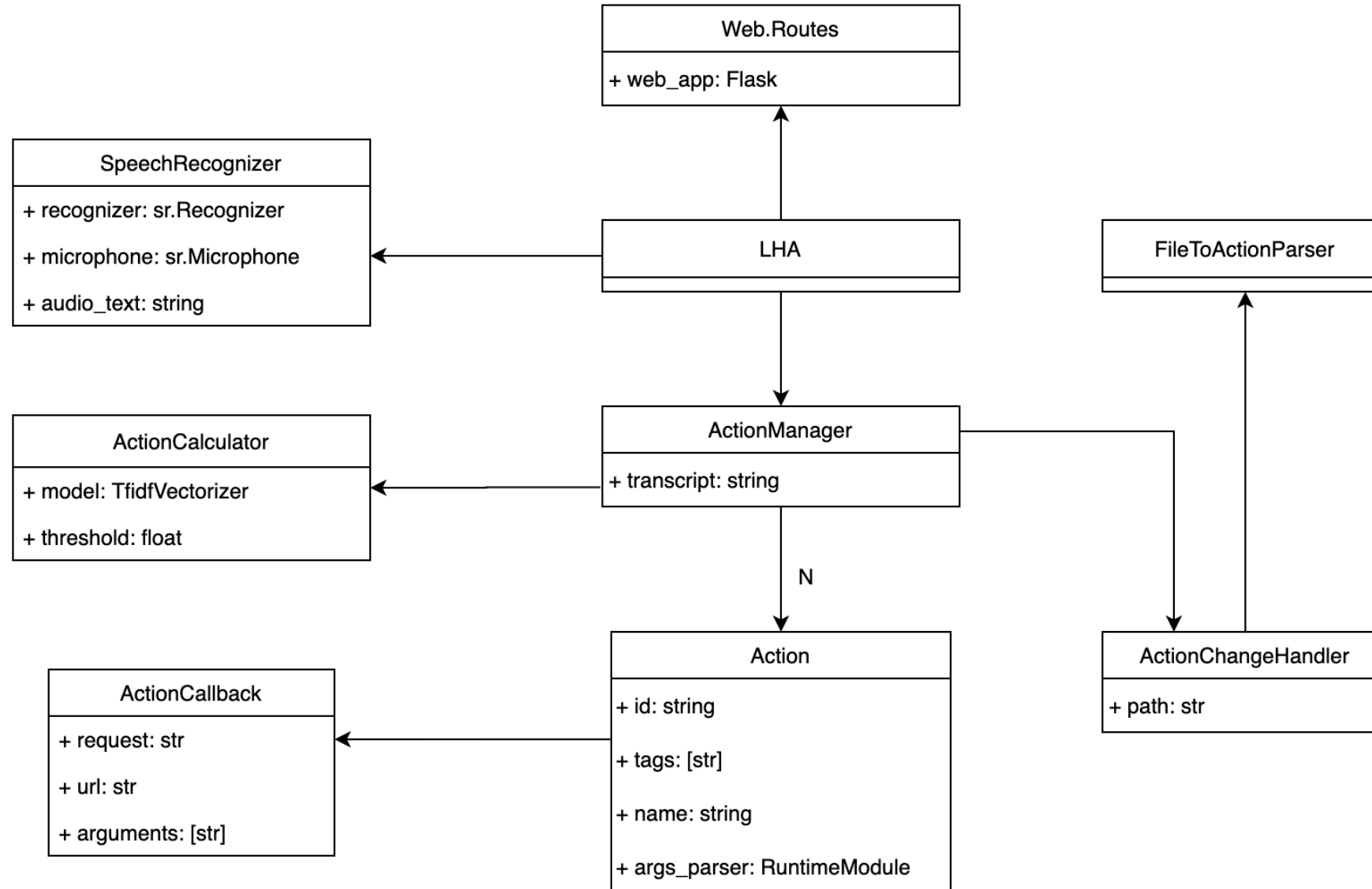
Implementation - 2

LEI - 2020

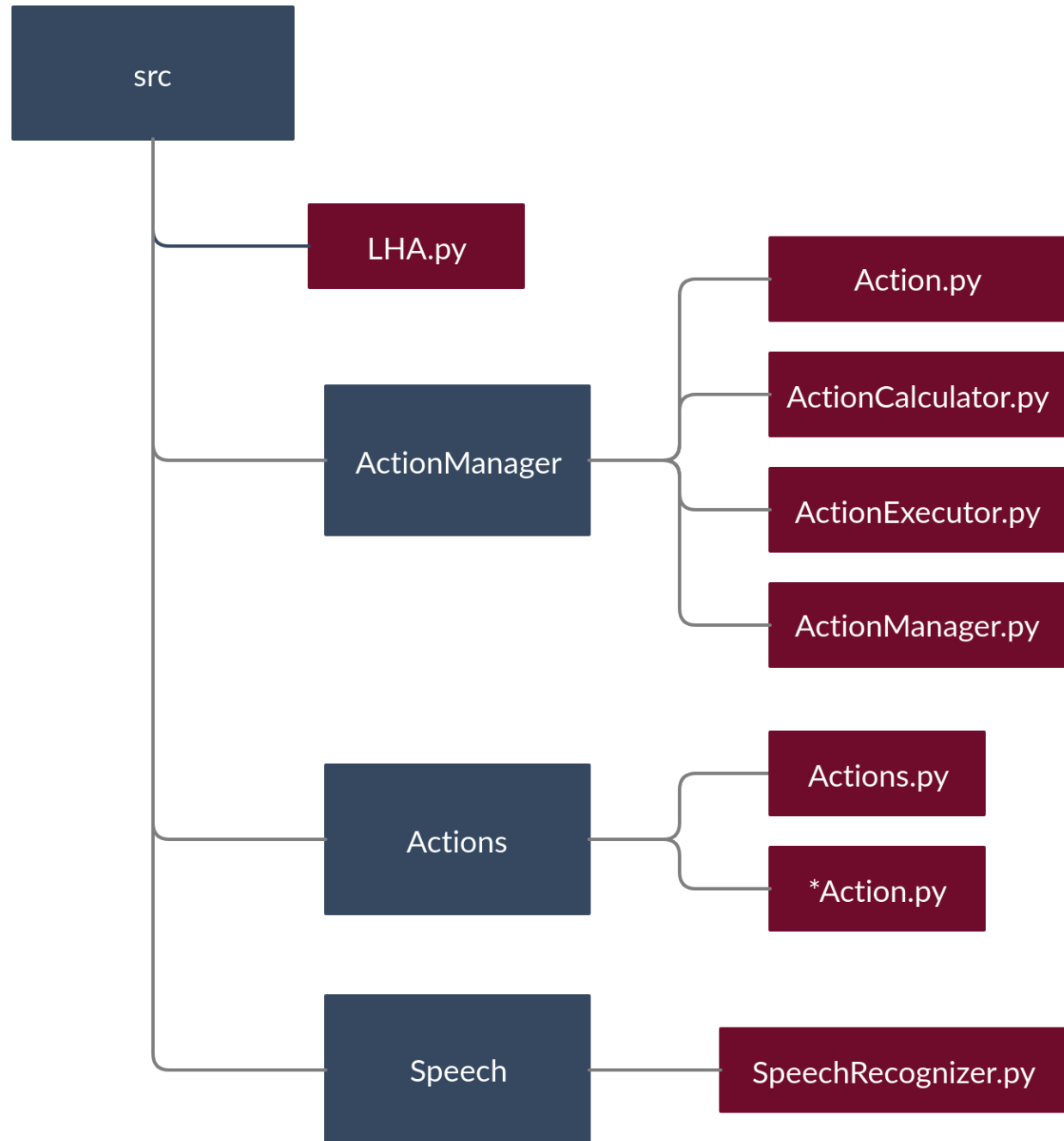
Voice recognition for education systems

Work Plan 04

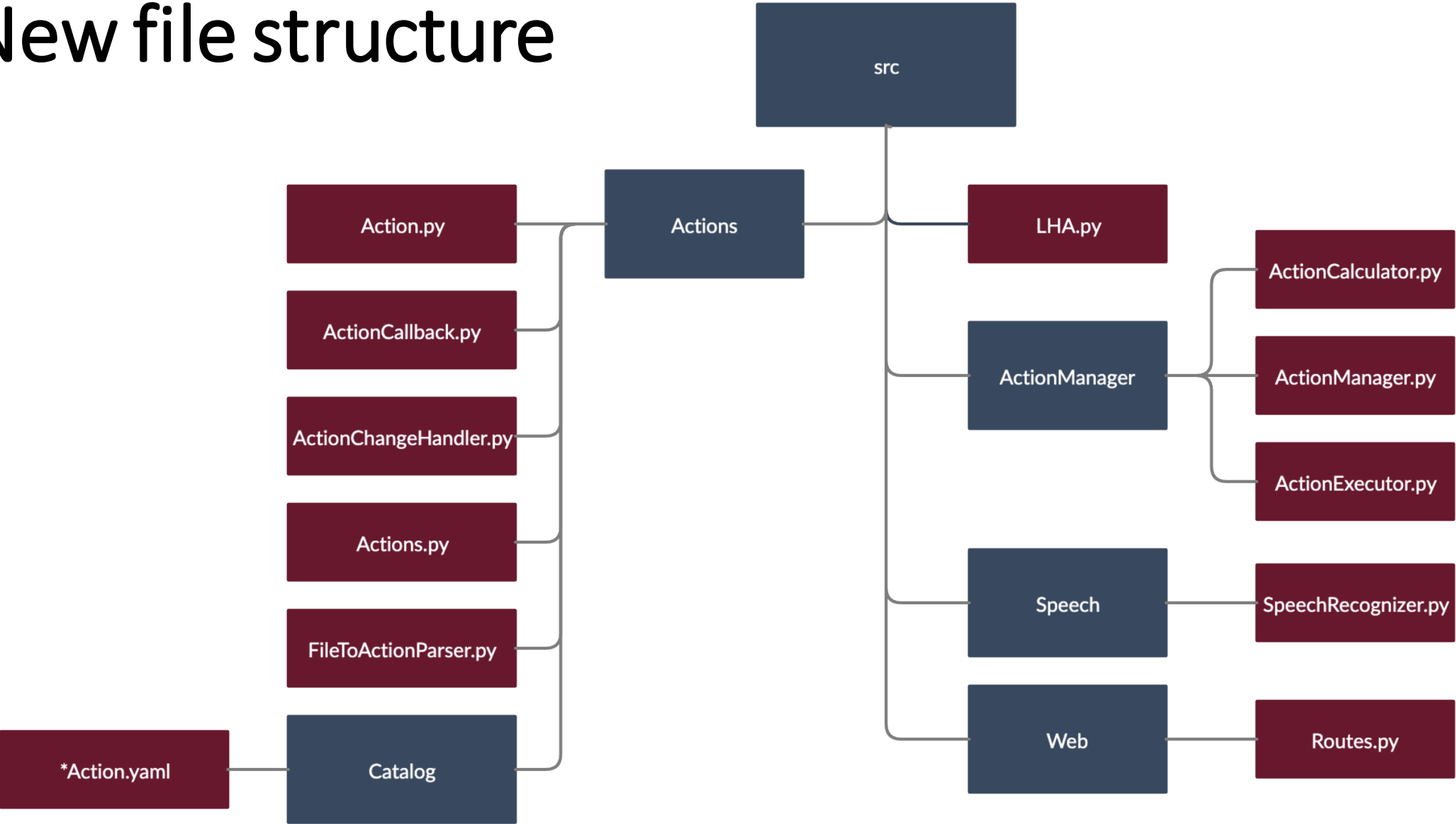
Parsing Actions from Configuration Files



Old file structure



New file structure



SelectOptionAction.yaml

tags:

- escolhe
- escolher
- seleciona
- selecionar
- bloqueia
- bloquear
- opção

callback:

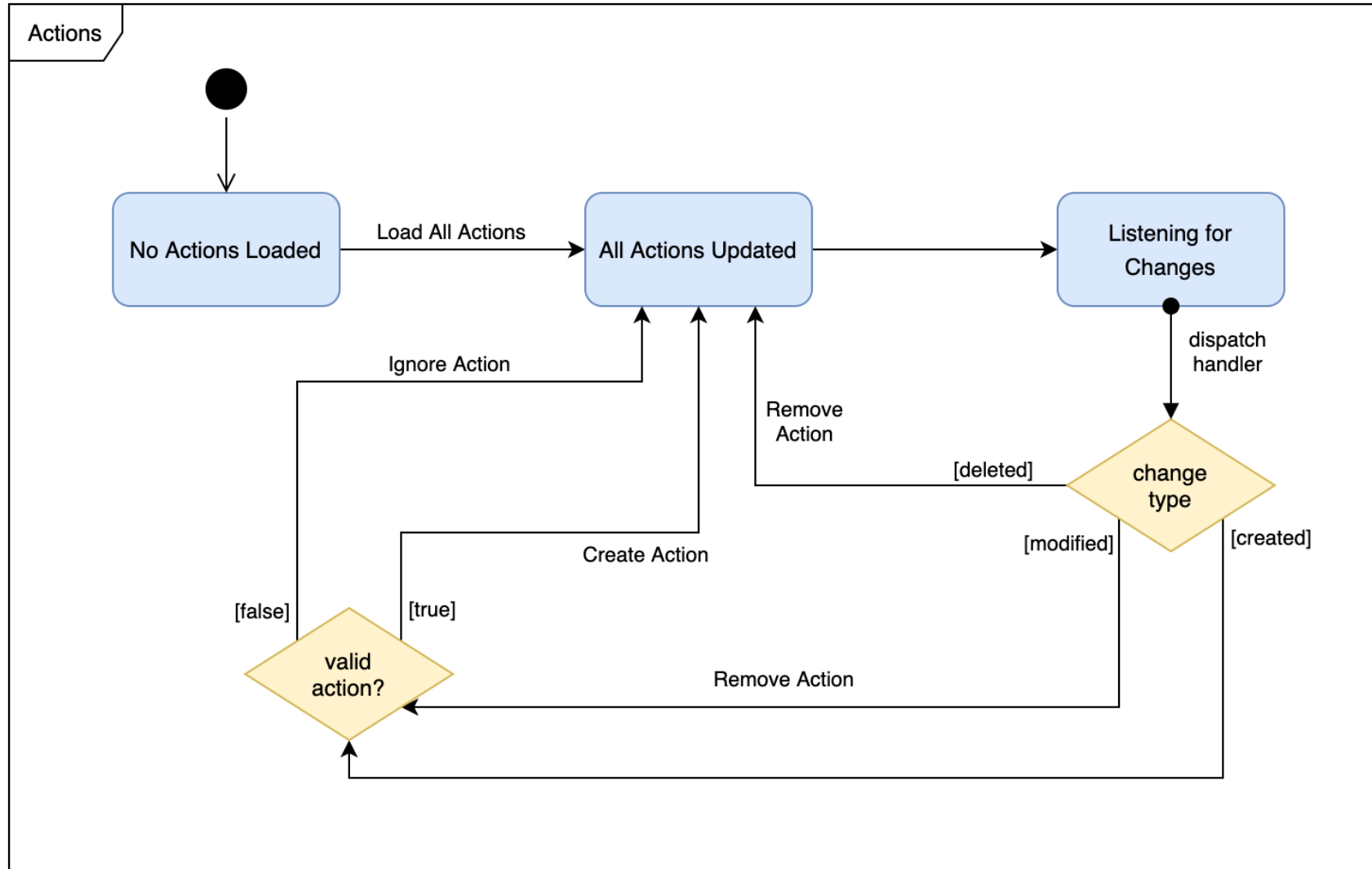
```
request: POST
url: http://localhost:3000/answer
arguments:
  - answer
```

parse_callback_arguments_from_transcript: |


```
import re


def parse(transcript: str):
    numbers: [int] = [int(s) for s in re.findall(r'\b\d+\b', transcript)]
    if len(numbers) > 0:
        return [numbers[0]]
    for (number, extensiveNumber) in extensiveNumbers:
        if extensiveNumber in transcript:
            return [number]
    raise ValueError('No number present in transcript "{}".format(transcript))
```

Actions Observer



Personal Voice Assistant? NO!

 Leonardo



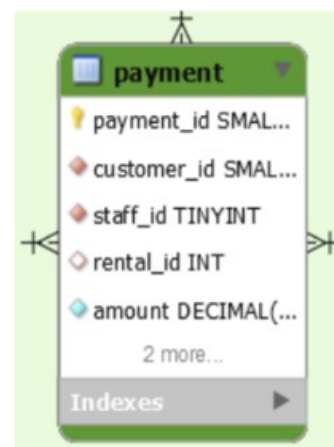
Studying

Question nº 1 | Level 2

Time left: 25

Consider the logical scheme presented in the figure. Which of the following queries allows you to indicate payments over '5' made by customers '1' and '5'?

- ☐ SELECT Customer_Id, Amount FROM Payment WHERE Customer_Id = 1 OR Id=5 AND Amount >= 5;
- ☐ SELECT Customer_Id, Amount FROM Payment WHERE Customer_Id IN (1,5) AND Amount > 5;
- ☒ SELECT DISTINCT Customer_Id, Amount FROM Payment WHERE Customer_Id IN (1,5) AND Amount > 5;
- ☐ SELECT Customer_Id, Amount FROM Payment WHERE Customer_Id IN (1,5) AND NOT <= 5;
- ☐ None of the above.



Resposta

Anterior

Seguinte

Terminar

Porquê

Explica

Como

Opinião

Estadísticas

Voz

Leonardo

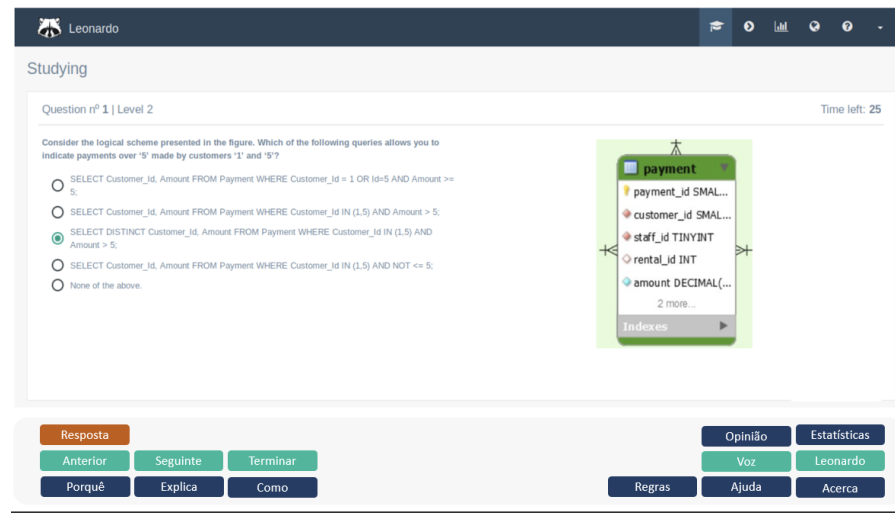
Regras

Ajuda

Acerca

Changes Needed:

- Use client microphone through browser
- Manage voice activation
- HTTP endpoint to process voice
- Change how callbacks for actions work
- Execute callback requests in the browser



...

callback:

request: POST

url: http://localhost:3000/answer

arguments:

- answer

...



Flask

web development,
one drop at a time

POST /voice { "audio": audio_blob }

SpeechRecognizer

ActionManager

Leonardo

Studying

Question nº 1 | Level 2Time left: 25

Consider the logical scheme presented in the figure. Which of the following queries allows you to indicate payments over '5' made by customers '1' and '5'?


☐ SELECT Customer_Id, Amount FROM Payment WHERE Customer_Id = 1 OR Id=5 AND Amount >= 5;

☐ SELECT Customer_Id, Amount FROM Payment WHERE Customer_Id IN (1,5) AND Amount > 5;

☒ SELECT DISTINCT Customer_Id, Amount FROM Payment WHERE Customer_Id IN (1,5) AND Amount > 5;

☐ SELECT Customer_Id, Amount FROM Payment WHERE Customer_Id IN (1,5) AND NOT <= 5;

☐ None of the above.



Resposta

Opinião

Estatísticas

Anterior

Seguinte

Terminar

Voz

Leonardo

Porquê

Explica

Como

Regras

Ajuda

Acerca

Implementation - 2

LEI - 2020

Voice recognition for education systems

Work Plan 04