

작품 계획서

작성일	2025-09-16
작품명	SIGMA
팀명	사사오입
팀원	김희영, 이창석, 임현성



목포해양대학교
MOKPO NATIONAL MARITIME UNIVERSITY

목 차

1. 개요 및 배경	3
2. 기존 사례	4
3. 주요 기능	6
4. 핵심 구현 기술	7
5. 시스템 구성도	8
6. 개발 일정	9
7. 팀원별 역할	10

1. 개요 및 배경

1-1. 개요

본 프로젝트는 대학 구성원을 위한 스마트 상호작용 플랫폼 '**SIGMA(Smart Interactive Group for Members of Academia)**'의 핵심 기능으로, 학생들이 교내 정보를 쉽고 편리하게 얻을 수 있도록 돕는 AI 기반 캠퍼스 챗봇을 개발하는 것을 목표로 합니다. 웹과 모바일 앱 등 익숙한 인터페이스를 통해 사용자는 학사 일정, 강의 정보, 시설 안내 등 다양한 질문에 실시간으로 답변을 받을 수 있습니다. 본 챗봇은 단순한 FAQ 기능을 넘어 교내 데이터베이스 및 포털 시스템과 연동하여 개인화된 맞춤형 정보를 제공함으로써, 학생 생활의 편의성을 획기적으로 향상시키고 나아가 교수 및 행정 인력까지 아우르는 대학 정보 허브로 발전할 수 있습니다.

1-2. 배경

오늘날 대학생들은 수강 신청, 장학금, 행정 절차 등 학교 생활에 필요한 수많은 정보를 여러 채널에서 찾아야 하는 불편함을 겪고 있습니다. 교내 포털, 학과 사이트, 학과 사무실, 학생 커뮤니티 등 정보가 분산되어 있어 원하는 정보를 신속하게 얻기 어렵습니다. 특히 신입생이나 교환학생은 학교 시스템에 익숙하지 않아 정보 탐색에 불필요한 시간을 소모하는 경우가 많습니다. 또한 기존의 정보 제공 방식은 주로 공지사항 게시나 FAQ 문

서 검색에 의존하고 있어, 실시간 상호작용이나 개인화된 서비스에는 한계가 존재합니다.
이에 따라 AI 챗봇을 중심으로 한 통합 플랫폼 구축의 필요성이 점점 더 커지고 있습니다.

2. 기존 사례

2-1. 유사 사례

기존 시장에는 다양한 형태의 챗봇이 존재하지만, 대부분은 특정 기능에만 국한되거나 범용적인 지식만을 제공하는 한계가 있습니다. 이를 바탕으로 '캠퍼스 챗봇'이 지향해야 할 방향을 명확히 설정할 수 있습니다.

서울대학교 SNU Bot: 교내 행정 정보와 공지사항 제공에 특화되어 있으나, 기능이 제한적이고 학사 변동 사항이나 개인화된 데이터에는 대응하지 못합니다. 규칙 기반 응답에 의존하기 때문에 확장성과 유연성이 부족합니다.

Kakao i 챗봇 기반 학과 챗봇: 키워드 매칭 중심의 단순 FAQ 자동화로, 심층 질의 처리에 한계가 있습니다. 학과별로 분산되어 있어 시스템 파편화 문제도 존재합니다.

MIT's AskMIT: 특정 서비스(도서관, IT 지원 등)별 전문성은 확보했지만, 서로 다른 서비스 간 연계가 부족하여 통합적 사용자 경험을 제공하지 못합니다.

범용 AI (ChatGPT, Gemini 등): 뛰어난 자연어 처리 능력을 보유했으나, 특정 대학의 실시간 데이터에는 접근할 수 없습니다. 따라서 실제 학사 일정이나 교내 시설 관련 질문에 부정확하거나 무응답을 줄 수밖에 없습니다.

2-2. 차별점

저희의 '**캠퍼스 챗봇(SIGMA)**'은 이러한 한계를 극복하기 위해 다음과 같은 차별적 가치를 제공합니다.

첫째, 실시간 기관 데이터 연동입니다. 학사 시스템, 도서관 DB, 식당 메뉴 관리 시스템 등 학교 내부 데이터베이스와 직접 연계하여 최신 정보를 제공합니다. 이는 범용 AI나 규칙 기반 챗봇이 결코 제공할 수 없는 기능입니다.

둘째, 개인화된 맞춤형 서비스입니다. 학번과 사용자 프로필을 기반으로, 개인 시간표·수강 신청 일정·장학금 기회 등 개인화된 정보를 선제적으로 제공합니다. 이는 학생의 정보 탐색 시간을 최소화하고 만족도를 극대화합니다.

셋째, 사용자 친화적인 통합 플랫폼입니다. 웹, 모바일 앱, 카카오톡 등 학생들이 자주 사용하는 플랫폼과 연동하여 하나의 챗봇으로 모든 학사·행정 정보를 해결할 수 있도록 합니다. 분산된 챗봇을 오갈 필요가 없으며, 장기적으로는 교수·행정 인력까지 포함하는 대학 구성원 전체의 통합 플랫폼으로 확장 가능합니다.

3. 주요 기능

3-1. 교내 정보 제공

학사 일정 안내: 개강, 수강 신청, 성적 입력/조회 기간 등 학사 일정을 교내 캘린더와 연동하여 제공합니다. 사용자가 *"이번 학기 개강일이 언제야?"*라고 물으면 정확한 날짜를 알려주며, 원할 경우 스마트폰 캘린더에 일정을 자동 등록할 수 있습니다.

수업 및 강의 정보: 과목명 또는 교수명 검색 시 강의실, 시간표, 교수 연락처 등 상세 정보를 제공합니다. 예를 들어 *"소프트웨어 공학 수업은 몇 시에 어디서 해?"*라는 질문에 즉시 답변을 제공합니다.

3-2. 교내 생활 지원

도서관 등 시설 안내: 도서관 열람실 좌석 현황, 도서 대출/반납 상태, 학생 식당 메뉴 등 실시간 데이터를 제공합니다. 예를 들어 *"지금 도서관 3열람실에 자리 있어?"*라는 질문에 실시간 좌석 현황을 안내합니다.

FAQ 응답: 휴학·복학, 장학금 신청, 동아리 가입 절차 등 자주 묻는 질문을 지식 기반 (knowledge base)을 활용하여 신속하고 정확하게 답변합니다. 이를 통해 학사·행정 담당부의 반복 민원을 경감시킵니다.

3-3. 개인화 서비스

맞춤형 알림: 도서 반납 기한 임박, 수강 신청 일정 알림 등, 사용자 학번·관심사에 따라 푸시 알림을 제공합니다.

3-4. 멀티 플랫폼 지원

다중 인터페이스 제공: 카카오톡, 모바일 앱(Android), 웹 기반 인터페이스를 모두 지원합니다. 카카오톡을 통해 별도 앱 설치 없이 간편 접근이 가능하며, 모바일 앱과 웹에서는 보다 풍부한 시각적·기능적 경험을 제공합니다.

4. 핵심 구현 기술

4-1. 프론트엔드:

React: 웹 인터페이스

React Native: 모바일 앱

4-2. 백엔드 / API 게이트웨이:

Python

FastAPI: API 서버, 인증/인가, 요청 라우팅, 응답 포매팅 등

Redis: 세션/캐시 관리

4-3. 대화 관리:

Rasa: NLU(자연어 이해, 의도 분류/엔티티 추출), DM(Dialogue Management, 대화 흐름 제어), Fallback(HuggingFace 모델 / GPT API 활용)

4-4. 데이터 계층:

PostgreSQL: 정형 데이터

pgvector: Postgres 확장, 벡터 검색 지원

4-5. 지능형 처리:

LLM: OpenAPI GPT / KoGPT (fallback, 답변 요약/정제, Text-to-SQL 보조)

임베딩 생성기: OpenAI Embeddings / Ko(SimCSE)

4-6. 커스텀 로직:

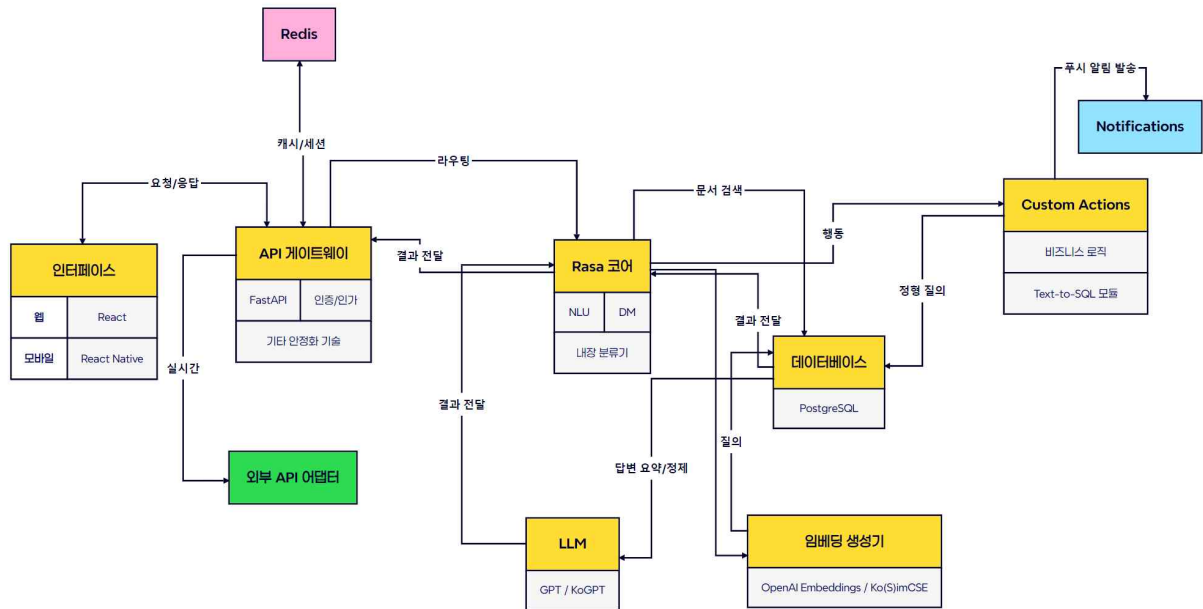
Custom Actions (Python): Text-to-SQL 변환 모듈(규칙 + LLM), 비즈니스 처리 로직

4-7. 부가 기능:

Notifications: FCM/APNs 등 푸시 서비스

외부 API 어댑터: 실시간 데이터 연동

5. 시스템 구성도 (그림으로 작성)



6. 개발 일정

주자	개발 내용
1~4주	조사 및 설계 <ul style="list-style-type: none"> - 팀 역할 분담 및 요구사항 정의 - React/React Native, FastAPI, Rasa, PostgreSQL, Redis, LLM API 기능 조사 - DB 스키마 및 아키텍처 설계 완료
5주	UI 기본 틀 완성 <ul style="list-style-type: none"> - React/React Native로 기본 화면 제작 - FastAPI 게이트웨이와 연결 테스트
6주	API 게이트웨이 구축 <ul style="list-style-type: none"> - 인증/인가(JWT) 적용 - Redis 세션 관리 - 요청/응답 표준화 로직 구현
7주	Rasa Core 초기화 <ul style="list-style-type: none"> - 기본 Intent 분류/엔티티 추출 - 단순 FAQ 대화 플로우 구현
8주	DB 연동

	<ul style="list-style-type: none"> - Custom Actions → PostgreSQL CRUD 구현 - 학사/수업/도서관 샘플 데이터 반영
9주	문서 검색 기능 <ul style="list-style-type: none"> - Embedding Generator(OpenAI/KoSimCSE) 연동 - pgvector 기반 검색 API 구성
10주	LLM Fallback 적용 <ul style="list-style-type: none"> - Rasa 분류기 신뢰도 낮을 때 GPT/KoGPT 호출 - 답변 요약/정제 로직 추가
11주	외부 API 연동 <ul style="list-style-type: none"> - 학사/도서관/식당 API 어댑터 구현 - FastAPI 통해 데이터 전달
12주	Custom Actions 고도화 <ul style="list-style-type: none"> - Text-to-SQL 모듈 구현 (규칙 + LLM 혼합) - 복잡 질의 처리
13주	알림 기능 구현 <ul style="list-style-type: none"> - Notifications(FCM 등) 연동 - 학사 일정·도서 반납 기한 푸시 발송

14주	통합 테스트 - UI~Gateway~Rasa~DB~외부 API 전체 테스트 - 버그 수정, 성능 점검
15주	발표 & 문서화 - 최종 시연 준비 - 보고서/발표 자료 정리 및 제출

7. 팀원별 역할

이름	역할
김희영	코어 담당 (AI & 챗봇 엔진): 학습/튜닝, LLM 연동 등 Python 관련 작업
이창석	인터페이스 담당(UI/UX & 게이트웨이): UI 구현, API 및 보조 기술 연결
임현성	데이터베이스 & 데이터 관리 담당: 스키마, 데이터 수집/적재, 최적화 등