

상담을 위한 사용자의 답변 적극성과 일관성 판단 및  
발화 생성 서비스  
설계서

(요구기능: 질문에 답변하는 사용자의 적극성 판단)

(요구기능 ID: CP\_001)

(참조 파일명:

[종합설계1\\_02\\_두리아\\_CP001\\_요구분석정의서.docx](#))

---

문서번호 :

Version 0.1

## 개정 이력

### 제.개정내역

버전	승인일자	개요	작성자

### 배포이력

버전	배포일자	배포처

### 검토이력

버전	검토일자	검토방법	검토자

## 목 차

1.	요구기능 설명 .....	3
1.1.	주요 상세기능 설명.....	4
1.2.	요구기능 동작 절차.....	4
1.3.	동작 규칙.....	5
1.4.	가정(Assumptions).....	6
2.	기술설계 #1.....	6
2.1.	기능 설계.....	6
2.2.	요소 설계.....	10
2.2.1.	Process.....	10
2.2.2.	Data.....	14
2.2.3.	저장소.....	15
3.	기타사항.....	17

## 1. 요구기능 설명

### 제공하는 주요 상세 기능

1. 적극성 평가: 사용자가 답변할 때, 얼마나 적극성을 가지고 질문에 답변하는지 평가한다.

### 기능의 동작 원리

1. 자연어 처리 (NLP) 기법 활용: 이 기능은 사용자의 답변을 분석하기 위해 자연어 처리 기술을 사용한다. 특히, 답변이 가지는 내면의 의미를 파악한다.
2. 적극성 점수 모델링: 사용자의 답변을 분석하여 적극성 점수를 계산. 이는 음성 데이터 분석, 유사성 분석, 키워드 일치 여부, 문맥적 정확성을 포함한 다양한 요소를 기반으로 한다.
3. 기계 학습: 시간이 지남에 따라 수집되는 데이터를 통해 모델을 지속적으로 학습시켜, 판단 기준을 최적화하고 정확도를 개선한다.

### 동작 조건

1. 데이터 입력: 사용자의 답변이 음성 형식으로 입력되어야 하며, 음성 데이터를 텍스트로 변환에 정확성이 필요하다.
2. 정확한 NLP 도구 선택: 한국어에 적합한 NLP 도구와 알고리즘이 필요하다.

### 주요 처리 대상

1. 사용자 답변: 사용자가 입력한 음성 데이터와 이를 텍스트로 변환한 데이터

2. 질문 데이터: 상담 질문의 내용 및 의도가 담긴 데이터

## **처리 결과**

1. 적극성 점수: 각 사용자 답변에 대한 적극성을 수치로 표현한 결과. 이 점수는 답변의 관련성 및 완전성을 반영한다.

### **1.1. 주요 상세기능 설명**

#### **- ID: FJ\_001**

- 정의: 주요 상세 기능 #1 은 사용자의 답변 적극성을 평가하는 데 초점을 맞춘다. 이 기능은 사용자가 질문에 어떻게 반응하고, 얼마나 열정적으로 참여하는지 측정하여 그 결과를 평가 점수로 제시한다.

- 상세 설명: 주요 상세 기능 #1 은 사용자의 답변에 나타나는 열정과 참여도를 분석하는 다양한 요소를 평가한다. 이는 답변 음성데이터의 양과 질, 응답 속도, 사용된 어휘와 표현의 다양성 및 전문성을 포함한다. 이러한 측정을 통해, 사용자가 주어진 질문에 얼마나 적극적으로 대응하는지를 종합적으로 평가하며, 이를 기반으로 개인의 답변 스타일과 적극성을 파악한다.

### **1.2. 요구기능 동작 절차**

단계 1: 질문을 통해 얻은 사용자의 답변을 바탕으로 음성 데이터 수집

단계 2: 음성 데이터를 분석하여 적극성 판단

### **[단계별 상세 내용]**

단계 1: 사용자는 시스템을 통해 상담 질문에 답변을 한다. 이 단계에서 시스템은 사용자의 답변을 수집하고 저장한다.

단계 2: 시스템은 수집된 답변을 분석하여 사용자가 얼마나 적극적으로 질문에 응답했는지를 평가한다. 이 평가는 사용자 답변의 내용, 양, 사용된 어휘와 문장 구성, 응답의 신속성 등을 포함한 다양한 기준에 따라 이루어진다. 평가 결과는 적극성 점수로 표현되며, 이는 사용자의 참여도와 질문에 대한 이해도를 반영한다.

### **1.3. 동작 규칙**

#### **동작 규칙 1: 데이터 보안과 프라이버시 준수**

사용자로부터 수집된 모든 데이터는 데이터 보호 규정과 프라이버시 정책에 따라 처리되어야 한다. 이는 사용자의 신뢰를 유지하고 법적 문제를 예방하는 데 중요하다. 데이터는 암호화되어 안전하게 저장되며, 오직 승인된 시스템과 인력만이 이에 접근할 수 있어야 한다.

#### **동작 규칙 2: 적응형 학습 절차의 적용**

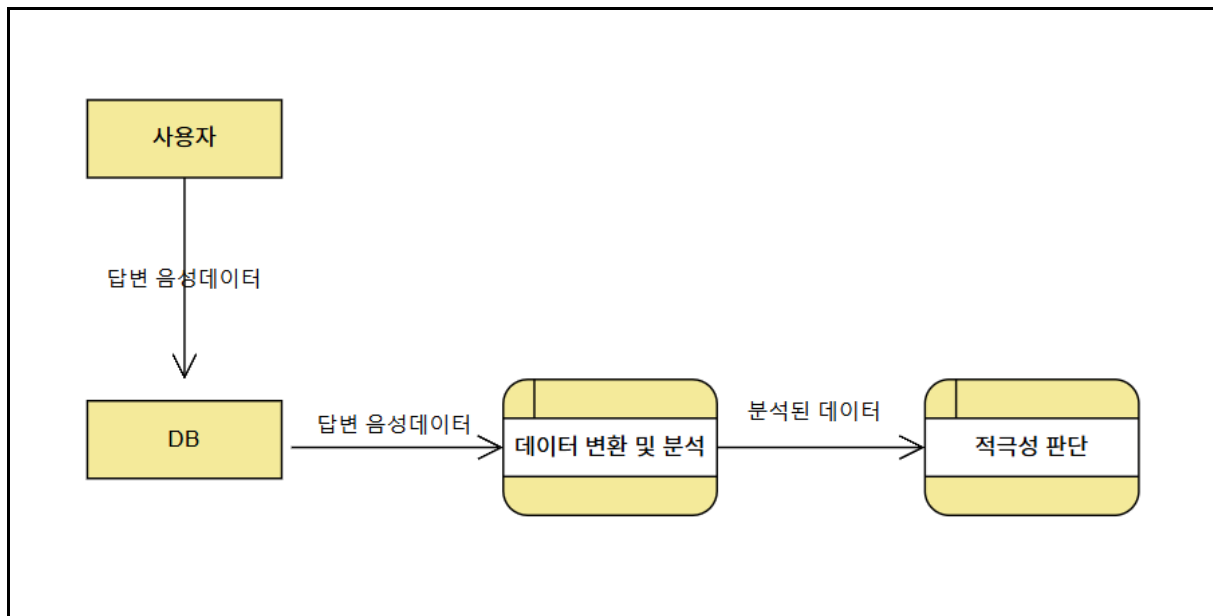
시스템은 수집된 데이터를 기반으로 지속적인 학습을 수행하며, 이를 통해 평가 기준을 자동으로 조정할 수 있어야 한다. 이는 머신 러닝 알고리즘의 적응형 학습 기능을 활용하여 이루어지며, 다양한 사용자 행동과 반응을 학습함으로써 시스템의 정확성과 유연성을 강화한다.

## 1.4. 가정(Assumptions)

- 상담은 정보보호수집 동의서에 동의를 거쳐야 한다.
- 사용자가 상담을 거절할 경우 상담을 유도하는 것이 아닌 상담을 종료한다.
- 음성 데이터의 텍스트화가 원활하게 진행되어야 한다.
- 음성 데이터의 개인 정보가 유출되지 않게 잘 보호되어야 한다.

## 2. 기술설계 #1

### 2.1. 기술 설계



1. 사용자의 음성 데이터를 DB에 저장한다.

```
@app.route('/upload_voice', methods=['POST'])
def upload_voice():
    user_id = request.form['user_id']
    voice_data = request.files['voice_data'].read()
    timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
```

```

conn = sqlite3.connect('voice_data.db')
cursor = conn.cursor()
cursor.execute("""INSERT INTO VoiceData (user_id, timestamp, voice_data) VALUES (?, ?, ?)""",
(user_id, timestamp, voice_data))
conn.commit()
conn.close()

return jsonify({'status': 'success'}), 200

```

- user\_id: 사용자의 고유 ID, 사용자 식별에 사용
- voice\_data: 사용자가 업로드한 음성 데이터 파일, 바이너리 형태로 저장
- timestamp: 음성 데이터가 업로드된 시점을 기록
- conn: SQLite 데이터베이스와의 연결 객체
- cursor: 데이터베이스 작업을 수행하는 커서 객체

## 2. 서버에서 음성 데이터 요청 및 전송한다.

```

@app.route('/request_voice/<user_id>', methods=['GET'])
def request_voice(user_id):
    conn = sqlite3.connect('voice_data.db')
    cursor = conn.cursor()
    cursor.execute("""SELECT voice_data FROM VoiceData WHERE user_id = ?""", (user_id,))
    data = cursor.fetchone()
    conn.close()

    if data:
        return jsonify({'voice_data': data[0]}), 200
    else:
        return jsonify({'error': 'No data found'}), 404

```



### 3. 네이버 STT API 를 활용하여 음성 데이터를 텍스트 데이터로 변환한다.

```
def stt_naver(voice_data):
    url = "https://naveropenapi.apigw.ntruss.com/recog/v1/stt?lang=Kor"
    headers = {
        "Content-Type": "application/octet-stream",
        "API-KEY-ID": CLIENT_ID,
        "API-KEY": CLIENT_SECRET
    }

    response = requests.post(url, headers=headers, data=voice_data)
    if response.status_code == 200:
        response_body = json.loads(response.text)
        return response_body['text']
    else:
        print("Error:", response.text)
        return None
```

- voice\_data: 이전에 설명된 데이터로 STT API 로 전송
- url: 네이버 STT API URL, 음성 데이터를 텍스트로 변환하기 위해 요청을 보낼 주소
- headers: 요청 헤더로, API 인증 정보를 포함
- Content-Type: 요청 데이터의 형식을 지정, application/octet-stream 으로, 바이너리 데이터를 의미
- API-KEY-ID: 네이버 클라이언트 ID
- API-KEY: 네이버 클라이언트 시크릿 키
- response: 네이버 STT API 로부터의 응답 객체, 응답 상태 코드와 응답 데이터
- response\_body: JSON 형식으로 파싱된 응답 데이터
- response\_body['text']: 변환된 텍스트 데이터를 포함하는 키

#### 4. 텍스트화 된 데이터를 NLP 분석을 실시한다.

```
def analyze_activeness(voice_data):
    text = stt_naver(voice_data)
    if text is None:
        return None

    # NLP 분석
    vectorizer = joblib.load('vectorizer.pkl')
    X = vectorizer.transform([text])
    model = joblib.load('activeness_model.pkl')
    score = model.predict_proba(X)[0][1]

    return score
```

- text: 네이버 STT API 가 사용된 텍스트 데이터
- vectorizer: 사전 학습 텍스트 벡터화 도구. 텍스트 데이터를 벡터로 변환
- X: 변환된 텍스트 데이터의 벡터 표현
- model: 적극성 예측 모델
- score: 적극성 점수

#### 5. 분석된 데이터를 기반으로 적극성을 평가한다

```
@app.route('/analyze_voice/<user_id>', methods=['GET'])
def analyze_voice(user_id):
    conn = sqlite3.connect('voice_data.db')
    cursor = conn.cursor()
    cursor.execute("SELECT voice_data FROM VoiceData WHERE user_id = ?", (user_id,))
    data = cursor.fetchone()

    if data:
        voice_data = data[0]
```

```

    activeness_score = analyze_activeness(voice_data)
    if activeness_score is not None:
        cursor.execute("""UPDATE VoiceData SET activeness_score = ? WHERE user_id = ?""",
(activeness_score, user_id))
        conn.commit()
        conn.close()
        return jsonify({'activeness_score': activeness_score}), 200
    else:
        conn.close()
        return jsonify({'error': 'STT failed'}), 500
else:
    conn.close()
    return jsonify({'error': 'No data found'}), 404

```

- voice\_data: 데이터베이스에서 조회한 음성 데이터
- activeness\_score: 음성 데이터 분석을 통해 계산된 적극성 점수

## 2.2. 요소 설계

### 2.2.1. Process

프로세스 1. 사용자 ID 와 음성 데이터 추출

#### 1. 사용자 ID 와 음성 데이터 추출

- `user\_id = request.form['user\_id']`: 데이터에서 사용자 ID 를 추출
- `voice\_data = request.files['voice\_data'].read()`: 파일 데이터에서 음성 데이터를 바이너리 형태로 읽음

#### 2. 데이터베이스에 데이터 삽입

- ``cursor.execute("INSERT INTO VoiceData (user_id, timestamp, voice_data) VALUES (?, ?, ?)", (user_id, timestamp, voice_data))``: SQL INSERT 문을 실행하여 사용자 ID, 타임스탬프, 음성 데이터를 데이터베이스에 삽입

## 프로세스 2. 서버에서 음성 데이터 요청 및 전송

### 1. 데이터베이스 조회

- `cursor.execute("SELECT voice_data FROM VoiceData WHERE user_id = ?", (user_id,))`: 이 SQL SELECT 문은 특정 사용자 ID에 해당하는 음성 데이터를 조회하는 데 사용된다. 데이터베이스에서 특정 데이터를 검색하여 가져온다.

### 2. 조회 결과 처리 및 응답 반환

- `data = cursor.fetchone()`: 조회된 결과를 가져오고, 결과가 있는지 확인한다.
- `if data: return jsonify({'voice_data': data[0]})`, 200: 조회된 데이터가 있을 경우 이를 JSON 형식으로 반환한다.

## 프로세스 3. 네이버 STT API를 활용하여 음성 데이터를 텍스트 데이터로 변환

### 1. 네이버 STT API 호출

- `response = requests.post(url, headers=headers, data=voice_data)`: 네이버 STT API로 POST 요청을 보내 음성 데이터를 텍스트로 변환. 이 메소드는 API 호출을 통해 음성 데이터를 처리하고 변환된 텍스트를 반환받는 핵심 작업을 수행한다.

### 2. 응답 처리

- if response.status\_code == 200:: API 호출이 성공했는지 확인하는 코드. 상태 코드 200 은 요청이 성공했음을 의미.
- response\_body = json.loads(response.text): 응답 데이터를 JSON 형식으로 파싱.
- return response\_body['text']: 변환된 텍스트 데이터를 반환 텍스트로 변환된 최종 결과물이다.

#### 프로세스 4. 텍스트화 된 데이터를 NLP 분석을 실시

##### 1. 음성 데이터를 텍스트로 변환

- text = stt\_naver(voice\_data): 네이버 STT API 를 호출하여 음성 데이터를 텍스트로 변환

##### 2. NLP 분석 및 적극성 점수 예측:

- vectorizer = joblib.load('vectorizer.pkl'): 사전 학습된 벡터라이저를 로드
- X = vectorizer.transform([text]): 변환된 텍스트 데이터를 벡터로 변환
- model = joblib.load('activeness\_model.pkl'): 사전 학습된 적극성 예측 모델을 로드
- score = model.predict\_proba(X)[0][1]: 벡터화된 텍스트 데이터를 사용하여 모델이 적극성 점수를 예측

## 프로세스 5. 분석된 데이터를 기반으로 적극성을 평가

### 1. 데이터베이스 조회

- `cursor.execute("""SELECT voice_data FROM VoiceData WHERE user_id = ?""`,  
`(user_id))`: SQL SELECT 문을 실행하여 특정 사용자 ID 에 해당하는 음성 데이터를  
조회한다.
- `data = cursor.fetchone()`: 조회된 결과 중 첫 번째 행을 가져온다. 결과가 없으면  
None 을 반환.

### 2. 음성 데이터 분석 및 적극성 점수 계산

- `voice_data = data[0]`: 조회된 음성 데이터를 추출한다.
- `activeness_score = analyze_activeness(voice_data)`: `analyze_activeness` 함수를  
호출하여 음성 데이터를 텍스트로 변환하고, NLP 모델을 통해 적극성 점수를 계산.

### 3. 데이터베이스 업데이트 및 응답 반환

- `cursor.execute("""UPDATE VoiceData SET activeness_score = ? WHERE user_id = ?""`,  
`(activeness_score, user_id))`: SQL UPDATE 문을 실행하여 사용자 ID 에 해당하는  
레코드의 적극성 점수를 업데이트.
- `conn.commit()`: 변경사항을 데이터베이스에 저장.
- `return jsonify({'activeness_score': activeness_score})`, 200: 성공 응답을 JSON  
형식으로 반환.

## 2.2.2. Data

이름	타입	범위	출력 프로세스	입력 프로세스	비고
user_id	문자열	사용자의 고유 ID	분석 요청 응답 반환	음성 데이터 업로 드, 분석 요청	사용자를 식 별하는 고유 ID
voice_data	바이너리	음성 파일 데이터	분석 요청	음성 데이터 업로 드, 분석 요청	사용자의 음 성 데이터
timestamp	문자열	yyyy-mm- dd hh:mm:ss	음성 데이터 업 로드 응답 반환	음성 데이터 업로 드	음성 데이터 업로드 시간
activeness _score	실수	0.0 ~ 1.0	분석 요청 응답 반환	음성 데이터 분석	NLP 분석 결과
response	JSON 객체	-	음성 데이터 업 로드 응답 반환, 분석 요청 응답 반환	음성 데이터 업로 드, 분석 요청	클라이언트 로 반환되는 응답
conn	데이터베이스 연결 객체	-	모든 데이터베이 스 관련 프로세 스	모든 데이터베이 스 관련 프로세스	SQLite 연결 객체
cursor	데이터베이스 커서 객체	-	모든 데이터베이 스 관련 프로세 스	모든 데이터베이 스 관련 프로세스	SQLite 커서 객체
text	문자열	음성 데이 터의 텍스 트 변환 결 과	NLP 분석	음성 데이터 분석	네이버 STT API 결과
url	문자열	네이버 STT API 엔 드포인트	음성 데이터 분 석	음성 데이터 분석	네이버 STT API 호출 URL
headers	사전	API 요청 헤더	음성 데이터 분 석	음성 데이터 분석	네이버 STT API 인증 정 보
response _body	JSON 객체	-	음성 데이터 분 석	음성 데이터 분석	네이버 STT API 응답
X	희소 행렬	-	NLP 분석	음성 데이터 분석	NLP 벡터화

					결과
vectorizer	객체	-	NLP 분석	음성 데이터 분석	사전 학습된 벡터라이저
model	객체	-	NLP 분석	음성 데이터 분석	사전 학습된 모델

### 2.2.3. 저장소

저장소 이름	목적	사용 프로세스	구성 요소
사용자 정보 데이터베이스 (UserDB)	사용자 개인 정보 및 인증 데이터 저장	음성 데이터 업로드, 분석 요청, 결과 저장	Users 테이블
음성 데이터 메타정보 데이터베이스 (VoiceMetaDB)	음성 데이터 메타정보 저장	음성 데이터 수집, 분석 요청	VoiceMeta 테이블
음성 데이터 저장소 (VoiceDataStorage)	사용자 입력 음성 데이터 안전하게 저장	음성 데이터 수집, 분석 요청	파일 시스템
음성 분석 데이터베이스 (VoiceAnalysisDB)	음성 데이터 분석 결과 저장	분석 결과 저장, 피드백 제공	AnalysisResults 테이블

#### 1. 사용자 정보 데이터베이스 (UserDB)

목적: 사용자 개인 정보 및 인증 데이터 저장

구성 요소: Users 테이블:

- user\_id (문자열, PRIMARY KEY): 사용자 고유 ID
- username (문자열): 사용자 이름



- password (문자열): 사용자 암호

- email (문자열): 사용자 이메일

## 2. 음성 데이터 메타정보 데이터베이스 (VoiceMetaDB)

목적: 음성 데이터 메타정보(경로, 업로드 시간 등) 저장

구성 요소: VoiceMeta 테이블:

- user\_id (문자열, FOREIGN KEY REFERENCES UserDB.Users(user\_id)): 사용자 고유 ID

- file\_path (문자열): 음성 데이터 파일 경로

- upload\_time (문자열, yyyy-mm-dd hh:mm:ss): 업로드 시점의 타임스탬프

## 3. 음성 데이터 저장소 (VoiceDataStorage)

목적: 음성 파일 데이터 저장

구성 요소: 파일 시스템 - 음성 데이터 파일 저장

## 4. 음성 분석 데이터베이스 (VoiceAnalysisDB)

목적: 음성 데이터 분석 결과 저장

구성 요소: AnalysisResults 테이블:

- user\_id (문자열, FOREIGN KEY REFERENCES UserDB.Users(user\_id)): 사용자 고유 ID

- timestamp (문자열, yyyy-mm-dd hh:mm:ss): 분석 시점의 타임스탬프

- activeness\_score (실수): 분석된 적극성 점수

### 3. 기타사항

FJ\_001을 통해 사용자와의 답변을 바탕으로 적극성 점수를 수집하고 이 점수를 바탕으로 CP\_003의 발화 생성 알고리즘을 사용하여 사용자와 대화를 지속한다.