

[Q]

안녕! 지금까지 네가 잘 도와준 덕분에, 'Phase 1'과 'Phase 2'를 성공적으로 완성할 수 있었어. 정말 고마워. 지금부터는 'Phase 3'에 해당하는 작업을 시작하려고 해.

Phase 3에서는, 우리팀이 그동안 Phase 1과 Phase 2에서 진행한 핵심적인 기술 흐름(즉, STT-Diarization-Speaker Tagging[2: Name-Tagging vs. Nickname-Tagging] 흐름)을 본질이자 중심으로 견지하고 사수하면서도, 이것을 바탕으로 회의 상황에서 유익하게 활용할 수 있는 회의 상황 전용 툴(도구), 회의 상황 전용 서비스로 나아가고자 해.

해당 툴/서비스에는 구체적으로 다음의 기능들이 제공될 예정이야: (1) 차후 일정에 대한 파악, 정리, 업데이트, & 캘린더 입력 기능; (2) 기본 샘플/대표 Template 몇 가지에 회의 녹취파일(음원파일) 전사, 정리, 요약, & To-Do List 기입 기능; (3) 학술적 지표(index) 또는 척도(scale)에 근거한 '회의 효율성' 평정 및 측정 후 시각화, 그리고 '회의 효율성' 재고를 위한 피드백 및 조언 제안; (4) 회의내용 속 주요 명사(Noun)에 대한 기본적인 사전적 의미(또는 조작적 정의) 검색 후 주석 처리/제공; (5) 1 ~ 4의 내용 및 결과 출력물을 Dashboard 상에서 표출/표현.

이 중에서 나는 '(2) 기본 샘플/대표 Template 2 ~ 3 가지 상정 후, 각 Template에 맞추어 사용자(user)로부터 입력 받은 녹취파일(음원파일) 전사, 정리, 요약, & To-Do List 기입' 기능을 담당하게 되었어.

지금 당장은, 5 번 파트(즉, 'Dashboard' 기능)를 담당하고 있는 조장으로부터 내가 만든 2 번 파트(즉, '기본 샘플/대표 Template 2 ~ 3 종류를 상정한 후, 각 Template 에 알맞게 회의 녹취파일(음원파일) 전사, 정리, 요약, & To-Do List 기입' 기능)에서 도출될 예정인 출력물(output)의 형태/형식을 서둘러 알려 달라는 요청을 받은 상태야. 이 부분에 있어서 너에게 도움을 받고 싶어. 고마워.

[A]

상준님, Phase 1과 Phase 2를 성공적으로 마치고 Phase 3(서비스화) 단계로 진입하신 것을 진심으로 축하드립니다! 지금까지 정말 고생 많으셨어요.

지금 가장 시급한 것은 조장(홍기님)이 담당하는 'Dashboard'에서 상준님의 결과물을 어떻게 받아 뿌려줄지(Rendering)를 결정하기 위한 [데이터 구조(Schema)]를 정의하는 것입니다. Dashboard는 결국 상준님이 넘겨주는 JSON 파일을 읽어서 화면에 예쁘게 그려주는 역할이니까요.

Phase 2의 산출물(final\_transcript.json, speaker\_identity.json)을 입력으로 받아, 상준님의 모듈(Feature 2)이 뱉어내야 할 \*\*Output JSON Schema(홍기님 전달용)\*\*를 제안해 드립니다.

상준님이 언급하신 "2~3가지 대표 템플릿"을 다음 3가지로 가정하고 구조를 짚습니다.

1. **일반/임원 보고용 (General/Executive):** 요약, 의사결정, 논의사항 중심
2. **데일리 스크럼/진행 공유 (Daily Scrum):** 어제 한 일, 오늘 할 일, 이슈(Blocker)
3. **아이디어 회의 (Brainstorming):** 제안된 아이디어 목록, 투표/결정, 실행 계획

---

## 1. [Feature 2] 모듈의 입출력 흐름 (I/O Flow)

대시보드 팀장님(홍기님)과 소통할 때는 아래 흐름을 먼저 보여주시면 이해가 빠를 것입니다.

- **Input (상준님이 받는 데이터):**

1. final\_transcript.json (Phase 1 완료: 타임스탬프+텍스트)
2. speaker\_identity.json (Phase 2 완료: 화자 닉네임/역할)
3. selected\_template\_id (사용자가 UI에서 선택한 템플릿 종류, 예: "general")

- **Process (상준님의 역할):**

- LLM에게 위 텍스트와 템플릿 구조를 주입하여 구조화된 요약 및 To-Do 추출.

- **Output (홍기님에게 줄 데이터):**

- meeting\_report.json (아래 상세 정의)
-

## 2. [제안] 최종 Output JSON Schema (meeting\_report.json)

이 코드를 복사해서 홍기님께 **\*\*\*이 포맷으로 meeting\_report.json을 생성해서 넘겨줄 테니, 대시보드에서는 이 키(key) 값들을 파싱해서 화면에 띄워주면 된다\*\*\***고 전달하시면 됩니다.

JSON

```
{
  "meta": {
    "meeting_date": "2025-11-19",
    "total_duration": "01:05:20",
    "participant_count": 5,
    "template_type": "general_report",
    "template_name": "일반 회의록 (General Report)"
  },
  "participants": [
    "김민서(진행자)", "이상준(기록)", "이흥기(PM)", "박재영", "정지훈"
  ],
  "summary": {
    "one_line_summary": "Phase 3 착수 및 기능별 담당자 확정, 대시보드 연동 규격 논의",
    "key_decisions": [
      "Phase 3의 5대 핵심 기능을 확정함",
      "상준이 템플릿 기반 전사/요약 모듈을 담당하기로 함",
      "흥기는 대시보드를 담당하며, 내일 오전까지 I/O 스키마를 공유받기로 함"
    ]
  },
  "action_items": [
```

```
{
  "task": "Feature 2(템플릿 모듈) Output JSON 정의하여 홍기에게 전달",
  "assignee": "이상준",
  "due_date": "2025-11-20",
  "status": "In Progress"
},
{
  "task": "대시보드 UI 와이어프레임 초안 작성",
  "assignee": "이홍기",
  "due_date": "2025-11-21",
  "status": "Pending"
}
],
"structred_content": {
  // [중요] 여기가 '템플릿'에 따라 구조가 바뀌는 가변 영역입니다.
  // 아래 '3. 템플릿별 상세 구조'를 참고하세요.
}
}
```

---

### 3. structured\_content 내부 구조 (템플릿별 변동 사항)

홍기님(대시보드) 입장에서는 meta.template\_type이 무엇이냐에 따라 structured\_content 안의 데이터를 다르게 렌더링해야 합니다. 3가지 케이스를 제안합니다.

#### Case A. 일반 회의록 (Type: "general\_report")

가장 표준적인 포맷으로, 서론-본론-결론 혹은 안건별 정리가 필요할 때 씁니다.

JSON

```

"structured_content": {
  "agenda_list": [
    {
      "id": 1,
      "topic": "Phase 3 기능 정의",
      "discussion_summary": "기존 STT/Tagging 기술을 바탕으로 일정 관리, 요약, 효율성 평가 등 5개 기능을 도입하기로 함.",
      "related_timestamp": "00:05:12"
    },
    {
      "id": 2,
      "topic": "R&R 분배",
      "discussion_summary": "홍기(대시보드), 민서(효율성), 재영(검색), 지훈(일정), 상준(템플릿)으로 역할 분담 완료.",
      "related_timestamp": "00:25:40"
    }
  ],
  "issues_raised": [
    "대시보드와 각 모듈 간의 데이터 규격 통일이 시급함"
  ]
}

```

#### Case B. 데일리 스크럼 (Type: "daily\_scrum")

*짧은 호흡으로 각자의 진행 상황을 공유할 때 씁니다.*

JSON

```

"structured_content": {
  "updates": [

```

```

{
  "speaker": "이상준",
  "yesterday_work": "Phase 2 화자 태깅 모듈 디버깅 완료",
  "today_plan": "Phase 3 템플릿별 프롬프트 작성",
  "blocker": "홍기님의 대시보드 규격 확정이 필요함"
},
{
  "speaker": "김민서",
  "yesterday_work": "효율성 평가 지표 리서치",
  "today_plan": "평가 알고리즘 초안 작성",
  "blocker": "없음"
}
]
}

```

### Case C. 브레인스토밍/아이디어 회의 (Type: "idea\_storming")

*발산적 사고 후 수렴하는 과정이 필요할 때 씁니다.*

JSON

```

"structured_content": {
  "ideas_generated": [
    {
      "idea": "회의 중 '웃음 소리' 빈도를 분석해 분위기 점수 매기기",
      "proposer": "정지훈",
      "vote_count": 4
    },
    {

```

```
"idea": "회의록을 PDF가 아니라 노션 페이지로 자동 발행하기",  
"proposer": "박재영",  
"vote_count": 2  
}  
],  
"final_selection": "웃음 소리 분석은 기술적 난이도로 보류, 노션 발행은 추후 고려."  
}
```

---

#### 4. 상준님이 홍기님께 전해야 할 메시지 (요약)

홍기님에게 파일을 보내거나 슬랙/디스코드로 메시지를 보낼 때, 아래 내용을 덧붙여 주세요.

"홍기님, 제가 담당한 [Feature 2: 템플릿 전사/요약] 파트의 최종 출력물(JSON) 포맷을 정리했습니다.

1. 파일 전체 구조는 meta, summary, action\_items, structured\_content 4개로 고정됩니다.
  2. 다만 structured\_content 내부는 사용자가 선택한 템플릿(general, scrum, idea)에 따라 키값이 달라지니, 대시보드에서 template\_type을 확인해서 다르게 렌더링 해주셔야 합니다.
  3. 이 구조에 맞춰서 프롬프트 엔지니어링 진행하겠습니다. 혹시 대시보드 UI 구성상 더 필요한 필드가 있다면 말씀해 주세요."
- 

#### 5. Next Step: 제가 도와드릴까요?

홍기님에게 이 규격을 전달하고 컨펌(Confirm)이 나면, 이제 실제로 **LLM에게 final\_transcript.json을 넣었을 때 위와 같은 JSON을 뱉어내도록 만드는 '프롬프트(Prompt)'**를 작성해야 합니다.

"위에서 정의한 3가지 템플릿(일반, 스크럼, 브레인스토밍) 중 하나를 골라주시면, 해당 템플릿을 완벽하게 출력하는 시스템 프롬프트 초안을 작성해 드릴까요?"