

2nd Implementation

열려라, 쿠 (: K U) 굴

지도교수 서명

백승우

201411303 이준호

201511243 김동언

201511262 박우진

201714160 권혜주

INDEX

열려라, 쿠(:KU)굴

1. 프로젝트 소개 : 설명
2. 최종 예상 산출물
3. 2차 구현물
4. Demo Scenario
5. System Test & Result
6. Pass/Fail Criteria
7. Traceability
8. Component Diagram
9. 최종 발표 계획

프로젝트 소개 : 설명

'열려라, 쿠(:KU)굴'이란?

도둑 우두머리라는 화자 인식을 하지 않고 '열려라 참깨'라는 명령어만 처리한 결과
도둑질 당한 '알리바바와 40인의 도둑'에서 영감을 받았습니다.

열려라, 쿠(:KU)굴은 화자 인식이 된 상태에서 명령어 처리를 하도록 하는 프로그램입니다.

'열려라, 쿠(:KU)굴'이 작동하는 원리는?

마이크를 이용해 말을 하고,

첫번째 단계로 등록된 화자인지의 여부를 판단하고 결과값이 참이면
두번째 단계로 음성을 문자 데이터로 변환해 그에 맞는 명령을 수행합니다.

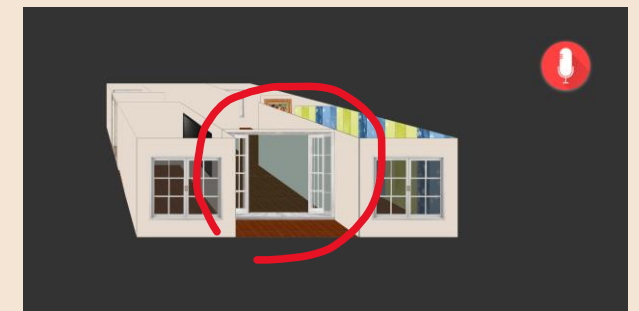
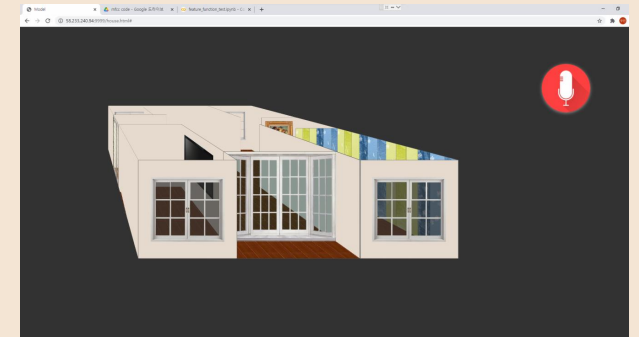
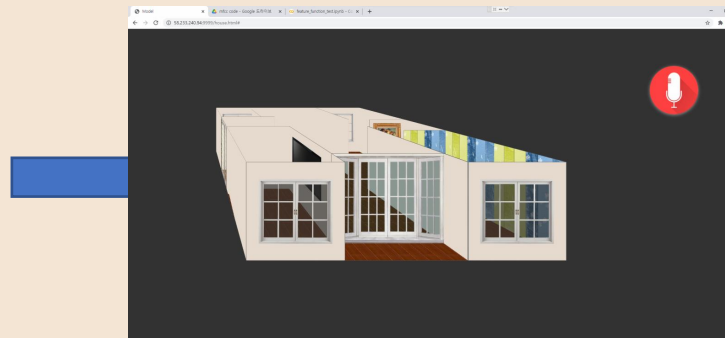
최종 예상 산출물



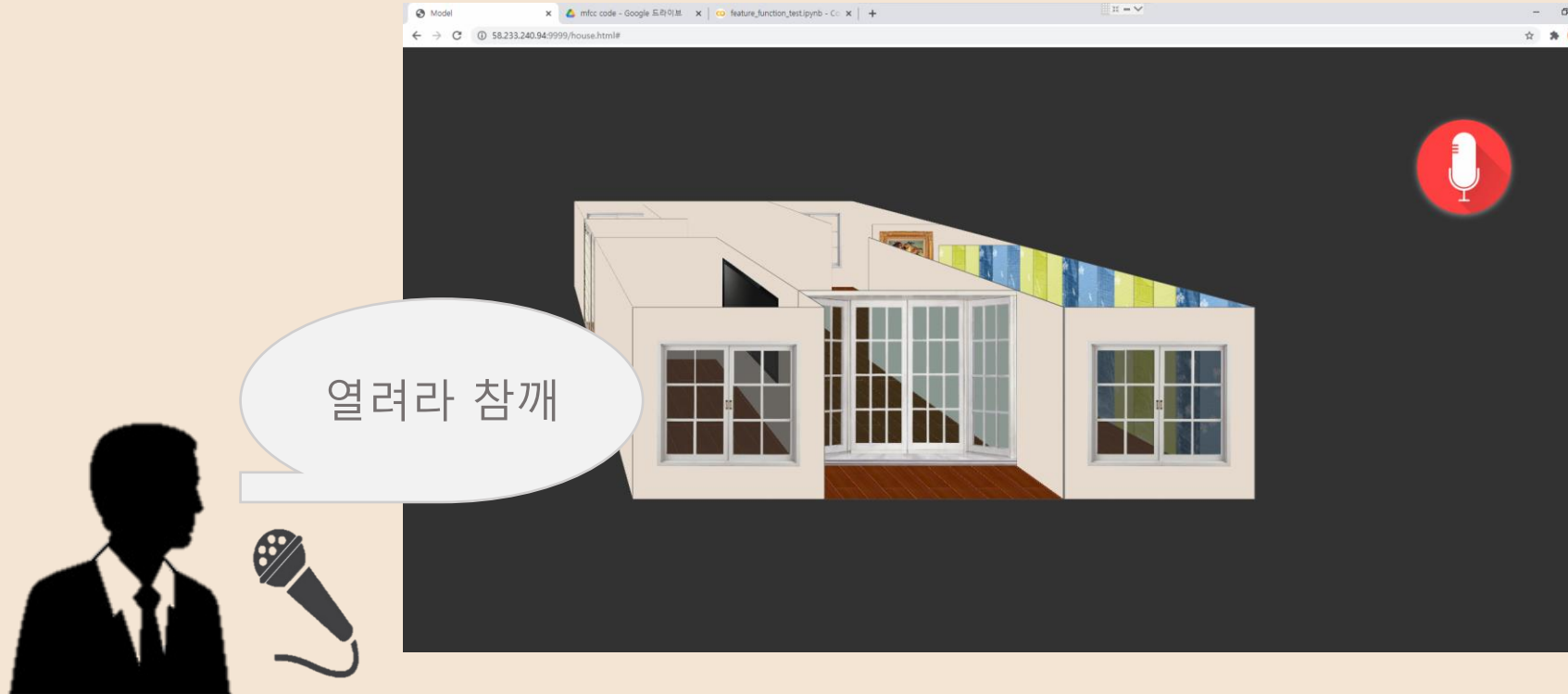
등록되지 않은 유저 A



등록된 유저 B

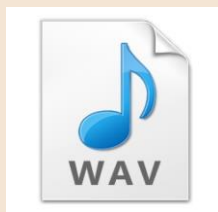


2차 구현물 : Web server



1. 사용자가 녹음 버튼을 클릭한다.
2. 사용자가 마이크를 통해 음성을 입력한다.

2차 구현물 : Feature Extraction

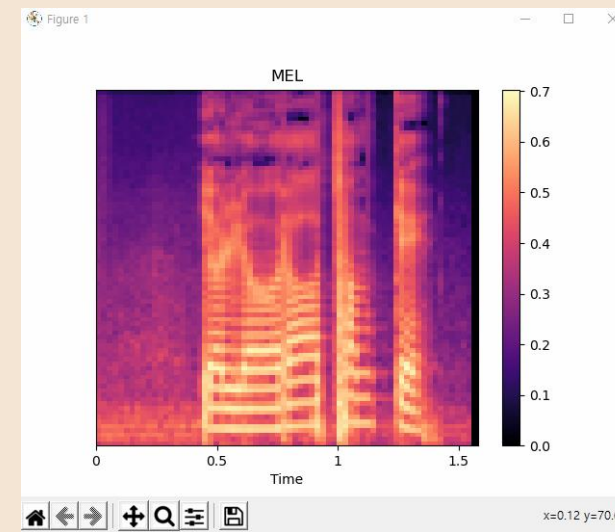
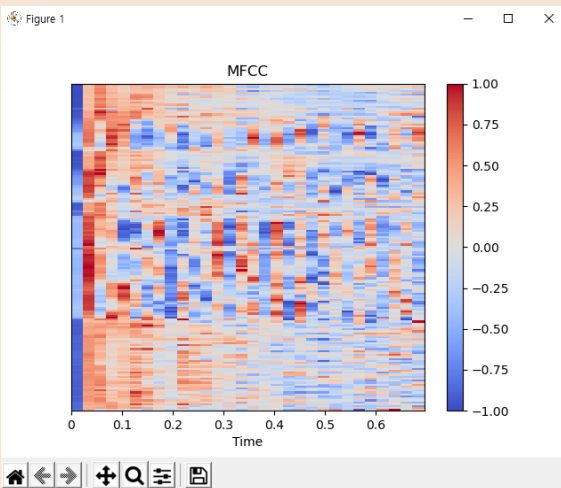


```
C:\Users\user\kugul\front\util>python extract.py
[[-0.7827202  0.3685689 -0.11142503 ...  1.          -0.00157098
  0.24949405]
 [-0.8623224  0.38415217  0.44046608 ...  0.0139421 -0.11064946
  0.49237686]
 [-0.8709392  0.34981114  0.3910621  ... -0.33645788  0.08340556
 -0.65462655]
 ...
 [-0.990838  0.30181062  0.61178553 ...  0.1703384  0.01317969
 -0.166747 ]
 [-0.9913872  0.29114956  0.549189  ... -0.00745713  0.16275284
  0.29610124]
 [-0.99754286  0.27697715  0.5902519 ...  0.07492517  0.06622265
 -0.06217406]]
```

```
C:\Users\user\kugul\front\util>python extract.py
[[0.37715  0.41797495  0.42734224 ...  0.37254304  0.34972984  0.          ]
 [0.45758447  0.4768083  0.49493617 ...  0.4313816  0.40550548  0.          ]
 [0.47561878  0.5026558  0.51294863 ...  0.4435901  0.4000197  0.          ]
 ...
 [0.19472913  0.1988987  0.18177691 ...  0.08500776  0.08955403  0.          ]
 [0.19078664  0.20060277  0.16999474 ...  0.08051071  0.08576365  0.          ]
 [0.14517975  0.15687037  0.12107296 ...  0.05216732  0.03568029  0.          ]]
```

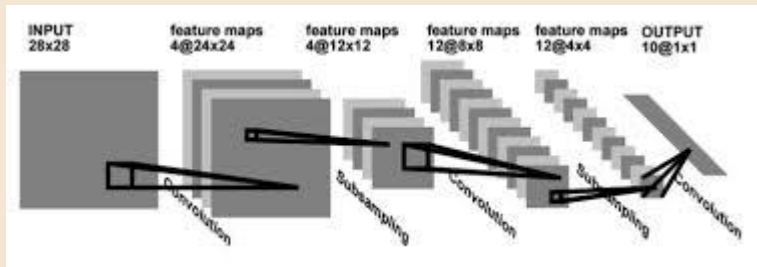


1. 서버에 업로드 된 음성파일의 특징 추출 방식으로 MFCC와 Mel spectrogram을 이용한다.



2차 구현물 : Predict (Test)

```
C:\Users\user\kugul\front\util>python extract.py
[[-0.7827202  0.3685689 -0.11142503 ...  1.          -0.00157098
  0.24949405]
 [-0.8623224  0.38415217  0.44046608 ...  0.0139421 -0.11064946
  0.49237686]
 [-0.8709392  0.34981114  0.3910621  ... -0.33645788  0.08340556
 -0.65462655]
 ...
 [-0.990838   0.30181062  0.61178553 ...  0.1703384  0.01317969
 -0.166747   ]
 [-0.9913872  0.29114956  0.549189   ... -0.00745713  0.16275284
  0.29610124]
 [-0.99754286  0.27697715  0.5902519  ...  0.07492517  0.06622265
 -0.06217406]]
```



SR

1. 학습된 SR 모델과, STT 모델을 각각 load한다.
2. 모델별로 변환된 음성 feature를 Predict(Test)하여, 각각의 결과를 반환한다.

```
C:\Users\user\kugul\front\util>python extract.py
[[0.37715    0.41797495  0.42734224 ...  0.37254304  0.34972984  0.
 ]
 [0.45758447  0.4768083  0.49493617 ...  0.4313816  0.40550548  0.
 ]
 [0.47561878  0.5026558  0.51294863 ...  0.4435901  0.4000197  0.
 ]
 ...
 [0.19472913  0.1988987  0.18177691 ...  0.08500776  0.08955403  0.
 ]
 [0.19078664  0.20060277  0.16999474 ...  0.08051071  0.08576365  0.
 ]
 [0.14517975  0.15687037  0.12107296 ...  0.05216732  0.03568029  0.
 ]]
```



Why CTC algorithm: no explicit alignment

CTC algorithm

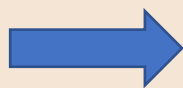
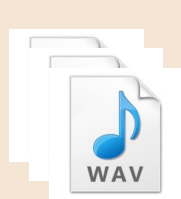
- 먼저 매 timestep마다 결과값을 예측한 후에

1. Collapse repeats
2. Remove blank tokens
3. Finish!!

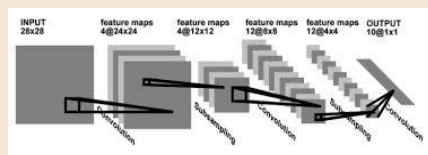


STT

2차 구현물 : Train



```
2020-09-07 21:43:12.161745: I tensorflow/stream_executor/platform/default/dso...
[[ 304.25276  137.15294  -38.741981  16.293688  -5.9188253
 2020-09-07 21:43:12.161745: I tensorflow/stream_executor/platform/default/dso...
[[ [-394.75276  127.15294  38.741981  -16.293688  5.9188253
 12.689705 ]
 [-596.4947  132.08122  98.59825  ... -14.441874  -38.486517
 -1.17066081 ]
 [-578.3329  117.72168  91.55144  ... -19.162182  -32.431872
 8.8941346 ]
 ...
 [-737.3877  182.25579  43.449886  ... -17.748533  -19.267479
 -8.789853 ]
 [-738.3183  181.98717  45.86222  ... -14.985735  -21.652264
 -22.662487 ]
 [-789.28284  117.83282  32.86899  ... -6.969739  -13.427218
 -6.889418 ]]
```



SR Model

Why CTC algorithm: no explicit alignment

CTC algorithm
- 먼저 매 timestep마다 결과값을 예측한 후에

1. Collapse repeats
2. Remove blank tokens
3. Finish!!

hello
h h e e l l l l l l o
h e x e l i o
h e l l l l o
hello Output

STT Model

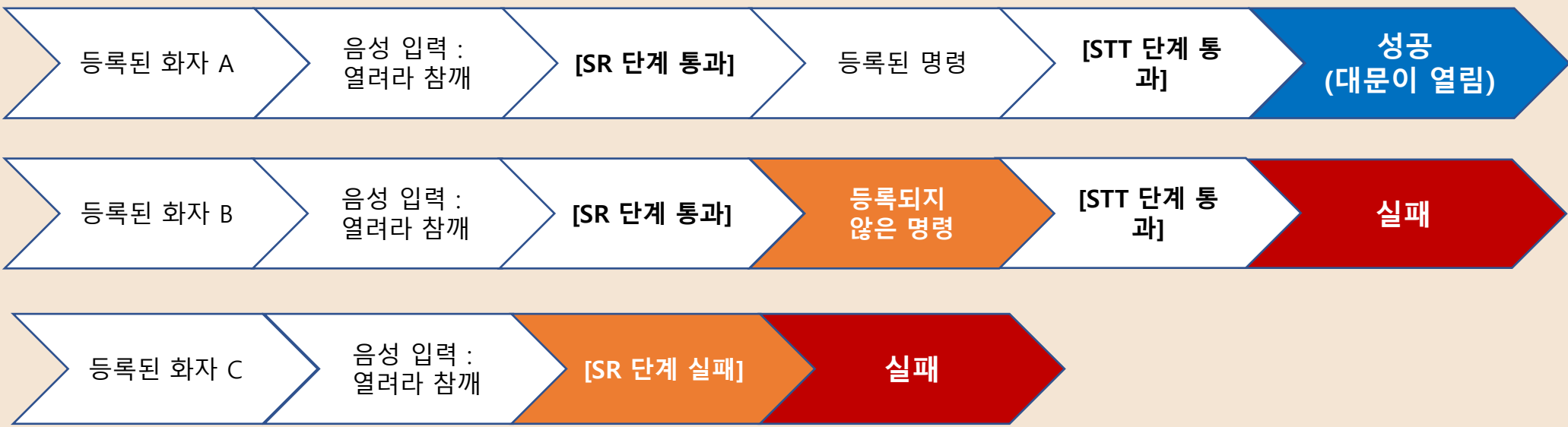
```
model
├── checkpoint
├── train_model.ckpt.data-00000-of-00001
├── train_model.ckpt.index
└── train_model.ckpt.meta
```



```
model
├── checkpoint
├── train_model.ckpt.data-00000-of-00001
├── train_model.ckpt.index
└── train_model.ckpt.meta
```


Demo Scenario

1. 등록된 화자일 경우



Demo Scenario

2. 등록되지 않은 화자일 경우



System Test & Result

ID	Description	Expected Result (1: Success, 2: Fail)	Result (○,△,X)
1.1	마이크가 입력 가능한 상태인지 테스트한다.	1. No Message 2. Error Message	○
1.2	사용자의 녹음 진행을 테스트한다.	1. No Message 2. Error Message	○
1.3	사용자의 음성 파일을 서버로 전송 후 .wav 파일로 변화되는지 테스트한다.	1. 사용자 음성 파일(wav format) 2. Error Message	○
2	음성 파일의 전처리 수행을 테스트한다.	1. Numpy file 2. Error Message	○
3	음성을 텍스트로 변환하여 출력이 되는지 테스트한다. (STT)	1. 음성의 문장과 동일한 텍스트 2. 음성의 문장과 동일하지 않은 텍스트	○
4	음성이 등록된 화자의 것인지 테스트한다. (SR)	1. 학습된 모델에 등록된 화자의 라벨 및 일치 확률 2. Error Message	○
5	SR 결과와 STT 결과가 데이터베이스에 등록되어 있는지 테스트한다.	1. 등록된 화자와 기능입니다 2. 등록되지 않은 화자 또는 기능입니다	○

System Test & Result

ID	Description	Expected Result (1: Success, 2: Fail)	Result (○,△,X)
6.1	[STT] 학습 파일을 이용하여 라벨링 된 테스트 결과와 같은지 테스트한다.	1. 음성의 문장과 동일한 텍스트 2. Error Message	○
6.2	학습 파일이 저장 되었는지 테스트한다.	1. 학습 파일 2. Error Message	○
7.1	[SR] 학습 파일을 이용하여 라벨링 된 테스트 결과와 같은지 테스트한다.	1. 학습된 모델에 등록된 화자의 라벨 및 일치 확률 2. Error Message	○
7.2	학습 파일이 저장 되었는지 테스트한다.	1. 학습 파일 2. Error Message	○

System Test & Result

ID	Description	Expected Result (1: Success, 2: Fail)	Result (○,△,X)
8.1	시스템이 5회 이내 인식을 하는지 테스트한다.	1. 등록된 사용자 중 특정인의 값과 몇 퍼센트 동일하다고 수치가 뜬 2. Error Message("등록되지 않은 사용자입니다")	○
8.2	사용자 음성의 문자 데이터와 STT에 적용한 결과가 일치하는지 테스트한다.	1. Success Message 2. Error Message	○
9	등록되어 있는 기능이 수행 되는지 테스트한다.	1. Success Message 2. Error Message	○
10	STT, SR 수행 후 데이터를 파기하고 오류 없이 완료 되는지 테스트한다.	1. Success Message 2. Error Message	○
11	특정 OS에 국한되지 않고 모든 OS의 Chrome Browser에서 실행되는지 테스트한다.	1. Function Execution Result Message 2. Error Message	○

Pass / Fail Criteria - Functional Requirement

ID	Description	Pass / Fail Criteria	Pass / Fail
No 1.1	Microphone Check Test	마이크가 연결 상태이고 입력가능한 상태이어야 한다	Pass
No 1.2	Voice Recording Test	사용자의 음성 녹음이 진행되어야 한다.	Pass
No 1.3	File Transfer Test	사용자의 음성파일이 WAV 파일 형식으로 저장 되어야한다.	Pass
No 2.1	Voice Pre-processing Test	입력 받은 음성의 특징 정보를 추출하고 이를 Numpy 파일 형식으로 저장이 되어야 한다.	Pass
No 3.1	STT Execution Test	음성의 문장과 동일한 텍스트가 90% 이상 나오면 성공	Pass
No 4.1	SR Execution Test (Authorized User)	등록된 화자는 일치 확률(80% 이상)이 나오면 성공	Pass
No 4.2	SR Execution Test (Unauthorized User)	등록되지 않은 화자는 실패 결과가 나오면 성공	Pass

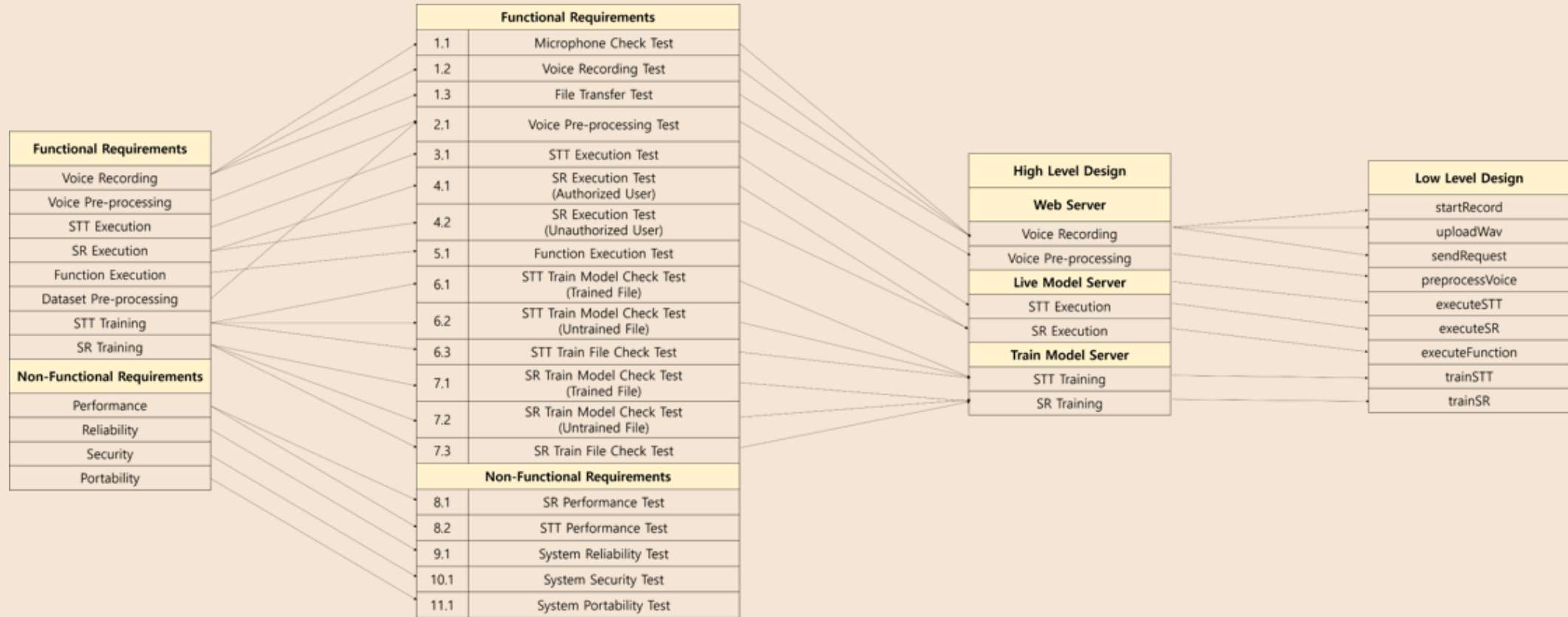
Pass / Fail Criteria - Functional Requirement

ID	Description	Pass / Fail Criteria	Pass / Fail
No 5.1	Function Execution Test	SR결과와 STT 결과값이 DB에 있는 경우 성공	Pass
No 6.1	STT Train Model Check Test (Trained File)	음성의 문장과 동일한 텍스트가 90% 이상 나오면 성공	Pass
No 6.2	STT Train Model Check Test (Untrained File)	음성의 문장과 동일한 텍스트가 90% 이상 나오면 성공	Pass
No 6.3	STT Train File Check Test	학습 파일이 저장되어 있다면 성공	Pass
No 7.1	SR Train Model Check Test (Trained File)	등록된 화자는 일치 확률(80% 이상)이 나오면 성공	Pass
No 7.2	SR Train Model Check Test (Untrained File)	화자의 등록 여부를 판단한다. 등록된 화자는 일치 확률(80% 이상)이 나오면 성공 등록되지 않은 화자는 실패 결과가 나오면 성공	Pass
No 7.3	SR Train File Check Test	학습 파일이 저장되어 있다면 성공	Pass

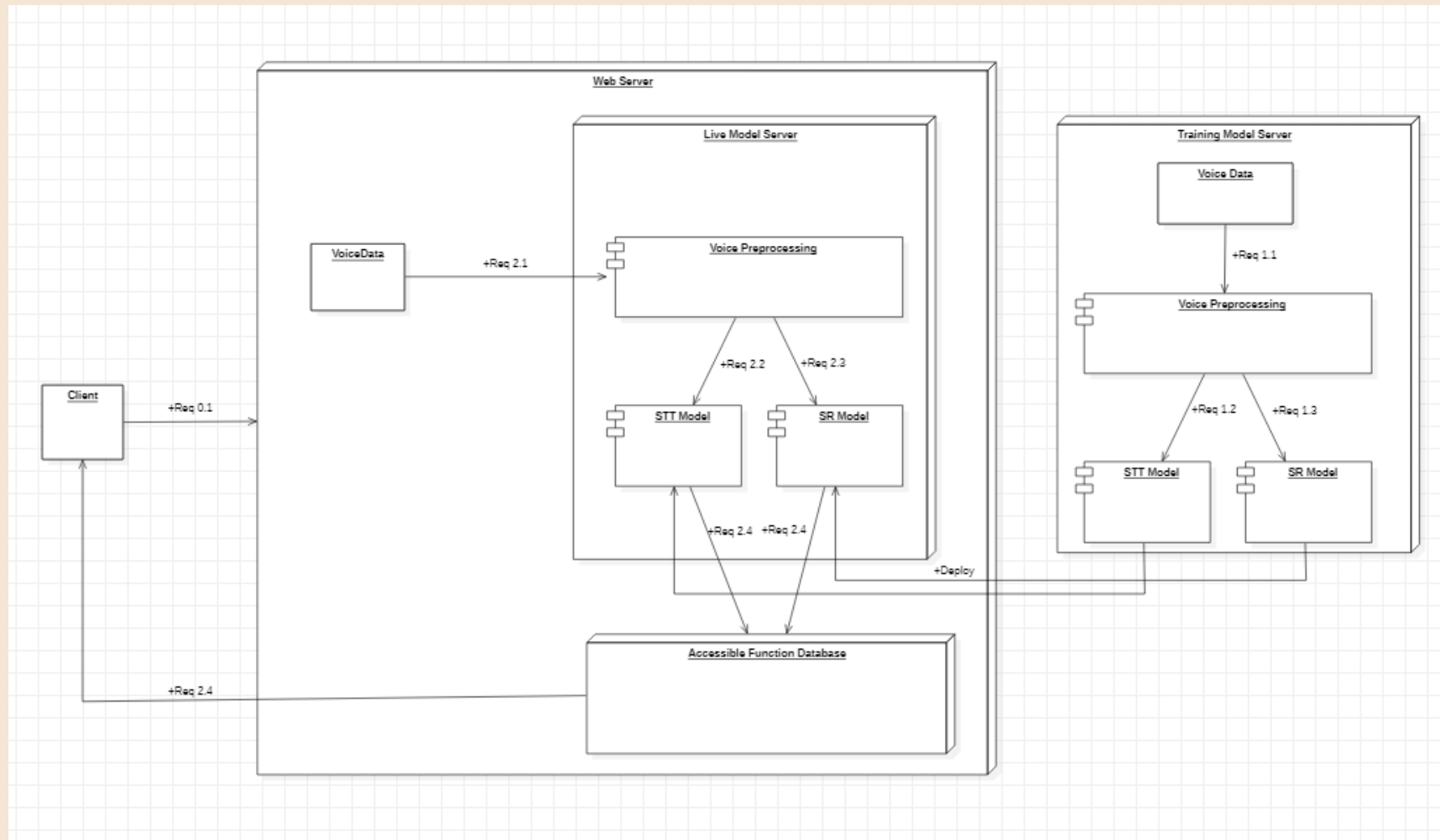
Pass / Fail Criteria - Non-Functional Requirement

ID	Description	Pass / Fail Criteria	Pass / Fail
No 8.1	SR Performance Test	5회 이내 정확한 결과를 도출하면 성공	Pass
No 8.2	STT Performance Test	음성의 문장과 동일한 텍스트가 90% 이상 나오면 성공	Pass
No 9.1	System Reliability Test	STT 결과값과 수행한 동작이 동일하면 성공	Pass
No 10.1	System Security Test	음성 파일이 삭제 되면 성공	Pass
No 11.1	System Portability Test	각 단계마다 오류 없이 수행이 된다면 성공 (오류 발생시만 메시지 창이 뜬)	Pass

Traceability



Component Diagram



최종 발표 계획

앞으로의 계획

마감기한까지 학습과 평가를 반복 진행하여 정확도를 더 높일 수 있는 모델을 위해 계속 테스트 할 예정

한계점

등록되지 않은 화자지만 등록된 목소리와 유사해 SR 단계를 성공할 경우가 발생할 가능성이 있다

