

Summary Meeting

[SuME] 서비스 제안서

서비스 이름	SuME	버전	2.0
팀	조현상(팀장), 정의현, 양지훈, 송예찬	작성일	2025. 2. 17

1. 서비스 기획

1. 서비스 개요

본 서비스는 회의 녹음을 기반으로 LLM(대규모 언어 모델)을 활용하여 자동으로 회의 내용을 요약하고, 핵심 논의 사항을 정리하여 캘린더에 일정을 저장하는 기능을 제공합니다. 사용자는 회의록을 직접 작성할 필요 없이 AI가 요약한 내용을 빠르게 확인할 수 있으며, 중요한 일정이 자동으로 정리되어 업무 효율성을 높일 수 있습니다.

2. 주요 기능

- 회의 녹음 및 AI 요약: 음성 데이터를 자동으로 텍스트로 변환하고, 핵심 내용을 요약하여 제공
- 일정 자동 생성: 논의된 일정 및 업무 마감일을 캘린더에 자동 등록

3. 기대 효과

- 업무 효율성 향상: 주요 논의 사항을 빠르게 확인하여 빠른 의사 결정 가능
- 편리한 일정 관리: 캘린더 연동을 통해 업무 계획을 체계적으로 정리

2. BM설계

1. 주요 고객(Target Customer)

- 기업 및 스타트업: 회의를 효율적으로 관리하고자 하는 조직
- 프리랜서 및 1인 기업: 고객 미팅 내용을 정리하고 일정화해야 하는 전문가
- 학생 및 연구자: 연구 및 스터디 모임의 기록과 일정 정리를 원하는 사용자

2. 핵심 가치(Value Proposition)

- AI를 활용한 자동 회의 요약으로 회의록 작성 시간 절감
- 캘린더 연동으로 업무 일정 관리 강화

3. 수익 모델(Revenue Model)

구독형 서비스(SaaS 모델)

- 무료 플랜: 일정 제한 시간과 캘린더 연동 서비스 기능 제공
- 프리미엄 플랜: 추가 녹음 시간과 캘린더 연동 서비스 기능 제공

3. API 정의서

1. API 개요

음성 업로드 및 일정 관리 시스템을 제공하는 Flask 기반의 API입니다. 사용자는 오디오 파일을 업로드하여 텍스트 변환(STT) 서비스를 이용하며, 이를 기반으로 일정을 자동으로 생성합니다. JWT 기반의 인증을 사용하여 사용자 계정을 보호하며, 이벤트를 관리합니다.

2. 기본 URL (Base URL)

<https://www.sume-ai.kro.kr>

3. API요청 시 인증방법

JWT Bearer Token 사용 ('Authorization: Bearer <토큰>')

3. 엔드포인트 목록

기능	HTTP 메서드	엔드포인트	설명
사용자 인증	POST	/auth/register	새로운 사용자를 등록
	POST	/auth/login	사용자 인증 후 JWT 토큰을 발급
오디오 업로드 및 변환	POST	/audio	사용자의 음성 파일 업로드
	GET	/audio	업로드된 오디오 파일 변환하여 텍스트로 반환
일정 관리	GET	/calendar/events	특정 사용자와 기간에 해당하는 이벤트 목록을 조회
	POST	/calendar/events	새로운 일정 추가
	DELETE	/calendar/events /<event_id>	특정 일정 삭제

4. 엔드포인트 상세

1. 사용자 인증 관련 엔드포인트

1.1 회원가입

- 엔드포인트: /auth/register
- 매서드: POST
- 설명: 새로운 사용자를 등록합니다.
- 요청예시:

```
{  
  "name": "John Doe",  
  "email": "john@example.com",  
  "password": "securepassword"  
}
```

- 응답예시:

```
{  
  "message": "Registration successful."  
}
```

1.2 로그인

- 엔드포인트: /auth/register
- 매서드: POST
- 설명: 새로운 사용자를 등록합니다.
- 요청예시:

```
{  
  "email": "john@example.com",  
  "password": "securepassword"  
}
```

- 응답예시:

```
{  
  "access_token": "your.jwt.token"  
}
```

2. 오디오 업로드 및 변환 엔드포인트

2.1 오디오 업로드

- 엔드포인트: /audio
- 매서드: POST
- 설명: 사용자의 음성파일을 업로드 합니다.
- 요청예시:

Form-data: {"audio": <파일>}

- 응답예시:

```
{
  "message": "File uploaded successfully",
  "filename": "audio_20240215_123456.wav",
  "file_path": "uploads/audio_20240215_123456.wav"
}
```

2.2 오디오 변환 및 STT 실행

- 엔드포인트: /audio
- 매서드: GET
- 설명: 업로드된 오디오 파일을 변환하여 텍스트로 반환합니다.
- 요청헤더:

Authorization: Bearer <JWT_TOKEN>

- 요청예시:

GET /audio?user_id=1&filename=audio_20240215_123456.wav

- 응답예시:

```
{
  "message": "텍스트 변환 결과"
}
```

3. 일정 관리 엔드포인트

3.1 일정조회

- 엔드포인트: /calendar/events
- 매서드: GET
- 설명: 특정 사용자와 기간에 해당하는 이벤트 목록을 조회합니다.
- 요청예시:

GET /calendar/events?user_id=1&start=2024-02-01&end=2024-02-28

- 응답예시:

```
[
  {
    "id": 1,
    "title": "회의",
    "description": "팀 회의 진행",
    "start_time": "2024-02-15T10:00:00",
    "end_time": "2024-02-15T11:00:00"
  }
]
```

3.2 일정 추가

- 엔드포인트: /calendar/events
- 매서드: POST
- 설명: 새로운 일정을 등록합니다.
- 요청예시:

```
{
```

```

    "title": "회의",
    "description": "팀 회의 진행",
    "start": "2024-02-15T10:00:00",
    "end": "2024-02-15T11:00:00",
    "user_id": 1
}

```

- 응답예시:

```

{
  "access_token": "Event created",
  "access_token": 1
}

```

3.3 일정삭제

- 엔드포인트: /calendar/events/<event_id>
- 매서드: DELETE
- 설명: 특정 일정을 삭제합니다.

- 요청예시:

```
DELETE /calendar/events/1
```

- 응답예시:

```

{
  "message": "Event delated"
}

```

4. 기획/개발/배포 과정

[기획 과정]

1단계: 요구사항 분석

- 기능적 요구사항:
 1. 음성 인식 및 텍스트 변환
 2. 캘린더 기능
 3. 요약 기능
 4. 사용자 관리(회원가입, 로그인)

2단계: 기술 스택 선정

- 프론트엔드:
 1. React (웹/앱 인터페이스 개발).
 2. TailwindCSS (디자인 간소화 및 사용자 친화적인 UI 개발)
 3. Fullcalendar (캘린더 기능 구현)
- 백엔드
 1. Flask (백엔드 서버 구축).
 2. JWT Token (인증 방식)
 3. REST API 방식 (API 친화적인 방법)
 4. SQLite (데이터베이스)
- AI
 1. Google Speech-to-Text (음성 -> 텍스트 변환)
 2. Google Gemini 1.5 Flash (텍스트 내용 요약)
 3. LangChain(AI 워크플로우 관리)
 4. LangGraph(AI 프로세스 최적화)

3단계: 아키텍처 설계

- 클라이언트-서버 구조:
프론트엔드: 사용자와 상호작용
백엔드: 데이터 처리 및 API제공
AI: 음성 처리 및 요약 기능 제공

- 데이터 플로우 설계:
녹음 업로드 → 텍스트 변환 → 요약 → 캘린더 생성

[개발 과정]

1단계: 음성 인식 및 텍스트 변환

- Google Speech-to-Text API 연동.
- 음성 데이터를 텍스트로 변환 및 저장

2단계: 요약

- Google Gemini 1.5 Flash API 연동
- 요약 기능 구현

3단계: 사용자 관리

- 회원가입 및 로그인 기능 개발
- 사용자 데이터 저장 및 관리

4단계: UI/UX 개발

- 녹음 및 요약 화면(실시간 녹음 시작, 중지 버튼과 요약 결과 표시 창)
- 캘린더 화면(생성된 일정 확인, 생성 및 삭제 가능)

[배포 과정]

1단계: Docker 환경 설정

- frontend와 backend의 실행 정보를 각각 Dockerfile로 정리.
- docker-compose.yml을 작성하여 frontend와 backend의 Dockerfile을 기반으로 컨테이너 실행을 관리.

2단계: 서버 클라우드 환경 구성

- 클라우드 서비스는 Azure를 사용하여 배포 환경 구축.

3단계: Nginx를 활용한 HTTPS 적용

- 오디오 기능을 위해 HTTPS가 필요하여 Nginx 설정 파일(.conf)을 작성.
- docker-compose.yml에 nginx를 추가하여 frontend, backend, nginx가 동일한 네트워크에서 작동하도록 구성.

4단계: 도메인 연결

- 무료 도메인 서비스인 '내도메인.한국'을 활용하여 도메인 등록 및 연결.

5단계: 배포 및 파일 전송

- 비공개 파일들을 scp 명령어를 사용해 로컬에서 서버로 전송.
- 모든 설정 완료 후 배포 진행, 성공적으로 서비스 운영 시작.

5. 운영 지침

1. 서비스 안정성 유지

- 서버 운영 및 장애 대응 체계 구축: 트래픽 급증 시 자동 확장(Auto Scaling) 및 장애 감지 시스템 운영
- 정기 유지보수 및 업데이트: 매월 정기 점검 및 AI 모델 성능 개선

2. 고객 지원 체제

- 환불 정책: 결제 후 7일 이내 3회 이하 사용 시 전액 환불 가능

3. 데이터 보호 및 보안 정책

- 데이터 암호화 적용
- 개인정보 보호법 준수
- 사용자 데이터 보존 정책: 일정 기간 미사용 계정의 자동 삭제 및 데이터 보호

4. 지속적인 서비스 개선

- 사용자 피드백 기반 업데이트: 주기적인 설문 조사 및 베타 테스트 운영
- AI 모델 최적화: 월간 업데이트 및 주요 성능 개선 반영