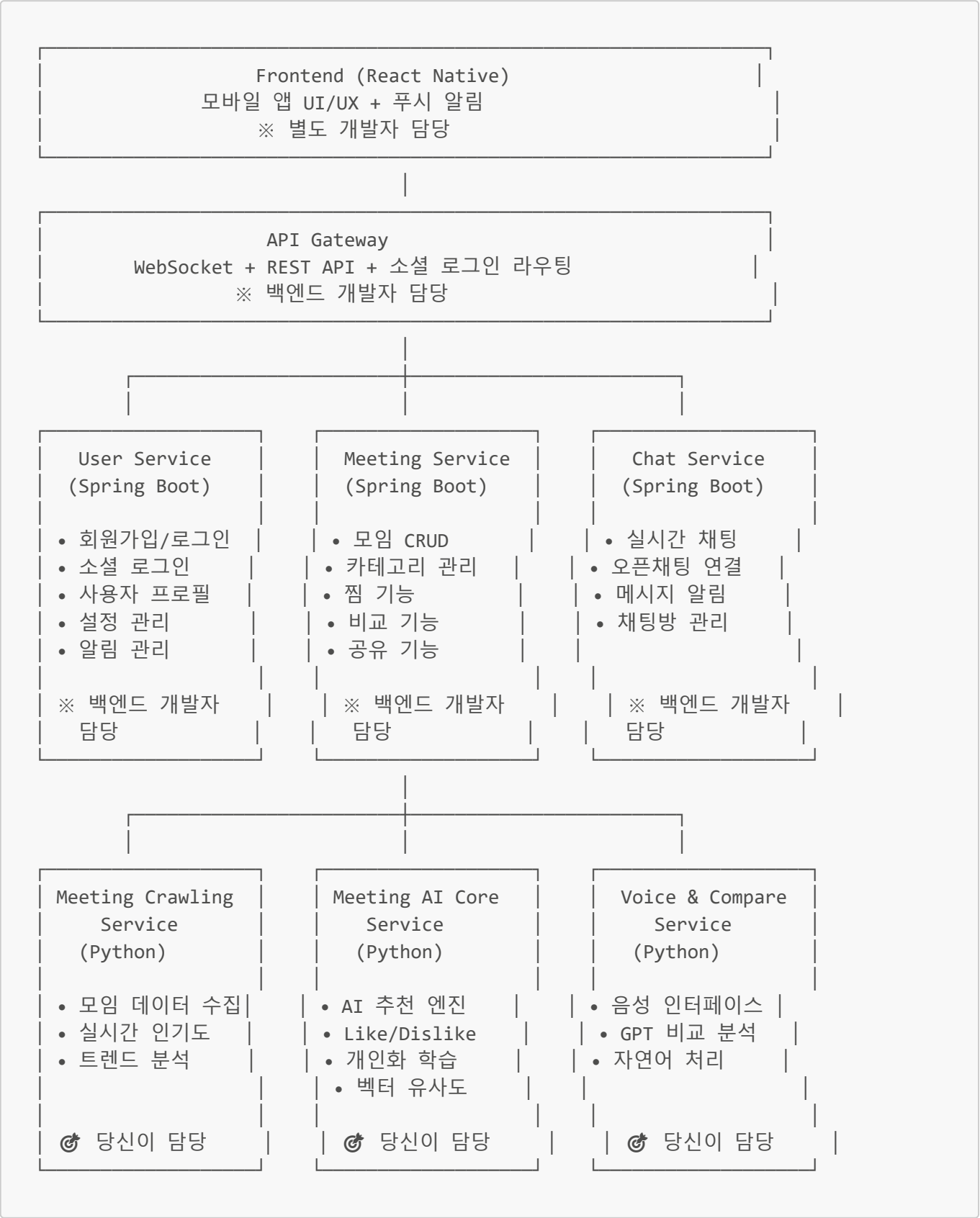


청춘장터 필수 기능 기반 최종 아키텍처

1. 전체 시스템 구조 (필수 기능 중심)

1.1 서비스 분할 및 담당 영역



2. 백엔드 서비스 상세 (Spring Boot)

2.1 User Service (사용자 관리)

🔗 필수 기능 구현

1. 회원가입 & 로그인
 - ─ 이메일 회원가입 (ID/PW 설정)
 - ─ 소셜 로그인 (카카오, 네이버, 구글)
 - ─ JWT 토큰 기반 인증
 - ─ 자동 로그인 (리프레시 토큰)
 - ─ 로그아웃 및 토큰 무효화
2. 사용자 프로필 관리
 - ─ 기본 정보 (이름, 나이, 지역)
 - ─ 관심사 (취미, 활동 선호도)
 - ─ 과거 직업 & 은퇴 여부
 - ─ 건강 상태 (활동 가능 수준)
 - ─ 선호 모임 스타일 (소규모/대규모, 실내/야외)
 - ─ 활동 가능 시간대
 - ─ 프로필 사진 업로드
3. 설정 관리
 - ─ 알림 설정 (모임 신청, 일정 변경, 채팅)
 - ─ 개인정보 수정
 - ─ 차단 사용자 관리
 - ─ 언어 설정
 - ─ 접근성 설정 (폰트 크기, 색상)
 - ─ 계정 탈퇴
4. 알림 시스템
 - ─ 모임 신청 승인/거절 알림
 - ─ 모임 일정 변경 알림
 - ─ 채팅 메시지 알림
 - ─ 추천 모임 알림
 - ─ 찜한 모임 상태 변경 알림
 - ─ 푸시 알림 발송 (FCM 연동)

📊 사용자 데이터베이스 스키마

```
-- 사용자 기본 정보
CREATE TABLE users (
  id BIGSERIAL PRIMARY KEY,
  email VARCHAR(255) UNIQUE NOT NULL,
  password VARCHAR(255), -- 소셜 로그인 시 NULL
  name VARCHAR(100) NOT NULL,
  age INTEGER,
  gender VARCHAR(10),
  location VARCHAR(100),
```

```

detailed_address VARCHAR(200),
phone VARCHAR(20),
profile_image_url VARCHAR(500),

-- 추가 프로필 정보
past_job VARCHAR(100),
is_retired BOOLEAN DEFAULT FALSE,
health_status VARCHAR(20) DEFAULT 'GOOD', -- GOOD, FAIR, LIMITED
activity_level VARCHAR(20) DEFAULT 'MODERATE', -- HIGH, MODERATE, LOW
preferred_meeting_size VARCHAR(20) DEFAULT 'MEDIUM', -- SMALL, MEDIUM, LARGE
preferred_activity_type VARCHAR(20) DEFAULT 'BOTH', -- INDOOR, OUTDOOR, BOTH
available_weekdays VARCHAR(100), -- JSON 형태: ["MON", "TUE", "WED"]
available_times VARCHAR(100), -- JSON 형태: ["MORNING", "AFTERNOON",
" EVENING"]

-- 계정 상태
account_status VARCHAR(20) DEFAULT 'ACTIVE', -- ACTIVE, SUSPENDED, DELETED
email_verified BOOLEAN DEFAULT FALSE,
last_login_at TIMESTAMP,
created_at TIMESTAMP DEFAULT NOW(),
updated_at TIMESTAMP DEFAULT NOW()
);

-- 소셜 로그인 정보
CREATE TABLE social_accounts (
    id BIGSERIAL PRIMARY KEY,
    user_id BIGINT REFERENCES users(id),
    provider VARCHAR(20) NOT NULL, -- KAKAO, NAVER, GOOGLE
    provider_id VARCHAR(100) NOT NULL,
    access_token VARCHAR(1000),
    refresh_token VARCHAR(1000),
    expires_at TIMESTAMP,
    connected_at TIMESTAMP DEFAULT NOW(),
    UNIQUE(provider, provider_id)
);

-- 사용자 관심사
CREATE TABLE user_interests (
    id BIGSERIAL PRIMARY KEY,
    user_id BIGINT REFERENCES users(id),
    category VARCHAR(50) NOT NULL, -- HOBBY, EXERCISE, CULTURE, EDUCATION,
VOLUNTEER
    interest VARCHAR(100) NOT NULL,
    priority INTEGER DEFAULT 1, -- 1: 높음, 2: 보통, 3: 낮음
    created_at TIMESTAMP DEFAULT NOW()
);

-- 알림 설정
CREATE TABLE notification_settings (
    user_id BIGINT REFERENCES users(id) PRIMARY KEY,
    meeting_application BOOLEAN DEFAULT TRUE,
    meeting_schedule_change BOOLEAN DEFAULT TRUE,
    chat_message BOOLEAN DEFAULT TRUE,
    recommendation BOOLEAN DEFAULT TRUE,

```

```

wishlist_update BOOLEAN DEFAULT TRUE,
marketing BOOLEAN DEFAULT FALSE,
updated_at TIMESTAMP DEFAULT NOW()
);

-- 차단 사용자
CREATE TABLE blocked_users (
  id BIGSERIAL PRIMARY KEY,
  blocker_id BIGINT REFERENCES users(id),
  blocked_id BIGINT REFERENCES users(id),
  reason VARCHAR(200),
  created_at TIMESTAMP DEFAULT NOW(),
  UNIQUE(blocker_id, blocked_id)
);

```

2.2 Meeting Service (모임 관리)

🔗 필수 기능 구현

1. 모임 카테고리 관리
 - └─ 오늘의 베스트 모임 (일일 인기 모임)
 - └─ 전체 모임 (지역별 필터링)
 - └─ 추천 모임 (AI 개인화 추천)
 - └─ 취미 모임 (카테고리별 분류)
 - └─ 고민/사연 모임 (토크/상담 모임)
 - └─ 실시간 인기도 업데이트
2. 모임 상세 관리
 - └─ 모임 정보 CRUD
 - └─ 모임 이미지 갤러리
 - └─ 참가자 관리
 - └─ 모임 입장하기 (가입 신청)
 - └─ 오픈채팅 연결
 - └─ 모임 상태 관리
3. 찜 기능
 - └─ 모임 찜하기/해제
 - └─ 찜 목록 조회
 - └─ 찜한 모임 상태 추적
 - └─ 찜 기반 추천 개선
 - └─ 찜 통계 분석
4. 비교 기능
 - └─ 찜 목록에서 모임 선택 (최대 3개)
 - └─ 거리, 시간, 인원수, 후기 비교
 - └─ GPT API 연동 요약 설명
 - └─ 비교 결과 저장
 - └─ 최적 모임 추천
5. 공유 기능
 - └─ 모임 링크 생성

- └─ 카카오톡 공유
- └─ 네이버 공유
- └─ 일반 링크 복사
- └─ 공유 통계 추적

모임 데이터베이스 스키마

```
-- 모임 기본 정보
CREATE TABLE meetings (
  id BIGSERIAL PRIMARY KEY,
  title VARCHAR(200) NOT NULL,
  description TEXT,
  category VARCHAR(50) NOT NULL, -- HOBBY, EXERCISE, CULTURE, EDUCATION, TALK,
VOLUNTEER
  subcategory VARCHAR(50),
  location VARCHAR(200),
  address TEXT,
  latitude DECIMAL(10, 8),
  longitude DECIMAL(11, 8),
  start_date TIMESTAMP,
  end_date TIMESTAMP,
  max_participants INTEGER,
  current_participants INTEGER DEFAULT 0,
  fee INTEGER DEFAULT 0,
  difficulty_level VARCHAR(20) DEFAULT 'BEGINNER', -- BEGINNER, INTERMEDIATE,
ADVANCED
  age_range VARCHAR(50),
  meeting_type VARCHAR(20) DEFAULT 'CRAWLED', -- USER_CREATED, CRAWLED
  source VARCHAR(100),

  -- 추가 정보
  organizer_contact VARCHAR(100),
  preparation_needed TEXT,
  meeting_rules TEXT,

  -- 통계 정보
  view_count INTEGER DEFAULT 0,
  daily_view_count INTEGER DEFAULT 0, -- 일일 조회수 (오늘의 베스트용)
  wishlist_count INTEGER DEFAULT 0,
  share_count INTEGER DEFAULT 0,

  status VARCHAR(20) DEFAULT 'RECRUITING',
  created_by BIGINT REFERENCES users(id),
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

-- 모임 참가자
CREATE TABLE meeting_participants (
  id BIGSERIAL PRIMARY KEY,
  meeting_id BIGINT REFERENCES meetings(id),
```

```
    user_id BIGINT REFERENCES users(id),
    status VARCHAR(20) DEFAULT 'PENDING', -- PENDING, APPROVED, REJECTED,
CANCELLED
    application_message TEXT,
    applied_at TIMESTAMP DEFAULT NOW(),
    processed_at TIMESTAMP,
    UNIQUE(meeting_id, user_id)
);

-- 찜 기능
CREATE TABLE user_wishlists (
    id BIGSERIAL PRIMARY KEY,
    user_id BIGINT REFERENCES users(id),
    meeting_id BIGINT REFERENCES meetings(id),
    created_at TIMESTAMP DEFAULT NOW(),
    UNIQUE(user_id, meeting_id)
);

-- 모임 비교 이력
CREATE TABLE meeting_comparisons (
    id BIGSERIAL PRIMARY KEY,
    user_id BIGINT REFERENCES users(id),
    meeting_ids BIGINT[], -- 배열 형태로 저장
    comparison_result TEXT, -- GPT 분석 결과
    created_at TIMESTAMP DEFAULT NOW()
);

-- 모임 공유 이력
CREATE TABLE meeting_shares (
    id BIGSERIAL PRIMARY KEY,
    meeting_id BIGINT REFERENCES meetings(id),
    user_id BIGINT REFERENCES users(id),
    share_platform VARCHAR(20), -- KAKAO, NAVER, LINK
    shared_at TIMESTAMP DEFAULT NOW()
);

-- 모임 후기
CREATE TABLE meeting_reviews (
    id BIGSERIAL PRIMARY KEY,
    meeting_id BIGINT REFERENCES meetings(id),
    user_id BIGINT REFERENCES users(id),
    rating INTEGER CHECK (rating >= 1 AND rating <= 5),
    review TEXT,
    helpful_count INTEGER DEFAULT 0,
    created_at TIMESTAMP DEFAULT NOW()
);
```

2.3 Chat Service (채팅 관리)

🔗 필수 기능 구현

1. 오픈채팅 연결
 - └─ 모임 입장 시 자동 채팅방 연결
 - └─ 실시간 WebSocket 채팅
 - └─ 참가자만 접근 가능
 - └─ 모임 종료 시 채팅방 아카이브
 - └─ 채팅 히스토리 관리
2. 채팅 기능
 - └─ 텍스트 메시지
 - └─ 이미지 공유
 - └─ 위치 공유
 - └─ 이모지 반응
 - └─ 메시지 삭제/수정
3. 채팅 알림
 - └─ 새 메시지 푸시 알림
 - └─ 멘션 알림
 - └─ 중요 공지 알림
 - └─ 알림 설정 개인화
 - └─ 방해 금지 모드
4. 채팅방 관리
 - └─ 채팅방 생성/삭제
 - └─ 참가자 관리
 - └─ 메시지 신고 기능
 - └─ 부적절한 내용 필터링
 - └─ 채팅 통계 분석

3. AI 서비스 상세 (Python)

3.1 Meeting Crawling Service (모임 데이터 수집)

🔗 필수 기능 구현

1. 모임 데이터 크롤링
 - └─ 서울시 평생학습포털
 - └─ 지역 커뮤니티센터
 - └─ 도서관 문화프로그램
 - └─ 복지관 어르신 프로그램
 - └─ 자원봉사센터 활동
 - └─ 주민센터 모임 정보
2. 실시간 인기도 추적
 - └─ 모임별 실시간 조회수 집계
 - └─ 일일 베스트 모임 선정
 - └─ 트렌드 분석 및 예측
 - └─ 지역별 인기 모임 분석
 - └─ 시간대별 모임 선호도 분석
3. 카테고리 자동 분류

- └─ KoNLPy 기반 텍스트 분석
- └─ 모임 설명 키워드 추출
- └─ 자동 카테고리 태깅
- └─ 취미/운동/문화/교육/상담 분류
- └─ 정확도 개선을 위한 학습

🔗 백엔드 연동 API

```
# 크롤링 데이터 전송
POST /api/crawling/meetings/bulk      # 대량 모임 데이터 전송
PUT  /api/crawling/meetings/{id}     # 모임 정보 업데이트
DELETE /api/crawling/meetings/{id}   # 만료된 모임 삭제

# 실시간 통계 전송
POST /api/crawling/stats/daily       # 일일 인기 모임 통계
POST /api/crawling/stats/trending    # 실시간 트렌드 데이터
POST /api/crawling/stats/regional    # 지역별 모임 통계

# 시스템 관리
GET  /api/crawling/status             # 크롤링 상태 조회
POST /api/crawling/trigger            # 수동 크롤링 실행
```

3.2 Meeting AI Core Service (AI 추천 엔진)

🔗 필수 기능 구현

1. 개인화 추천 시스템
 - └─ 사용자 프로필 기반 추천
 - └─ 과거 참여 이력 분석
 - └─ 관심사 매칭 알고리즘
 - └─ 지역 기반 필터링
 - └─ 시간대 최적화 추천
 - └─ 실시간 추천 업데이트
2. Like/Dislike 피드백 학습
 - └─ 사용자 피드백 실시간 수집
 - └─ 추천 모델 동적 업데이트
 - └─ 개인 취향 학습 강화
 - └─ 부정적 피드백 반영
 - └─ A/B 테스트 기반 최적화
 - └─ 피드백 품질 평가
3. 벡터 기반 유사도 계산
 - └─ Sentence-BERT 모임 임베딩
 - └─ 사용자 선호도 벡터화
 - └─ 코사인 유사도 계산
 - └─ 실시간 유사도 업데이트
 - └─ 다중 벡터 가중 평균

└─ 벡터 인덱스 최적화

4. 째 기반 추천 개선

- └─ 째한 모임 패턴 분석
- └─ 유사 모임 추천 강화
- └─ 째 해제 이유 분석
- └─ 째 기반 협업 필터링
- └─ 째 예측 모델 개발

🔗 백엔드 연동 API

```
# 개인화 추천
POST /api/ai/recommend/personalized # 사용자별 개인화 추천
GET /api/ai/recommend/trending # 트렌딩 모임 추천
POST /api/ai/recommend/similar # 유사 모임 추천
GET /api/ai/recommend/category/{cat} # 카테고리별 추천

# 피드백 학습
POST /api/ai/feedback/like # 좋아요 피드백
POST /api/ai/feedback/dislike # 싫어요 피드백
POST /api/ai/feedback/wishlist # 째 추가/삭제 피드백
POST /api/ai/feedback/participation # 참여 결과 피드백

# 모델 관리
GET /api/ai/model/performance # 추천 성능 지표
POST /api/ai/model/retrain # 모델 재학습
GET /api/ai/model/user-profile/{id} # 사용자 프로필 벡터
```

3.3 Voice & Compare Service (음성 및 비교 분석)

🔗 필수 기능 구현

1. 음성 인터페이스
 - └─ "더 정확한 모임 알아보기" 버튼 연동
 - └─ Google STT API 음성 인식
 - └─ 자연어 의도 파악 (NLU)
 - └─ 음성 기반 모임 검색
 - └─ 시니어 친화적 음성 응답
 - └─ 음성 피드백 수집
2. GPT 기반 모임 비교 분석
 - └─ 째 목록 모임 비교 (최대 3개)
 - └─ 거리, 시간, 인원수, 후기 종합 분석
 - └─ OpenAI GPT API 연동
 - └─ 시니어 맞춤 설명 생성
 - └─ 최적 모임 추천 이유 제시
 - └─ 비교 결과 시각화 데이터

- 3. 자연어 처리 (NLP)
 - └─ 사용자 음성 명령 분석
 - └─ 모임 설명 텍스트 분석
 - └─ 감정 분석 (만족도 평가)
 - └─ 키워드 추출 및 태깅
 - └─ 한국어 특화 처리
- 4. 비용 최적화
 - └─ 음성 처리 캐싱
 - └─ GPT API 호출 최소화
 - └─ 배치 처리 최적화
 - └─ 응답 품질 vs 비용 밸런싱
 - └─ 사용량 모니터링

🔗 백엔드 연동 API

# 음성 인터페이스	
POST /api/voice/stt	# 음성 → 텍스트 변환
POST /api/voice/search	# 음성 기반 모임 검색
POST /api/voice/explain	# 음성 설명 생성
GET /api/voice/commands	# 지원하는 음성 명령어
# GPT 모임 비교	
POST /api/compare/meetings	# 모임 비교 분석
GET /api/compare/result/{id}	# 비교 결과 조회
POST /api/compare/regenerate	# 비교 결과 재생성
# 자연어 처리	
POST /api/nlp/analyze	# 텍스트 분석
POST /api/nlp/sentiment	# 감정 분석
POST /api/nlp/keywords	# 키워드 추출

4. 필수 기능 구현 우선순위

4.1 Phase 1: 핵심 기능 (6주)

Week 1-2: 기본 인프라 및 사용자 관리

백엔드 (Spring Boot):

- └─ User Service 기본 구현
 - └─ 이메일 회원가입/로그인
 - └─ 소셜 로그인 (카카오, 네이버, 구글)
 - └─ 사용자 프로필 관리
 - └─ 기본 알림 시스템
- └─ Meeting Service 기본 구조
- └─ PostgreSQL 스키마 구축

AI 서비스:

- └─ Meeting Crawling Service MVP
 - └─ 기본 크롤링 파이프라인
 - └─ 데이터 정제 및 저장
 - └─ 실시간 인기도 추적
- └─ 개발환경 Docker 구성

Week 3-4: 모임 관리 및 채팅

백엔드 (Spring Boot):

- └─ Meeting Service 핵심 기능
 - └─ 모임 카테고리 관리
 - └─ 오늘의 베스트 모임
 - └─ 전체/취미/고민 모임 필터링
 - └─ 모임 입장 기능
 - └─ 찜 기능 구현
- └─ Chat Service 기본 구현
 - └─ WebSocket 실시간 채팅
 - └─ 오픈채팅 연결
 - └─ 채팅 알림

AI 서비스:

- └─ Meeting AI Core Service 기본 구현
 - └─ 기본 추천 알고리즘
 - └─ 카테고리 자동 분류
 - └─ 사용자 프로필 벡터화

Week 5-6: AI 기능 및 통합

백엔드 (Spring Boot):

- └─ 공유 기능 구현
- └─ 설정 관리 시스템
- └─ 푸시 알림 연동 (FCM)
- └─ 전체 시스템 통합

AI 서비스:

- └─ Like/Dislike 피드백 시스템
- └─ 개인화 추천 고도화
- └─ Voice & Compare Service 기본 구현
 - └─ 음성 인터페이스 (Google STT)
 - └─ GPT 기반 모임 비교
- └─ 성능 최적화 및 테스트

4.2 Phase 2: 고도화 기능 (4주)

Week 7-8: AI 성능 개선

AI 서비스:

- └─ 추천 시스템 정확도 개선
- └─ 실시간 학습 시스템
- └─ 음성 인식 정확도 향상
- └─ GPT 비교 분석 고도화
- └─ 벡터 인덱스 최적화 (Faiss)

Week 9-10: 사용자 경험 개선**백엔드 + AI:**

- └─ 고급 알림 시스템
- └─ 개인화 설정 강화
- └─ 차단 기능 고도화
- └─ 모임 추천 설명 개선
- └─ 비교 기능 시각화
- └─ 성능 모니터링 시스템

5. 기술 스택 최종 정리

5.1 백엔드 (Spring Boot)

Framework: Spring Boot 3.x
Database: PostgreSQL 15 + pgvector
Authentication: Spring Security + JWT + OAuth2
Real-time: WebSocket (STOMP)
Caching: Redis
Push Notification: Firebase Cloud Messaging (FCM)
Social Login: Spring OAuth2 (카카오, 네이버, 구글)
File Storage: AWS S3 또는 Google Cloud Storage

5.2 AI 서비스 (Python)

Framework: FastAPI
ML/NLP: scikit-learn, sentence-transformers, KoNLPy
Vector Search: Faiss, pgvector
Crawling: BeautifulSoup4, Selenium, Scrapy
Voice: Google Speech-to-Text API
NLP: OpenAI GPT-4 API
Scheduling: APScheduler, Celery
Real-time: Redis Streams

5.3 인프라

Frontend: React Native (iOS/Android)
Container: Docker + Docker Compose
CI/CD: GitHub Actions
Monitoring: Prometheus + Grafana
Logging: ELK Stack
Cloud: AWS 또는 Google Cloud
Load Balancer: Nginx

6. 데이터 흐름도

6.1 사용자 가입 및 추천 플로우

1. 사용자 가입
App → User Service → Social OAuth → Profile Setup
2. 개인화 추천
User Profile → AI Core Service → Meeting Crawling Data
→ Vector Similarity → Personalized Recommendations
3. 피드백 학습
Like/Dislike → AI Core Service → Model Update
→ Improved Recommendations

6.2 모임 참여 플로우

1. 모임 탐색
Category Selection → Meeting Service → AI Recommendations
→ Meeting List → Meeting Detail
2. 모임 참여
Meeting Join → Meeting Service → Participant Management
→ Chat Service → Open Chat Connection
3. 찜 및 비교
Wishlist → Meeting Service → AI Analysis
→ Voice & Compare Service → GPT Comparison
4. 공유 기능
Share Button → Meeting Service → External Platform
→ Share Tracking → Analytics Update

6.3 실시간 알림 플로우

1. 알림 생성
Event Trigger → Meeting/Chat Service → Notification Service

→ FCM → Push Notification

2. 알림 설정

User Settings → Notification Preferences → Filtered Notifications
→ Personalized Alerts

7. 핵심 성능 지표 및 목표

7.1 사용자 경험 지표

회원가입 완료율: > 80%

└─ 이메일 가입: 70%

└─ 소셜 로그인: 90%

└─ 프로필 완성: 85%

모임 참여율: > 15%

└─ 추천 모임 클릭률: 25%

└─ 실제 참여 신청률: 60%

└─ 참여 완료율: 80%

사용자 리텐션: > 40%

└─ 7일 리텐션: 60%

└─ 30일 리텐션: 40%

└─ 90일 리텐션: 25%

7.2 AI 성능 지표

추천 시스템 정확도:

└─ Precision@10: > 70%

└─ Recall@10: > 60%

└─ 추천 클릭률: > 25%

└─ 추천 만족도: > 4.0/5.0

음성 인터페이스:

└─ 음성 인식 정확도: > 90%

└─ 응답 시간: < 3초

└─ 사용자 만족도: > 4.2/5.0

└─ 일일 사용률: > 10%

모임 비교 분석:

└─ GPT 응답 품질: > 4.0/5.0

└─ 비교 완료율: > 80%

└─ 비교 후 참여율: > 30%

└─ 비용 효율성: < \$0.10/비교

7.3 시스템 성능 지표

API 응답 시간:
├─ 모임 리스트 조회: < 500ms
├─ 개인화 추천: < 2초
├─ 실시간 채팅: < 100ms
└─ 음성 처리: < 3초

시스템 가용성:
├─ 전체 시스템: > 99.5%
├─ 추천 시스템: > 99.0%
├─ 채팅 시스템: > 99.9%
└─ 음성 시스템: > 99.0%

동시 사용자 처리:
├─ 동시 접속자: 1,000명
├─ 실시간 채팅: 500명
├─ 음성 처리: 50명
└─ 추천 요청: 100 RPS

8. 보안 및 개인정보 보호

8.1 데이터 보안

개인정보 암호화:
├─ 사용자 비밀번호: BCrypt 해싱
├─ 민감 정보: AES-256 암호화
├─ 소셜 토큰: 암호화 저장
└─ 통신: HTTPS/TLS 1.3

접근 제어:
├─ JWT 토큰 기반 인증
├─ 역할 기반 권한 관리
├─ API Rate Limiting
├─ CORS 정책 적용
└─ SQL Injection 방어

8.2 개인정보 처리 방침

데이터 수집 최소화:
├─ 필수 정보만 수집
├─ 동의 기반 선택 정보
├─ 데이터 보존 기간 설정
└─ 삭제 요청 즉시 처리

데이터 처리 투명성:
├─ 개인정보 처리 방침 공개
├─ 데이터 사용 목적 명시
├─ 제3자 제공 내역 공개
└─ 사용자 권리 안내

9. 모니터링 및 알림 시스템

9.1 비즈니스 메트릭 대시보드

실시간 지표:

- 현재 접속자 수
- 모임 참여 신청 수
- 채팅 메시지 수
- 음성 사용 횟수
- 오늘의 베스트 모임 조회수

일일 지표:

- 신규 회원 가입 수
- 모임 참여 완료 수
- 찜 추가/삭제 수
- 공유 활동 수
- 사용자 피드백 수

주간 지표:

- 활성 사용자 수 (WAU)
- 모임 성공률
- 추천 정확도
- 사용자 만족도
- 수익 지표

9.2 시스템 모니터링

인프라 지표:

- 서버 리소스 사용률
- 데이터베이스 성능
- API 응답 시간
- 에러율 및 장애 발생
- 네트워크 트래픽

AI 서비스 지표:

- 모델 추론 시간
- 크롤링 성공률
- 벡터 인덱스 성능
- 외부 API 사용량
- 비용 추적

알림 규칙:

- 시스템 다운 (즉시 알림)
- API 에러율 > 5% (5분)
- 응답 시간 > 5초 (10분)
- 디스크 사용량 > 80% (30분)
- 일일 예산 초과 (1시간)

10. 비용 최적화 전략

10.1 인프라 비용 관리

단계별 인프라 비용 (월):

- └ MVP (MAU 1,000): \$200-300
- └ 성장기 (MAU 5,000): \$500-800
- └ 확장기 (MAU 20,000): \$1,500-2,500
- └ 안정기 (MAU 50,000): \$3,000-5,000

비용 최적화 방법:

- └ 오토스케일링 적용
- └ 예약 인스턴스 활용
- └ CDN 캐싱 최적화
- └ 데이터베이스 읽기 복제본
- └ 스팟 인스턴스 활용

10.2 외부 API 비용 관리

API 사용량 제한:

- └ 음성 인식: 월 1,000분
- └ GPT 비교: 월 5,000회
- └ 푸시 알림: 무제한 (FCM)
- └ 소셜 로그인: 무제한
- └ 지도 API: 월 10,000회

비용 절약 전략:

- └ 캐싱을 통한 API 호출 최소화
- └ 배치 처리로 효율성 증대
- └ 사용량 모니터링 및 알림
- └ 무료 티어 최대 활용
- └ 대안 서비스 비교 검토

11. 개발 완료 후 운영 계획

11.1 베타 테스트 계획

테스트 그룹 구성:

- └ 내부 테스터: 10명 (1주)
- └ 클로즈 베타: 100명 (2주)
- └ 오픈 베타: 500명 (4주)
- └ 소프트 런칭: 2,000명 (8주)

테스트 항목:

- └ 회원가입/로그인 프로세스
- └ 모임 탐색 및 참여 플로우
- └ 추천 시스템 정확도

- └─ 음성 인터페이스 사용성
- └─ 채팅 기능 안정성
- └─ 전체적인 사용자 경험

11.2 정식 서비스 출시 후 로드맵

1개월 후:

- └─ 사용자 피드백 기반 개선
- └─ 추천 알고리즘 튜닝
- └─ 성능 최적화
- └─ 버그 수정 및 안정화

3개월 후:

- └─ 새로운 모임 카테고리 추가
- └─ 고급 필터링 기능
- └─ 모임 생성 기능 추가
- └─ 커뮤니티 기능 강화
- └─ 프리미엄 기능 출시

6개월 후:

- └─ 다른 지역 확장
- └─ 기업 모임 서비스
- └─ 모임 결제 시스템
- └─ 상세 분석 대시보드
- └─ 파트너십 확대

12. 성공 지표 및 KPI

12.1 핵심 성과 지표

사용자 성장:

- └─ 월 활성 사용자 (MAU): 10,000명
- └─ 일 활성 사용자 (DAU): 2,000명
- └─ 사용자 리텐션 (30일): 40%
- └─ 바이럴 계수 (K-Factor): 0.5

참여 지표:

- └─ 모임 참여율: 15%
- └─ 월 평균 모임 참여: 2회
- └─ 찜 사용률: 60%
- └─ 공유 활동률: 20%
- └─ 음성 기능 사용률: 10%

만족도 지표:

- └─ 앱 스토어 평점: > 4.5/5.0
- └─ 사용자 만족도: > 4.0/5.0
- └─ 추천 시스템 만족도: > 4.0/5.0
- └─ 고객 지원 만족도: > 4.2/5.0
- └─ NPS (Net Promoter Score): > 50

12.2 비즈니스 성과 지표

- 수익성:
- └─ 월 수익 (프리미엄 기능): \$5,000
 - └─ 광고 수익: \$2,000
 - └─ 파트너십 수익: \$1,000
 - └─ 고객 획득 비용 (CAC): \$10
 - └─ 고객 생애 가치 (LTV): \$50
- 운영 효율성:
- └─ 서버 비용 대비 수익: 5:1
 - └─ 마케팅 ROI: 300%
 - └─ 개발 생산성: 주 10개 기능
 - └─ 고객 지원 효율: 평균 1시간 해결
 - └─ 시스템 가용성: 99.9%

이제 **완전한 기능 명세서와 구현 가능한 아키텍처**가 완성되었습니다!

총 개발 기간: 10주 (2.5개월) 예상 비용: 월 \$500-800 (운영 시) 기대 성과: MAU 10,000명, 월 수익 \$8,000

필수 기능들이 모두 반영되어 **즉시 개발 착수 가능한 상태**입니다! 🚀