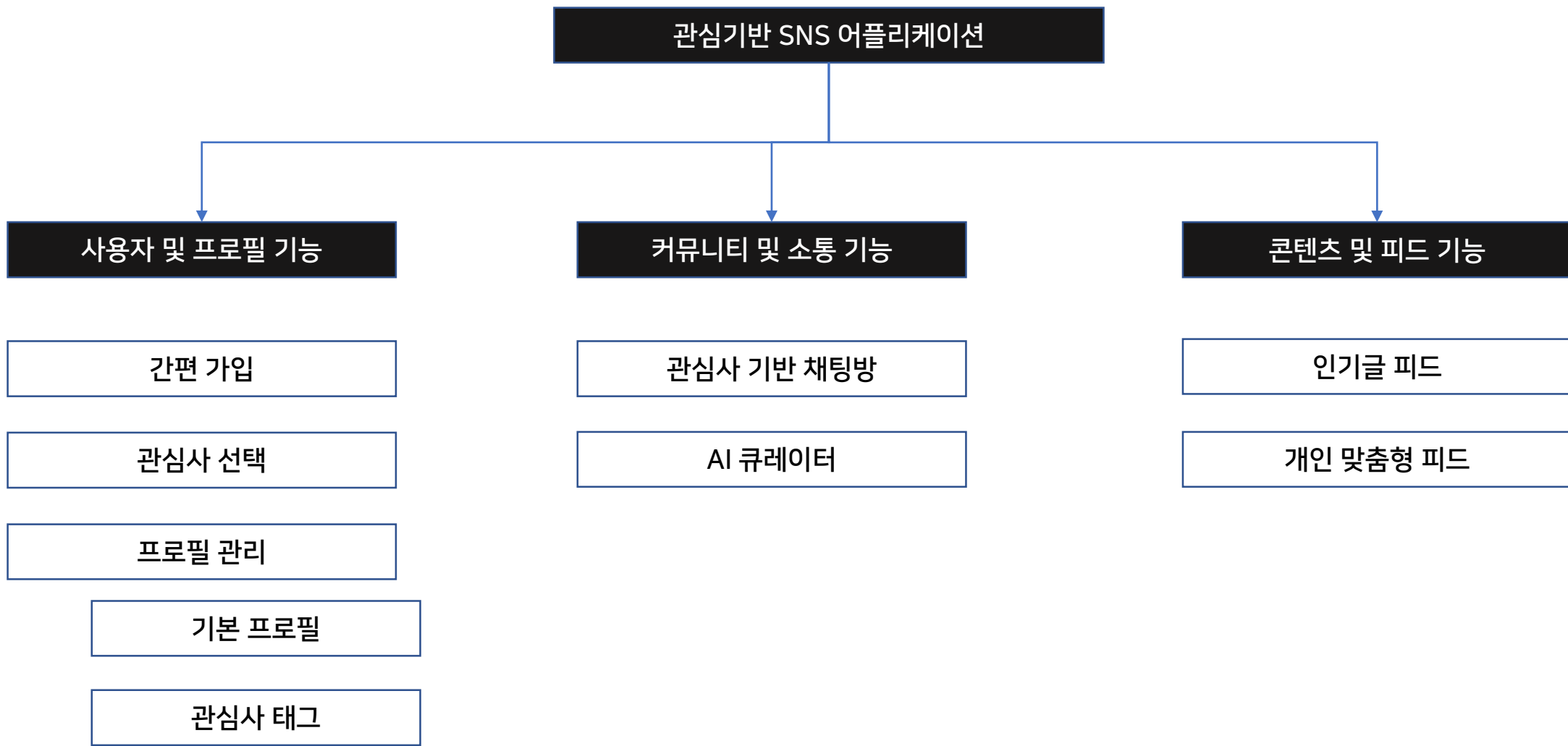
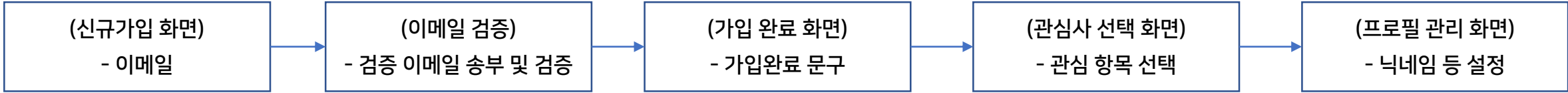


기능도

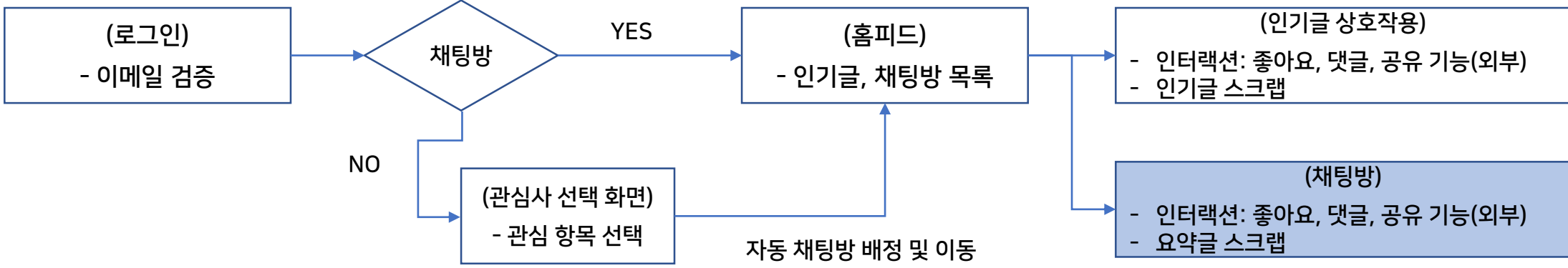


Task Flow

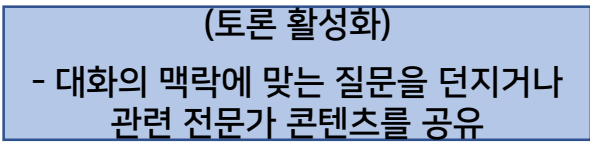
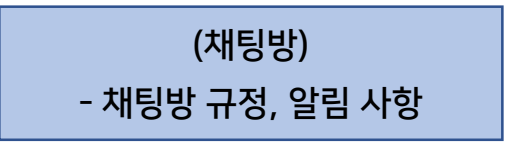
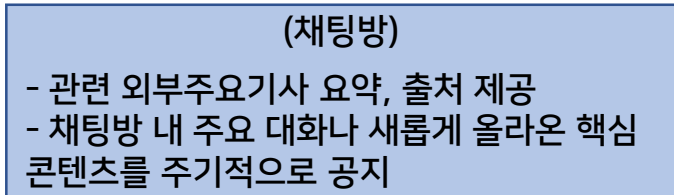
사용자 간편가입



관심사 기반 채팅방



AI 큐레이터



바이브코딩 - Prompt

사용자 간편가입(온보딩)

화면 1: 시작 화면

- ◆ **목적:** 사용자가 서비스 가입 또는 로그인을 시작하는 진입점입니다.
- ◆ **UI 컴포넌트:**
 - 타이틀: "로그인" 또는 서비스 로고
 - 버튼: "시작하기"
- ◆ **사용자 액션:** 사용자가 '시작하기' 버튼을 클릭합니다.
- ◆ **시스템 로직:**
 - '시작하기' 버튼 클릭 시, **화면 2: 이메일 입력 및 검증 화면**으로 이동합니다.

화면 2: 이메일 입력 및 검증

- ◆ **목적:** 사용자의 이메일 주소를 입력받고, 가입 가능한 이메일인지 서버를 통해 검증합니다.
- ◆ **UI 컴포넌트:**
 - 텍스트: "Email :"
 - 입력 필드(Input Field): 이메일 주소 입력
 - 버튼: "전송"
- ◆ **사용자 액션:** 이메일을 입력하고 '전송' 버튼을 클릭합니다.
- ◆ **시스템 로직:**
 - **(Front-end):** '전송' 버튼 클릭 시, 입력된 값이 유효한 이메일 형식인지 클라이언트 단에서 1차 검증합니다.
 - **(Back-end API 호출):** 유효한 형식일 경우, 서버로 API를 호출하여 해당 이메일이 이미 가입되었는지, 혹은 유효한지 검증을 요청합니다.
 - **(Front-end):** 서버로부터 '사용 가능' 응답을 받으면 **화면 3: 관심사 선택 화면**으로 이동합니다. 이메일 정보는 다음 단계로 전달하기 위해 상태(state)에 저장합니다.

화면 3: 관심사 선택

- ◆ **목적:** 사용자의 관심사를 파악하여 개인화된 콘텐츠 및 커뮤니티 추천의 기반을 마련합니다.
- ◆ **UI 컴포넌트:**
 - 관심사 버블(Bubble) / 태그 (Tag): "오픈 AI", "기술", "테니스" 등 미리 정의된 관심사 목록 (클릭 가능)
 - 입력 필드(선택 사항): "관심사 입력" (사용자가 직접 입력할 수 있는 기능)
 - 버튼: "다음"
- ◆ **사용자 액션:** 제시된 관심사 버블 중 하나 이상을 선택하거나 직접 입력한 후 '다음' 버튼을 클릭합니다.
- ◆ **시스템 로직:**
 - 사용자가 선택한 관심사 목록을 상태(state)에 저장합니다.
 - 최소 1개 이상의 관심사를 선택해야 '다음' 버튼이 활성화되도록 로직을 구현합니다.
 - '다음' 버튼 클릭 시, **화면 4: 닉네임 설정 화면**으로 이동합니다.

화면 4: 닉네임 설정

- ◆ **목적:** 서비스 내에서 사용할 사용자의 닉네임을 합니다.
- ◆ **UI 컴포넌트:**
 - 텍스트: "닉네임 :"
 - 입력 필드: 닉네임 입력
 - 버튼: "다음" (또는 "완료")
 - **사용자 액션:** 닉네임을 입력하고 '다음' 버튼을 클릭합니다.
- ◆ **시스템 로직:**
 - **(Front-end):** 닉네임의 유효성(글자 수, 특수문자 등)을 검증합니다.
 - **(Back-end API 호출):** '다음' 버튼 클릭 시, 상태(state)에 저장해 온 이메일, 관심사 목록, 닉네임 정보를 종합하여 서버의 가입 API로 전송합니다.
 - **(Back-end):** 서버는 닉네임 중복을 한 번 더 확인하고, 유효할 경우 사용자 계정을 생성합니다. 성공적으로 생성되면 인증토큰(JWT 등)을 발급하여 클라이언트에 전달합니다.
 - **(Front-end):** 서버로부터 인증토큰을 수신하여 안전하게 저장하고, 사용자를 **화면 5: 채팅 화면**으로 리디렉션(redirection)합니다.

화면 5: 채팅

- ◆ **목적:** 온보딩을 완료한 사용자가 처음 마주하는 서비스의 핵심 화면입니다.
- ◆ **UI 컴포넌트:**
 - 상단 탭 네비게이션: "홈", "채팅", "프로필"
 - 콘텐츠 영역: "주요 대화 내용 요약", "채팅방" 등
 - 입력 영역: "대화글 입력"
- ◆ **시스템 로직:**
 - 이 화면으로 진입했다는 것은 사용자의 **온보딩 및 로그인이 성공적으로 완료**되었음을 의미합니다.
 - 온보딩 시 선택했던 관심사를 기반으로 추천 채팅방에 자동으로 참여시키며, 관련 콘텐츠를 '주요 대화 내용 요약' 영역에 보여주는 로직을 구현합니다.

바이브코딩 - Prompt

관심사 기반 채팅방

- ◆ 본 문서는 '관심사 기반 채팅방' 기능 구현을 위한 개발 가이드입니다. 이 플로우는 **채팅 탭** 내에서 크게 4가지 화면(View)으로 구성됩니다: ① 피드 뷰, ② 채팅방 목록 뷰, ③ 개별 채팅방 뷰, ④ 구독글 뷰. 사용자는 이 흐름을 통해 여러 커뮤니티의 핵심 정보를 탐색하고, 직접 대화에 참여하며, 원하는 정보를 검색할 수 있습니다.
- ◆ 공통 컴포넌트
 - 상단 메인 네비게이션: [홈], [채팅], [프로필] 탭은 최상단에 항상 고정됩니다.
 - 상단 서브 네비게이션: [홈] 탭 내부에 [인기글], [구독글] 등의 서브 탭이 존재하며, 이 탭들을 통해 화면 6과 화면 9 사이를 전환합니다

화면 6: 피드 뷰 (인기글)

- ◆ 목적: 여러 채팅방에서 발생한 주요 대화나 인기글을 모아 보여주는 뉴스피드 형태의 화면입니다. ([홈] - [인기글])
- ◆ UI 컴포넌트:
 - 콘텐츠 리스트: Bluesky의 피드 형태
 - 프로필 영역: 글쓴이의 프로필 사진, 닉네임 등을 표시합니다. (회색 원으로 표현됨)
 - 콘텐츠 영역: 메시지 본문 또는 이미지 썸네일을 표시합니다.
 - 액션 버튼 그룹: [댓글], [좋아요], [저장], [공유] 버튼
- ◆ 사용자 액션:
 - 아래로 스크롤하여 더 많은 피드를 로드합니다
 - [인기글] / [구독글] 서브 탭을 클릭하여 피드 종류를 변경할 수 있습니다
- ◆ 개발 포인트 (Logic):
 - 상태 관리: 각 피드 아이템의 '좋아요' 여부, '저장' 여부 등은 클라이언트 상태로 관리하여 사용자의 인터랙션에 즉각적으로 반응하도록 합니다.

화면 7: 채팅방 목록 뷰

- ◆ 목적: 사용자가 참여 중인 모든 채팅방 목록을 보여주며, 새로운 채팅방을 탐색하거나 개설하는 역할을 합니다.
- ◆ UI 컴포넌트:
 - 기능 버튼: [채팅방 검색], [채팅방 만들기].
 - 채팅방 리스트: 각 아이템은 하나의 채팅방을 나타냅니다. (일반적으로 '채팅방 이름', '마지막 메시지', '안 읽은 수'가 포함됩니다)
- ◆ 사용자 액션:
 - 리스트에서 특정 채팅방 아이템을 클릭하면 **화면 8 (개별 채팅방 뷰)**으로 이동합니다.
- ◆ 개발 포인트 (Logic):
 - API 연동: API를 호출하여 사용자가 속한 채팅방 목록 데이터를 가져옵니다.
 - 실시간 업데이트: WebSocket을 통해 새로운 메시지가 도착하면 해당 채팅방의 '마지막 메시지'와 '안 읽은 수'가 실시간으로 변경되어야 합니다.

화면 8: 개별 채팅방 뷰

- ◆ 목적: 사용자들이 실시간으로 메시지를 주고받는 핵심 대화 공간입니다.
- ◆ UI 컴포넌트:
 - 채팅방 헤더: 채팅방 이름, 참여자 정보 등.
 - 메시지 스크롤 영역: 대화 내용이 표시되는 메인 영역.
 - AI 큐레이터 공지 : 주기적으로 채팅방 주제와 연관된 주요 기사 및 출처를 공유합니다. 채팅방의 주요 내용을 요약 합니다.
 - 메시지 입력 필드: 사용자가 메시지를 입력하고 전송하는 하단 영역.
- ◆ 사용자 액션:
 - 메시지를 입력하고 전송합니다.
 - 과거 대화를 보기 위해 위로 스크롤합니다.
- ◆ 개발 포인트 (Logic):
 - WebSocket 연동: 실시간 통신입니다.
 - 화면 진입 시: 특정 roomId로 소켓에 join.
 - API 연동: 화면에 처음 진입할 이전 대화 기록을 불러옵니다.

화면 9: 구독글뷰

- ◆ 목적: 사용자가 참여되어 있는 채팅방의 AI 큐레이터 공지 및 요약글을 모아 보여줍니다. . ([홈] - [구독글])
- ◆ UI 컴포넌트:
 - 검색 입력 필드: 사용자가 검색어를 입력하는 공간.
 - 검색 결과 리스트: 검색 결과가 화면 6과 유사한 형태로 표시됩니다.
 - 액션 버튼 그룹: [댓글], [좋아요], [공유] 버튼
- ◆ 사용자 액션:
 - 검색 필드에 키워드를 입력하고 검색을 실행합니다
- ◆ 개발 포인트 (Logic):
 - API 연동: 사용자가 검색어를 입력하고 검색을 실행하면 API를 호출하여 결과를 받아옵니다.

바이브코딩 - Prompt

AI 큐레이터

- ◆ 본 문서는 'AI 큐레이터' 역할에 대한 설명으로 AI 큐레이터는 커뮤니티의 사회자(Moderator), 기록자(Archivist), 발행인(Publisher) 역할을 동시에 수행합니다. 이 기능은 백그라운드에서 비동기적으로 작동하며, 그 결과물을 채팅방과 홈 피드에 제공합니다.
- ◆ AI 큐레이터 핵심 기능
 - 콘텐츠 제공 및 토론 활성화 (사회자 역할) : 채팅방의 핵심 주제와 관련된 외부 기사, 전문가 블로그, 관련 콘텐츠 링크를 주기적으로 공유합니다. 또한, 대화가 소강상태 일 때 토론을 활성화할 수 있는 질문을 던집니다.
 - 대화 내용 요약 및 기록 (기록자 역할) : 일정 시간(예: 매일 자정) 또는 일정 대화량(예: 메시지 100개)을 기준으로, 해당 기간 동안의 핵심 논의 내용을 자동으로 요약합니다.
 - 요약본 발행 (발행인 역할) : 생성된 대화 요약본을 [홈] - [구독글] 피드에 하나의 독립된 콘텐츠 카드 형태로 발행(Publish)합니다.
- ◆ AI 큐레이터는 사용자의 직접적인 요청 없이 서버 단에서 독립적으로 작동해야 합니다.
- ◆ (Trigger) 이벤트 트리거:
 - 스케줄링 기반 (Cron Job): 매일 아침, 6시간마다 등 정해진 시간에 작동합니다. (예: 대화 요약, 주간 인기 토픽 공유)
 - 최소 1시간 이상 대화없을때
- ◆ (Process) 데이터 처리 (Back-end):
 - 트리거가 발생하면, 서버는 해당 채팅방의 최근 메시지 로그를 데이터베이스에서 조회합니다.
 - 조회한 텍스트 데이터를 **LLM API (예: OpenAI API)**로 전송합니다. 이때, API에 전달하는 프롬프트가 각 기능의 핵심입니다.
 - 기사 추천 요청 프롬프트 예시:
 - "다음 대화 내용을 기반으로, 이 토론을 심화시킬 수 있는 최신 온라인 아티클 3개를 찾아줘. 각 아티클의 제목과 URL을 JSON 형식으로 반환해줘."
 - 대화 요약 요청 프롬프트 예시:
 - "다음 채팅 대화록의 핵심 논의 내용을 세 가지 불렛 포인트로 요약해줘. 이 요약은 '오늘의 대화 요약'이라는 제목으로 게시될 거야."
- ◆ (Action) 결과 실행:
 - 채팅방 메시지 전송: LLM으로부터 받은 결과(기사 링크, 질문, 요약 등)를 type: 'ai-curator'와 같은 특별한 시스템 메시지 형태로 해당 채팅방에 전송합니다. (WebSocket을 통해 클라이언트에 전달)
 - 데이터베이스 저장: 생성된 대화 요약본은 별도의 Summaries 또는 Feeds 컬렉션(테이블)에 저장합니다. 이 데이터에는 생성된 채팅방 ID, 요약 내용, 생성 일시 등이 포함되어야 합니다.
- ◆ (Display) 화면 표시:
 - 채팅방: type: 'ai-curator' 메시지를 수신한 클라이언트는 일반 사용자 메시지와 다른 UI 컴포넌트(예: 강조된 배경색, 아이콘)로 렌더링합니다.
 - 홈 피드 ([구독글]): API는 사용자가 구독한 채팅방에서 생성된 요약본(Summaries 컬렉션)을 조회하여 피드에 맞는 카드 형태로 내려줍니다

바이브코딩 - Prompt

프로필

홈

채팅

프로필

프로필 관리 나의 활동 점수

◆ UI 컴포넌트

- **닉네임 입력 필드 (Input Field):** 현재 닉네임이 표시되고, 수정이 가능한 텍스트 필드입니다.
- **관심사 태그 목록 (Tag List):** 현재 설정된 관심사 태그들이 표시되는 영역입니다. 각 태그 옆에는 삭제(X) 아이콘이 포함될 수 있습니다.
- **관심사 태그 추가 입력 필드 (Input Field):** 새로운 관심사를 입력하고 추가할 수 있는 텍스트 필드입니다.
- **저장/완료 버튼 (Button):** 변경된 프로필 정보를 서버에 저장하는 버튼입니다.

◆ 사용자 액션

- **닉네임 변경:** 닉네임 입력 필드를 터치하고 새로운 닉네임을 입력한 후 저장합니다.
- **관심사 태그 추가:** 사용자가 관심사 태그 추가 필드에 새로운 태그를 입력하고 '추가' 또는 '완료' 버튼을 누릅니다.
- **관심사 태그 삭제:** 사용자가 기존에 등록된 관심사 태그 옆의 삭제(X) 아이콘을 터치하여 태그를 제거합니다.

◆ 개발 포인트 (Logic)

- **API 연동:** 사용자가 '저장' 버튼을 누르면, 변경된 닉네임과 관심사 태그 정보를 서버로 전송하는 API를 호출해야 합니다.
- **데이터 유효성 검사 (Validation):** 닉네임 변경 시, 중복 닉네임 확인 및 비속어, 특수문자 사용 제한 등의 유효성 검사 로직이 필요합니다.

홈

채팅

프로필

프로필 관리 나의 활동 점수

◆ UI 컴포넌트

- **지표 표시 레이블 및 값 (Label & Value):** '총 활동 시간', '총 대화 횟수' 등 각 지표의 이름과 해당 수치를 보여주는 텍스트 컴포넌트입니다.
- **총 활동 시간:** 서비스 내에서 체류한 누적 시간
- **총 대화 횟수:** 작성한 총 메시지 수
- **핵심 주제어 대화 횟수:** 참여 중인 채팅방의 주요 주제어가 포함된 메시지 수
- **받은 답장/리액션:** 다른 사용자로부터 받은 답장 및 '좋아요' 등의 리액션 총 횟수
- **방장 :** 대화방 개설 수
- **방 참여자수 :** 대화방에 참여한 사용자 수

◆ 사용자 액션

- **페이지 조회:** 사용자가 프로필 페이지 내의 '나의 활동 점수' 섹션 메뉴를 통해 활동 점수 모니터링 페이지로 진입합니다.
- **지표 확인:** 각 활동 지표의 구체적인 수치를 확인하여 자신의 커뮤니티 기여도를 파악합니다.

◆ 개발 포인트 (Logic)

- **데이터 로깅 (Data Logging):** 사용자의 모든 활동(앱 체류 시간, 메시지 전송, '좋아요' 수신, 채팅방 개설 등)을 서버에 기록하는 로직이 필요합니다.
- **일일 데이터 집계 (Daily Batch Processing):** 서버에 쌓인 로그 데이터를 매일 자정 등 정해진 시간에 분석하고 합산하여 각 사용자별 활동 지표를 계산하는 배치(Batch) 작업이 필요합니다.
- **핵심 주제어 대화 카운팅:** 채팅방의 주요 주제어와 메시지 내용을 비교하여 '핵심 주제어 대화 횟수'를 계산하는 로직이 필요합니다. (자연어 처리(NLP) 기술이 사용될 수 있습니다.)

바이브코딩 - Prompt

관리자 페이지

1. 대시보드 (Dashboard)

서비스의 전반적인 현황을 한눈에 파악하는 초기 화면입니다.

UI 컴포넌트

핵심 지표 카드 (Statistic Cards): 총 가입자 수, 일일 활성 사용자(DAU), 신규 가입자, 총 채팅방 수 등 핵심 데이터를 표시합니다.

현황 그래프 (Charts): 사용자 증가 추이, 시간대별 활동량 등 주요 데이터의 변화를 보여주는 시각 자료입니다.

최신 활동 로그 (Activity Feed): 주요 관리자 활동이나 사용자 신고 등 최근 이벤트를 시간순으로 보여줍니다.

관리자 액션

서비스의 주요 지표를 실시간으로 확인하여 서비스 상태를 점검합니다.
데이터 추이를 보고 이상 징후나 패턴을 빠르게 파악합니다.

개발 포인트 (Logic)

데이터 집계 API: 여러 데이터베이스의 정보를 요약하여 대시보드에 필요한 통계 데이터를 제공하는 API를 구현합니다.

주기적 업데이트: 스케줄러(Scheduler)를 이용해 주기적으로 데이터를 최신 상태로 업데이트하는 로직이 필요합니다.

2. 사용자 관리 (User Management)

모든 사용자의 정보를 조회하고 관리하는 기능입니다.

UI 컴포넌트

사용자 목록 테이블 (User Table): 사용자 ID, 닉네임, 가입일, 활동 점수, 계정 상태(활성/정지) 등의 정보를 목록으로 보여줍니다.

검색 및 필터 (Search & Filter): 닉네임, 이메일 등으로 특정 사용자를 검색하거나 가입일, 활동 점수 순으로 정렬하는 기능을 제공합니다.

사용자 상세 정보 모달/페이지 (User Detail Modal): 특정 사용자의 프로필 정보, 활동 내역, 제재 기록 등 모든 상세 정보를 보여줍니다.

관리 액션 버튼 (Action Buttons): 사용자 계정을 '정지', '삭제', '정보 수정' 할 수 있는 버튼입니다.

관리자 액션

특정 사용자의 활동 내역과 정보를 조회합니다.

커뮤니티 정책을 위반한 사용자의 계정을 정지시키거나 탈퇴 처리합니다.

사용자의 요청에 따라 닉네임 등 프로필 정보를 수정합니다.

개발 포인트 (Logic)

사용자 CRUD API: 사용자 정보의 생성(Create), 조회(Read), 수정(Update), 삭제>Delete)를 처리하는 API가 필요합니다.

관리자 활동 로그: 어떤 관리자가 언제, 누구의 정보를 수정하거나 제재했는지 기록을 남겨 책임 소재를 명확히 해야 합니다.

계정 상태 관리: '정지' 상태의 사용자는 로그인이나 활동이 불가능하도록 처리하는 로직을 구현합니다.

바이브코딩 - Prompt

관리자 페이지

3. 콘텐츠 관리 (Content Management)

사용자가 생성하는 채팅방, 메시지 등을 모니터링하고 관리하는 기능입니다.

UI 컴포넌트

신고 내역 목록 (Report Queue): 사용자들이 신고한 메시지나 사용자를 리스트 형태로 보여 주고, 처리 상태(대기/완료)를 표시합니다.

콘텐츠 검색 (Content Search): 특정 키워드가 포함된 메시지나 채팅방을 검색하는 기능입니다.

콘텐츠 제재 버튼 (Action Buttons): 문제가 되는 메시지를 '삭제'하거나 해당 채팅방을 '폐쇄'하는 버튼입니다.

관리자 액션

사용자들이 신고한 내용을 확인하고 사실관계를 파악합니다.

욕설, 비방, 광고 등 부적절한 메시지를 삭제합니다.

문제가 지속되는 채팅방을 강제 폐쇄합니다.

개발 포인트 (Logic)

신고 시스템 연동: 사용자가 앱 내에서 신고하면 해당 내용이 관리자 페이지의 신고 내역 목록에 자동으로 추가되는 시스템이 필요합니다.

콘텐츠 검색 기능: 대용량 메시지 데이터베이스에서 특정 키워드를 효율적으로 검색할 수 있도록 전문 검색 엔진(예: Elasticsearch) 도입을 고려할 수 있습니다.

데이터 보존 정책: 삭제된 데이터는 법적 분쟁을 대비하여 즉시 영구 삭제하는 대신, 일정 기간 보관 후 삭제(Soft Delete)하는 로직을 구현합니다.

4. 활동/포인트 관리 (Activity/Point Management)

사용자의 활동 점수 및 향후 도입될 포인트 시스템을 관리합니다.

UI 컴포넌트

활동 점수 가중치 설정 (Weight Configuration): '총 활동 시간', '총 대화 횟수' 등 각 활동 지표가 점수에 얼마나 반영될지 가중치를 설정하는 입력 필드입니다.

포인트 지급/회수 도구 (Point Adjustment Tool): 특정 사용자에게 수동으로 포인트를 지급하거나 회수할 수 있는 입력 폼입니다. (사유 입력 필수)

포인트 로그 (Point Log): 모든 포인트의 지급 및 사용 내역을 시간순으로 기록한 목록입니다.

관리자 액션

서비스 정책에 맞게 활동 점수 계산 방식을 수정합니다.

이벤트 보상 등으로 특정 사용자에게 보너스 포인트를 지급합니다.

시스템 어뷰징 사용자에게서 포인트를 회수합니다.

개발 포인트 (Logic)

포인트 정책 설정: 관리자가 설정한 가중치에 따라 모든 유저의 활동 점수를 재계산하는 로직을 구현합니다.

수동 지급/차감 API: 관리자의 포인트 조정 요청을 처리하고, 모든 과정을 로그로 기록하는 API가 필요합니다.

트랜잭션 관리: 포인트 지급/회수 과정에서 오류가 발생할 경우, 데이터가 꼬이지 않도록 원상 복구(Rollback)하는 트랜잭션 처리가 필수적입니다.