
Software Requirements Specification

for

SKKU of Course

Version 1.0 approved

Prepared by Heesu Kim, Sungjin Park, Cheyung Park

Group 2

06/01/2025

Table of Contents

Table of Contents	2
1. Introduction	3
1.1 Purpose	3
1.2 Document Conventions	3
1.3 Intended Audience and Reading Suggestions	3
1.4 Product Scope	4
1.5 References	5
2. Overall Description	5
2.1 Product Perspective	5
2.2 Product Functions	6
2.3 User Classes and Characteristics	7
2.4 Operating Environment	8
2.5 Design and Implementation Constraints	9
2.6 User Documentation	10
2.7 Assumptions and Dependencies	10
3. External Interface Requirements	11
3.1 User Interfaces	11
3.2 Hardware Interfaces	19
3.3 Software Interfaces	20
3.4 Communications Interfaces	20
3.5 Functional Requirements	21
4. System Features	24
4.1 강의 정보 조회 및 검색 기능	24
4.2 강의 추천 기능	25
5. Other Nonfunctional Requirements	26
5.1 Performance Requirements	26
5.2 Safety Requirements	26
5.3 Security Requirements	27
5.4 Software Quality Attributes	27
5.5 Business Rules	27
Appendix A: Relevant Diagrams	28

1. Introduction

1.1 Purpose

본 문서는 성균관대학교 재학생을 위한 “LLM 기반 수강신청 도우미 챗봇 시스템”(이하 본 제품)의 소프트웨어 요구사항을 정의한다. 본 제품은 학기마다 학생들이 겪는 수강신청 과정의 정보 탐색의 불편함, 그리고 선택의 어려움을 해소하고자 만들어진 대화형 언어모델 기반 챗봇 서비스이다.

본 제품은 기존의 수강신청 과정에서 사용자들이 경험하는 여러 문제점을 개선 및 완화하고자 한다.

강의계획서, TO, 강의평, 교강사 정보와 같은 중요한 정보가 여러 플랫폼에 분산되어 있어서 기존 시스템 밑에서는 사용자가 직접 정보를 수집해서 비교 분석해야만 했다. 본 제품은 이러한 정보를 체계적으로 수집해 데이터베이스를 구축하고, 대규모 언어 모델(LLM)을 활용해 사용자들에게 친숙한 인터페이스로 접근할 수 있도록 설계해서 학생 개개인에게 맞춤형 수업 추천과 시간표 설계를 지원하는 것이 목표이다.

본 SRS는 시스템의 전체 기능 중 **사용자 인터페이스, 데이터 수집 및 처리 모듈, LLM 기반 질의응답 시스템, 개인화 추천 알고리즘**, 그리고 **챗봇 통합 기능**을 포함하여 정의한다.

1.2 Document Conventions

이 문서는 다음과 같은 표기 및 작성 규칙을 따른다:

- **굵은 글씨체**는 핵심 시스템 구성요소, 기능, 또는 사용자 행위를 나타낸다.
- 요구사항은 우선순위에 따라 다음 태그를 갖는다: [필수], [권장], [선택]

1.3 Intended Audience and Reading Suggestions

이 문서는 다음과 같은 독자층을 대상으로 한다:

- **개발자**: 기능 구현 및 기술 설계 시 참조한다.

- **프로젝트 관리자:** 범위 정의, 일정 수립 및 자원 할당에 활용한다.
- **테스터:** 기능 검증을 위한 테스트 케이스 작성 기준으로 활용한다.
- **디자이너:** UI/UX 요구사항을 이해하고 설계에 반영한다.
- **이해관계자:** 기능 요구사항 및 시스템 목표를 검토한다.

1.4 Product Scope

본 제품은 수강신청 시 필요한 강의 정보 탐색, 비교, 선택, 저장 기능을 통합한 AI 기반 도우미 챗봇이다.

시스템의 주요 목표는 다음과 같다:

- 첫째, 흩어진 수강신청 관련 정보를 하나의 플랫폼에 통합하여 제공함으로써, 학생들이 여러 웹사이트를 번갈아 방문하지 않고도 필요한 모든 정보를 신속하게 접근하고 비교할 수 있는 환경을 제공한다.
- 둘째, 자연어 기반의 대화형 인터페이스를 통해 수강신청이라는 복잡한 작업을 누구나 쉽게 접근할 수 있도록 한다. 사용자는 키워드 검색이 아닌, "이번 학기 영상처리 수업 중 추천해줄 만한 거 있어?"와 같은 질문을 입력함으로써 필요한 정보를 직관적으로 획득할 수 있다.
- 셋째, 학생의 이수 과목, 전공, 진로 계획, 시간표 조건 등 다양한 학사 정보를 기반으로 맞춤형 강의 추천을 제공하여, 보다 전략적인 수강 설계와 학업계획 수립이 가능하도록 돕는다.

이러한 기능들을 통합적으로 제공하는 본 시스템은 사용자가 학교 계정으로 로그인한 후, 간단한 대화를 통해 복잡한 수강신청 정보를 탐색하고 정리할 수 있도록 설계된다. 더불어 성균관대학교 학사정보시스템(GLS), 에브리타임, 교수 연구 정보 등의 외부 시스템으로부터 데이터를 수집하고 연동하며, 향후 학사 시스템 통합이나 교내 포털과의 연계 확장을 고려하여 모듈형 구조로 개발된다. 결과적으로 본 제품은 수강신청 시즌에 발생하는 정보 격차와 선택 부담을 해소하며, 학생 개개인의 전공 이수, 졸업 요건 충족, 진로 탐색을 돕는 데 실질적인 기여를 할 수 있다. 더 나아가 디지털 학사 행정의 자동화와 스마트화에 기반이 되는 시스템으로 확장 가능성이 높다.

1.5 References

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.
New York: IEEE, 1998.

Python Software Foundation. "Python 3.10.0 Documentation." Python 3.10.0, October 4, 2021. [Online]. Available: <https://docs.python.org/3.10/>. [Accessed: June 1, 2025].

Langchain Python Documentation. [Online]. Available: [<https://python.langchain.com/>]. [Accessed: June 1, 2025].

Svelte. "Introduction." SvelteKit Documentation. [Online]. Available: <https://kit.svelte.dev/docs/introduction>. [Accessed: June 1, 2025].

Fette, I. and A. Melnikov. "The WebSocket Protocol." RFC 6455, Internet Engineering Task Force, December 2011. [Online]. Available: <https://www.rfc-editor.org/info/rfc6455>. [Accessed: June 1, 2025].

2. Overall Description

2.1 Product Perspective

본 제품의 핵심 기술은 최신 대규모 언어 모델(LLM)을 기반으로 하고 동시에 대학생 커뮤니티 플랫폼인 '에브리타임'과 성균관대학교 공식 홈페이지의 정보를 수집해서 데이터베이스의 형태로 운영해 학생들에게 맞춤형 수강신청 지원 서비스를 제공하는 것이 본 제품의 최종 목표이다. 본 제품은 독립적으로 운영되는 새로운 시스템이지만, 필요한 정보의 수집과 기존 학교 체계와의 연동을 위해 외부 시스템과 긴밀히 연동한다.

2.1.1 System Interface

본 제품은 다음과 같은 시스템 인터페이스를 사용한다.

- **성균관대학교 공식 홈페이지:** 교강사 정보, 학사 일정, 커리큘럼 내용, 평가 내용 등 수강신청 관련 정보를 얻기 위해 홈페이지 스크래핑을 통해 얻어서 내부 데이터베이스에 저장한다.

- **에브리타임**: 강의평가, 강의후기, 평가정보와 같은 학생이 작성한 정보를 얻기 위해 스크래핑을 통해 얻어서 내부 데이터베이스에 저장한다.
- **LLM엔진**: 사용자의 자연어 질의를 받아서 내부 데이터베이스에 존재하는 정보를 바탕으로 최적의 답변을 구상해서 사용자에게 전달한다.
- **내부 데이터베이스**: 수집된 정보를 저장하고 검색하기 용이하게 사용된다.

2.1.2 User Interface

본 제품의 사용자 인터페이스는 다음과 같다.

- **챗봇 인터페이스** : 학생들은 웹 브라우저로 접근해서 최근 ChatGPT와 같은 제품으로 보편화된 대화형 챗봇 인터페이스를 통해 자연어로 문답을 진행한다. 인터페이스는 간결하게 만들어 접근성을 높인다.

2.2 Product Functions

본 제품은 성균관대학교 재학생들의 수상신청 과정을 돕기 위해 다음과 같은 기능을 제공한다. 모든 기능은 LLM을 기반으로 한 대화형 인터페이스를 통해 제공되는 것을 목표로 한다.

2.2.1 강의 정보 조회 및 검색

- [필수] 특정 강의의 정보 (강의명, 교수명, 강의 시간, 강의실, 학점, 강의 계획서, 선수 과목)를 검색한다.
- [필수] 강의와 관련된 정보를 홈페이지와 에브리타임에서 수집한 정보와 연계해서 제공한다.

2.2.2 강의 추천

- [필수] 학생의 현재 상황정보(학년, 전공, 관심 분야, 진로, 선호하는 강의 방식 등)을 고려해서 맞춤형 강의를 추천한다.

- [권장] "학점 잘 주는 교양 찾아줘", "부담 없는 전공 과목 찾아줘", "졸업 필수 과목 찾아줘" 등과 같은 비정규 요청도 수행 가능하다.

2.2.3 시간표 시각화

- [필수] 학생이 관심을 표하는 강의들의 조합을 만들어 시간표로 보여준다.
- [권장] 생성된 시간표의 상세정보를 사용자가 조정할 수 있는 기능 제공한다.
- [권장] "오전 수업은 듣기 싫어", "월요일을 공강으로 만들어"와 같은 조건에 맞춰서 시간표를 생성하는 기능 지원한다.

2.2.4 개인 최적화

- [필수] 학생의 개인 관심사, 진로 등을 입력받으면 이를 토대로 강의를 추천한다.
- [권장] 관심 연구 분야를 입력받으면 관련 교수님 강의를 추천한다.

2.3 User Classes and Characteristics

본 제품의 대상 사용자는 성균관대학교 재학생이며 다음과 같이 분류가 가능하다.

2.3.1 학부생

예상 사용자의 대부분을 차지하는 그룹으로 학년, 복학여부와 같은 조건으로 추가적으로 세분화할 수 있다.

2.3.1.1 신입생

- 처음 경험하는 수강신청 시스템에 익숙하지 않을 가능성이 높다.
- 계열제 학생의 경우 전공이 정해지지 않을 수 있다.
- 원하는 강의를 어떤지 잘 모를 수 있다.

2.3.1.2 재학생

- 수강신청 시스템에 대한 일정 수준의 이해도를 가지고 있다.
- 관심 분야, 진로, 졸업요건에 대해 관심이 높을 것이며 본인이 원하는 강의가 어떤지 어느 정도 알고 있다.

2.3.1.3 복학생

- 변경된 졸업 요건, 학사 시스템에 대해 모르는 것이 있을 수 있다.
- 신입생보다는 수강신청 과정에 대한 이해도가 높으며 진로 및 관심분야를 정했을 가능성이 높다.

2.4 Operating Environment

본 제품을 안정적으로 구동하기 위해 구축되어야 하는 환경은 다음과 같다.

2.4.1 서버 환경

- **하드웨어:** 본 제품은 클라우드 기반 가상서버(VPS) 혹은 물리적 서버에서 구동되는 것을 전제로 제작되었다.
 - CPU, 메모리, 디스크 용량 및 I/O는 성균관대학교 재학생 수를 기반으로 최대 부하를 예측해 확보해야 한다.
 - 수강신청의 시기적 특성에 맞춰서 특정 기간에 스케일이 되도록 설계한다.
- **소프트웨어:**
 - 운영체제: Linux 기반 운영체제 사용을 권장한다.
 - 웹 서버: nginx, apache와 같은 안정적이고 검증된 서버를 사용한다.
 - 기타 라이브러리 및 도구 : Python 3.10 이상, FastAPI 0.115 이상, Langchain 0.3 이상, Pytorch 2.7 이상, HyperCLOVAX-SEED-Text-Instruct-3B, sqlite, 등 필요한 라이브러리를 사용한다.

2.4.2 클라이언트 환경

- **하드웨어:** 웹 브라우저로 접속이 가능한 웹 어플리케이션으로 브라우저가 구동 가능한 모든 하드웨어를 지원하는 것을 목표로한다.
- **소프트웨어:**
 - 운영체제: MS Windows, macOS, Linux, Android, iOS과 같은 모든 주요 운영체제를 지원한다.
 - 브라우저: Chrome, Edge, Firefox과 같이 최신 브라우저를 모두 지원한다.

2.5 Design and Implementation Constraints

- [필수] 사용자는 모바일/데스크탑 브라우저를 통해 동일한 사용자 경험을 얻어야 하며, 반응형 UI로 제작해야 한다.
- [필수] 클라이언트 애플리케이션은 Svelte 5.0 이상으로 작성되며, 서버와는 RESTful API로 통신한다.
- [필수] Python 버전 3.10 이상, Langchain 0.3 버전 이상, Transformers 버전 4.51 이상, Pytorch 2.7 버전 이상을 이용해 개발하며, 모든 패키지는 uv를 통해 관리한다.
- [필수] 챗봇 서버는 Langchain 프레임워크를 기반으로 구현되며, API 서버는 FastAPI 0.115 버전 이상으로 구성한다.
- [필수] 챗봇의 백엔드에는 HyperCLOVAX-SEED-Text-Instruct-3B를 4 bit 양자화 해 질의응답 모델로 사용하며, KR-SBERT-V40K-klueNLI-augSTS 임베딩 모델을 사용한다.
- [필수] 수강 신청 정보를 위한 SQLite를 사용한다.
- [필수] 대용량 문서 검색을 위한 벡터 인덱스는 FAISS(Facebook AI Similarity Search) 1.11 버전 이상을 이용해 임베딩 벡터를 구축한다.
- [필수] LLM과 벡터 인덱싱/임베딩 처리는 Linux 기반 Docker 컨테이너 상에서 동작해야 한다.

- [필수] 시스템은 최소 2GB 메모리를 확보해야 하며, GPU 서버는 VRAM 8GB를 가져야 한다.
- [필수] 개인정보 및 학사 정보는 보안상의 이유로 HTTPS 통신을 사용해야 하며, 사용자의 민감 정보는 저장하지 않는다.
- [필수] 챗봇은 다음과 같은 기능을 필수로 포함해야 한다:
 - 과목, 교수자 정보, 강의평을 포함한 통합 정보 제공
 - 자연어 기반의 챗봇 UI로 직관적인 질의응답 지원
 - 이수 과목, 전공, 시간표 기반의 개인 맞춤형 추천
 - 추천 수업 저장이 가능한 책가방 기능
- [필수] 시스템 반응 시간은 3초 이내가 되어야 하며, 5초를 초과할 경우 오류 알림을 제공한다.
- [필수] 코드 품질 유지를 위해 Git 기반 버전 관리를 하며, PR 리뷰 및 CI 테스트를 사용한다.

2.6 User Documentation

- **사용자 매뉴얼:** 브라우저로 접속하고 사용할 수 있는 기능과 명령어 예시를 제공한다. 주요 기능을 사용하는 데모 영상도 함께 제공한다.
- **하드웨어 및 소프트웨어 요구사항:** 소프트웨어를 실행하기 위해 필요한 최소 사양을 작성한다.
- **개인정보 동의 정책:** 사용자의 학교 계정을 연동하기 위해 필요한 인증 정보 및 개인 학적 정보를 이용한다는 점을 명시하고 설명한다.
- **연락처/문의 채널:** 기술적 문제나 과목 데이터 오류 시 문의할 수 있는 이메일 또는 GitHub 이슈 트래커를 제공한다.

2.7 Assumptions and Dependencies

- 사용자는 인터넷 환경에서 웹 브라우저(Chrome, Edge, Safari)를 통해 접속한다고 가정한다.
- 챗봇의 질의응답 품질은 LLM 모델(HyperCLOVAX-SEED-Text-Instruct-3B)의 성능에 의존한다.

- 초기에는 SQLite DB를 사용하지만, 동시 사용자 수가 증가하면 대용량 데이터 처리를 위한 DB로 확장될 수 있다.

3. External Interface Requirements

3.1 User Interfaces

Table 1 User Interface of Login Page

이름	학교 계정 인증을 통한 접속
목적/내용	사용자가 학교 계정의 아이디와 비밀번호를 입력해 개인 채팅 화면으로 이동할 수 있도록 한다.
입력 주체/출력대상	사용자 / 서버
범위/정확도 /오차범위	정확도: 반드시 인증된 사용자만 접근할 수 있도록 보장한다.
단위	sec (지연 시간)
시간/속도	2초 이내
기타 입출력관계	키보드 입력은 실시간으로 입력창에 반영되며, 로그인 요청 시간 초과 시 팝업창을 띄워 시각적 피드백 제공한다. 운영체제 또는 브라우저에서 제공하는 비밀번호 자동 입력 기능에 따라 입력 폼이 채워질 수 있다.

화면 형식과 구성	<p>아이디와 비밀번호 입력 폼이 중앙에 위치한다.</p> <p>비밀번호 입력 폼 우측에 비밀번호를 가릴 수 있는 버튼이 있다.</p> <p>아래는 로그인 정보를 저장하고 다시 사용할지 여부를 확인하는 체크박스가 있으며, 사용자가 선택 시 초록색 체크를 표시한다.</p> <p>하단에는 로그인 버튼을 클릭해 로그인 정보를 서버로 전송한다.</p> <p>인증된 사용자는 개인 채팅 페이지로 넘어간다.</p>
데이터 유형	문자열, 응답 상태 코드, 토큰
명령 유형	POST 요청
종료 메시지	N/A

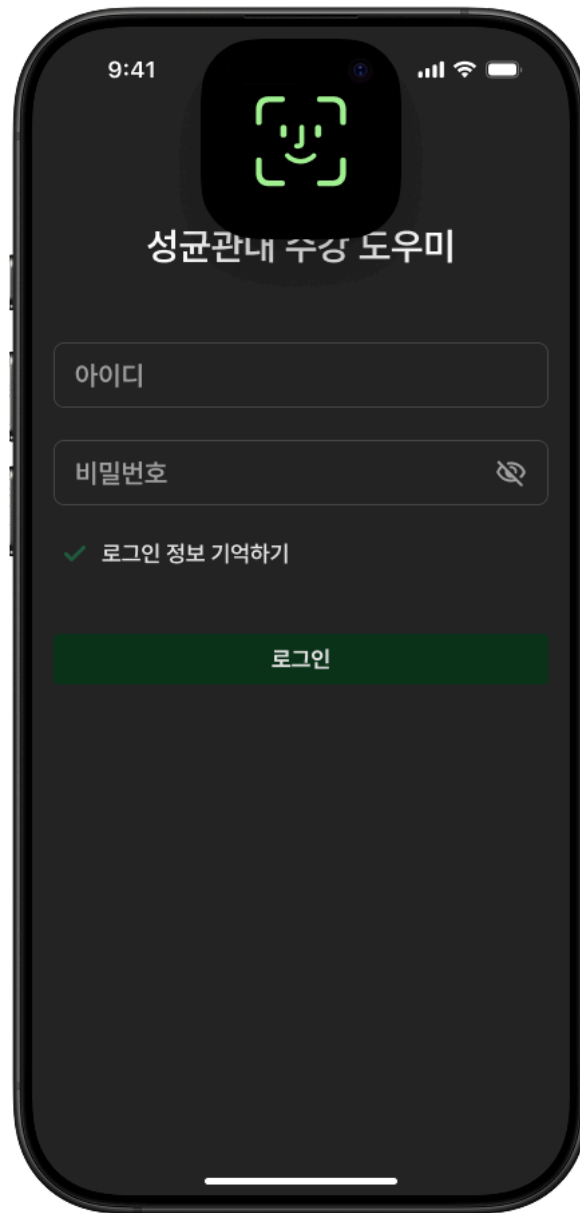


Figure 1 Login page

Table 2 User Interface of Main Page - Chatting

이름	개인 채팅 화면
----	----------

목적/내용	사용자가 텍스트 입력을 통해 챗봇으로부터 응답을 받는다.
입력 주체/출력대상	사용자 / 사용자 화면
범위/정확도 /오차범위	N/A
단위	sec (반응 시간)
시간/속도	3초 이내
기타 입출력관계	사용자가 요청한 쿼리는 즉각 화면에 표시되며, 서버에서 HTTP Streaming을 이용해 청크 단위의 텍스트를 전송할 때마다 즉각 사용자 화면에 출력한다.
화면 형식과 구성	<p>입력창: 하단에 고정된 입력창이 위치하며, 스크롤과 관계없이 일정한 크기 및 위치를 유지한다. 사용자 입력 전 "텍스트를 입력하세요"를 표시해 입력 가능하다는 점을 나타낸다.</p> <p>일반 응답: 사용자의 질문은 별도의 박스로 묶어 표시하며, 응답은 사용자 질문 아래에 출력된다.</p> <p>책가방: 사용자가 책가방에 담고 싶다고 요청할 경우, 해당하는 과목을 카드 UI로 출력한다.</p> <p>시간표: 사용자가 시간표를 요청하면 표 형식으로 책가방에 담긴 과목을 출력한다. 과목명, 장소는 각각의 시작/종료 시간에 맞게 배치한다. 모바일 화면에서는 우측으로 overflow를 허용하며, 하단에 스크롤바를 표시해 상호작용이 가능한 요소임을 보여준다.</p>

데이터 유형	문자열
명령 유형	POST 요청
종료 메시지	N/A

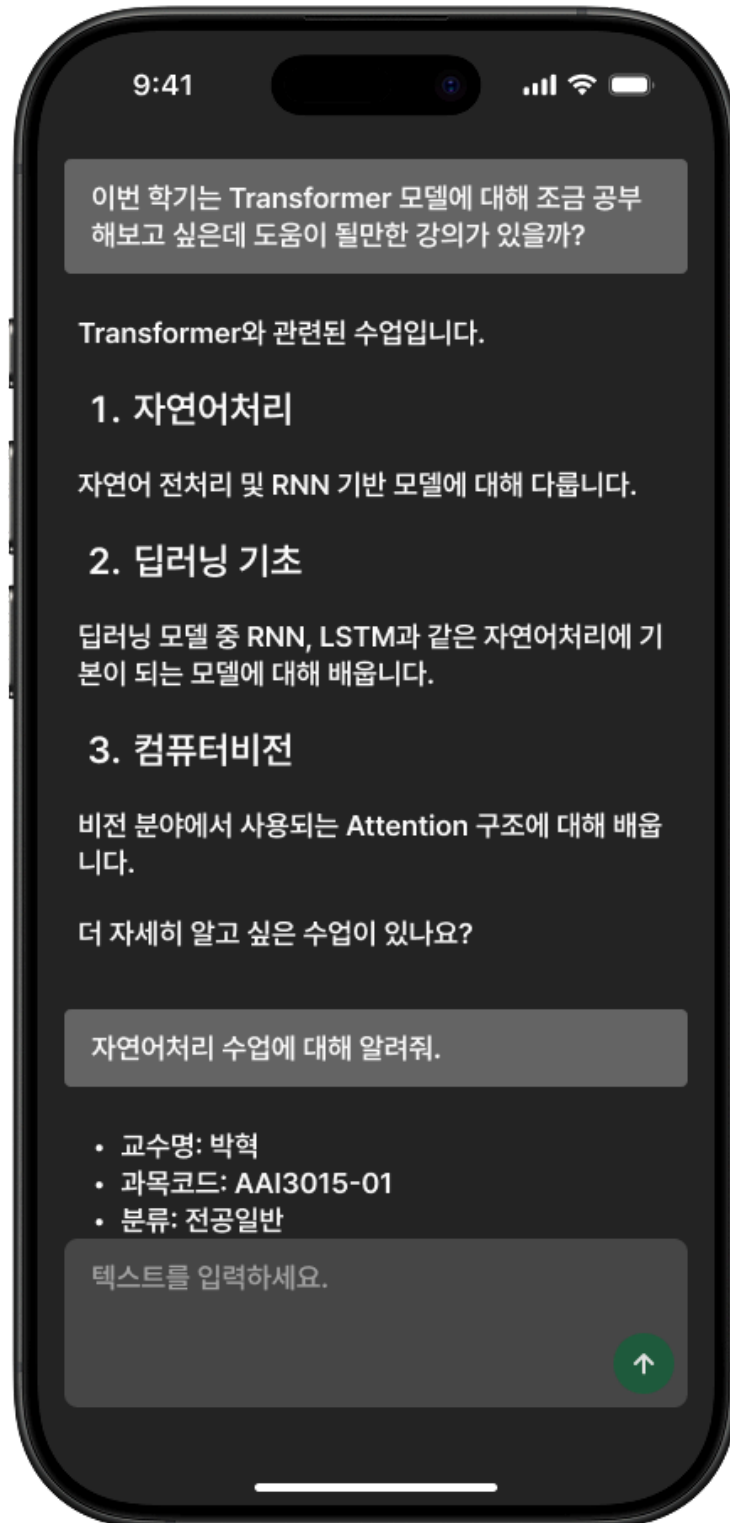


Figure 2 Main page - Chatting UI

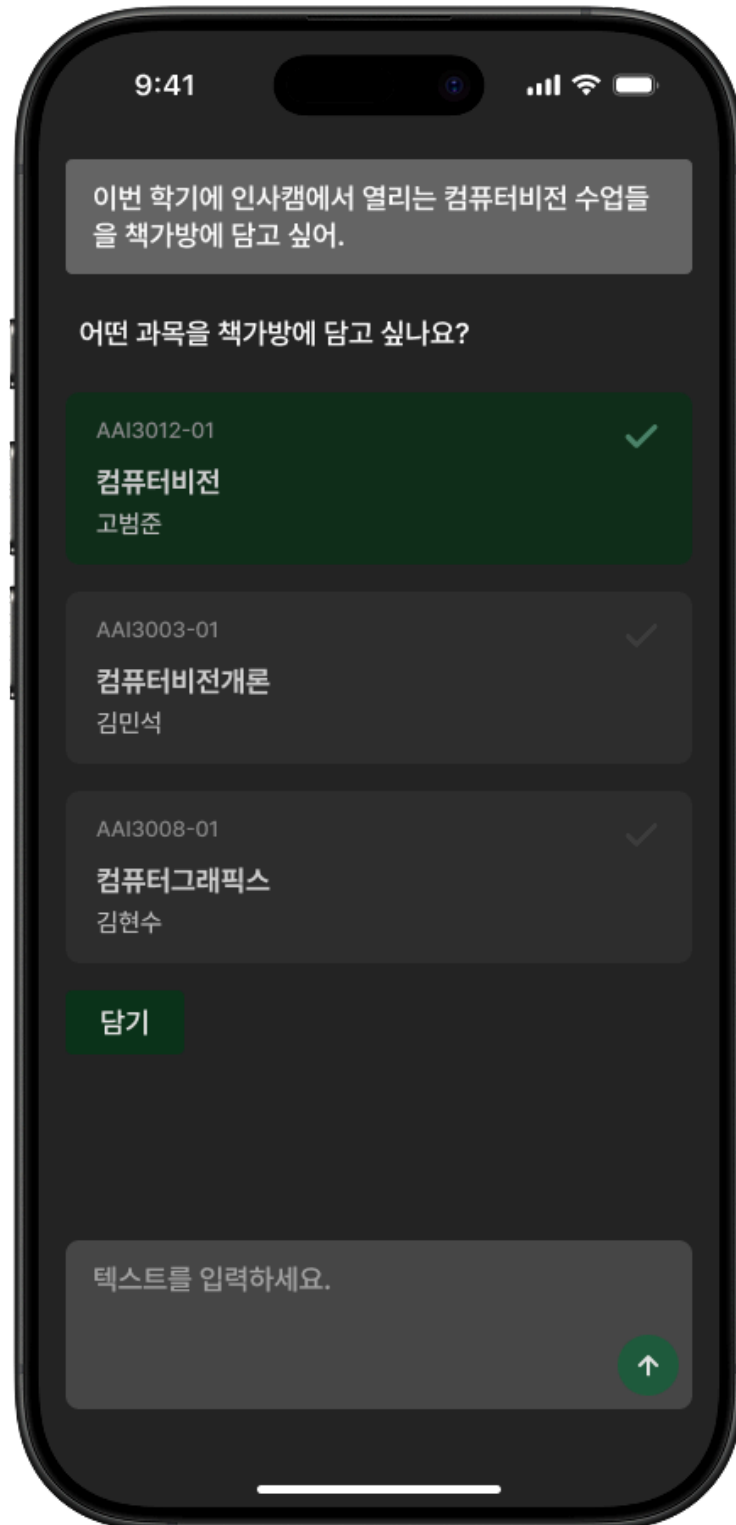


Figure 3 Main page - Card UI

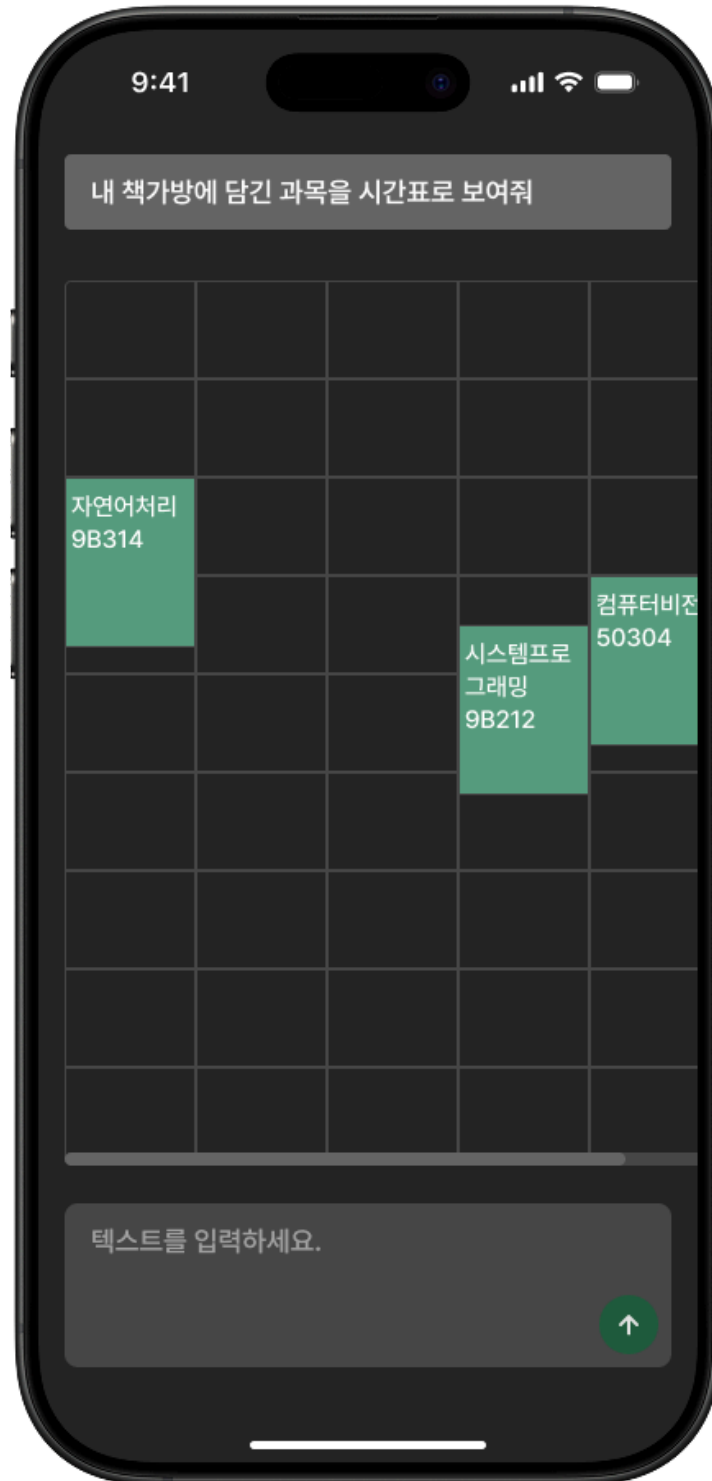


Figure 4 Main page - Table UI

3.2 Hardware Interfaces

Table 3 Hardware Interface of User Device

이름	사용자의 실행 디바이스
목적/내용	키보드, 마우스를 사용한 사용자의 입력 및 응답 출력을 위한 화면
입력 주체/출력대상	사용자 / 디바이스 화면
범위/정확도 /오차범위	정확도: N/A
단위	ms (지연 시간)
시간/속도	사용자 입력 / 요청한 입력에 대한 처리
기타 입출력관계	키보드/마우스 입력은 실시간으로 반응한다.
화면 형식과 구성	N/A
데이터 유형	픽셀
명령 유형	클릭, 키보드 입력, 스크롤, 터치
종료 메시지	N/A

3.3 Software Interfaces

- **데이터베이스 인터페이스**
 - [필수] 내부 데이터베이스는 sqlite DBMS를 사용해 필요 데이터를 관리한다.
- **언어 모델 인터페이스**
 - [필수] 언어 모델과 API 서버는 별도의 통신 프로토콜 없이 함수 호출 방식으로 직접 상호작용한다.
 - [필수] 함수 호출 결과 필요한 텍스트를 별도의 데이터 형식을 따르지 않고 원본 텍스트 그대로 반환된다.
- **대학교 홈페이지, 에브리타임 수집 인터페이스**
 - [필수] API 서버와 동일한 환경에서 주기적으로 파이썬 스크립트를 실행시켜 성균관대학교 홈페이지와 에브리타임 강의 평가 페이지에서 정보를 수집한다. 이 때 사용하는 라이브러리는 Selenium으로 Headless Web Browser를 사용하고 운영자의 로그인 정보를 사용해서 에브리타임 강의 평가 페이지에 접근한다.
 - [필수] 주기적으로 업데이트 스크립트를 실행시켜 최신화하되 수강신청 기간이 아닐 때에는 실행시키지 않는다. 수강신청 기간은 2월부터 3월까지, 8월부터 9월까지로 정의하고 수강신청 기간 도중에는 1주일에 1번 스크립트를 실행시킨다.
 - [권장] 만약 사용자로부터 정보가 최신화되어 있지 않다고 문의가 들어오면 관리자가 임의로 스크립트를 실행시켜 데이터베이스를 최신화한다.

3.4 Communications Interfaces

- **사용자(프론트엔드) ↔ 서버(백엔드) 소통**
 - [필수] 사용자와 챗봇 서버 간의 실시간 양방향 통신을 위해 웹소켓(WebSockets, RFC 6455) 프로토콜을 우선적으로 사용한다.

- [필수] 모든 통신 채널은 TLS(Transport Layer Security) 1.2 이상을 사용하여 종단 간 암호화를 보장해야 한다. 서버는 신뢰할 수 있는 인증 기관(CA)에서 발급한 SSL/TLS 인증서를 사용해야 한다.
- [필수] 사용자 요청 및 서버 응답의 주요 데이터 형식은 JSON을 사용한다. 메시지에는 다음 필드를 포함할 수 있다:
 - messageId: 각 메시지의 고유 식별자
 - timestamp: 메시지 생성 시간 (ISO 8601 형식)
 - type: 메시지 유형 (userQuery, botResponse, systemNotification, error)
 - sender: 메시지 발신자 (user, bot)
 - text: 실제 메시지 내용 (사용자 질문 텍스트, 챗봇 답변 객체, 오류 정보 객체)
- 백엔드 ↔ LLM 소통
 - [필수] 별도의 통신 프로토콜 없이 함수 호출 방식으로 직접 상호작용한다.
 - [필수] 함수 호출 결과 필요한 텍스트를 별도의 데이터 형식을 따르지 않고 원본 텍스트 그대로 반환된다.
- 서버 ↔ 학교 홈페이지/에브리타임 소통
 - [필수] 서버에서 파이썬 크롤러 스크립트를 구동해 주기적으로 크롤링한다.
 - [필수] 챗봇 서버의 데이터 수집 모듈과 대학교 공식 홈페이지 및 에브리타임 서버 간의 통신은 HTTP/HTTPS 프로토콜을 사용한다.
 - [필수] HTML 형태로 된 문서를 가지고 필요한 정보를 추출해 JSON 포맷으로 저장했다가 데이터베이스에 저장한다.

3.5 Functional Requirements

3.5.1 로그인

Table 4 Use Case - 로그인

Use Case Name	로그인
Actor	회원가입이 완료된 모든 사용자
Description	서비스 사용을 위해 ID와 비밀번호를 입력해서 로그인한다.
Normal Course	<ol style="list-style-type: none"> 1. 본인의 ID와 비밀번호를 입력한다. 2. ID와 비밀번호가 데이터베이스에 저장된 것과 일치하지 않으면 다시 시도한다 3. ID와 비밀번호가 데이터베이스에 저장된 것과 일치하면 다음 페이지로 넘어간다
Precondition	<p>서비스 제공 서버가 정상적으로 작동되어야 한다.</p> <p>서버와 사용자 사이 네트워크 연결이 원활하게 이어져 있어야 한다.</p>
Postcondition	N/A
Assumptions	N/A

3.5.2 회원가입

Table 5 Use Case - 회원가입

Use Case Name	회원가입
Actor	회원가입이 완료되지 않은 모든 사용자
Description	학교 계정을 본 서비스와 연동하는 과정을 거친다.

Normal Course	<ol style="list-style-type: none"> 1. 본인의 ID와 비밀번호를 입력한다. 2. ID를 데이터베이스에 조회하고 중복되지 않으면 회원 정보를 데이터베이스에 등록한다.
Precondition	<p>서비스 제공 서버가 정상적으로 작동되어야 한다.</p> <p>서버와 사용자 사이 네트워크 연결이 원활하게 이어져 있어야 한다.</p>
Postcondition	데이터베이스가 새로운 사용자의 정보로 업데이트된다.
Assumptions	N/A

3.5.3 강의 추천

Table 6 Use Case - 강의 추천

Use Case Name	강의 추천
Actor	로그인한 모든 사용자
Description	사용자가 질의를 채팅 인터페이스를 통해 입력하면 적절한 답변을 반환한다.
Normal Course	<ol style="list-style-type: none"> 1. 사용자가 요구사항을 입력하면 API 서버로 json 형식으로 전달된다. 2. 사용자가 작성한 관심 분야, 진로, 선호 요일, 선호 시간, 기타 조건을 바탕으로 데이터베이스에 저장된 정보를 조회한다. 3. 조회된 정보를 바탕으로 최초에 사용자가 입력한 텍스트와 함께 언어모델에 입력해서 텍스트를 반환받는다. 4. 반환된 텍스트를 json 형식으로 사용자의 client에 전송한다.

	5. 받은 텍스트 데이터를 채팅 인터페이스의 형식으로 출력한다.
Precondition	서비스 제공 서버가 정상적으로 작동되어야 한다. 서버와 사용자 사이 네트워크 연결이 원활하게 이어져 있어야 한다. 사용자가 성공적으로 로그인을 해서 채팅 인터페이스를 접속해 있다.
Postcondition	N/A
Assumptions	학교에서 개설되는 모든 강의 정보가 데이터베이스에 저장되어 있다.

4. System Features

본 섹션은 "2.2 제품 기능"에서 기술된 주요 기능들을 보다 상세하게 설명하며, 각 기능이 사용자에게 제공하는 핵심적인 가치와 세부적인 동작 방식을 명시한다.

4.1 강의 정보 조회 및 검색 기능

본 기능은 사용자가 필요로 하는 강의 정보를 효과적으로 찾고 이해할 수 있도록 지원한다.

4.1.1 상세 정보 제공

본 시스템은 사용자가 채팅 인터페이스로 특정 강의에 대한 상세 정보를 검색하고 조회하는 기능을 제공해야 한다. 이 때 상세 정보는 강의명, 교수명, 학수 번호, 학점, 강의 시간, 강의실, 강의 계획서의 요약 내용, 선수 과목, 개설 학과, 평가 방식 등이 존재한다.

4.1.2 정보 통합 기능

대학교 홈페이지 및 에브리타임에서 수집한 정보를 통합해서 일관된 형식으로 사용자에게 정보를 제공해야 한다. 정보의 출시를 표시해서 이후 사용자가 직접 확인할 수 있도록 해서 신뢰도를 높일 수 있다.

4.1.3 복합 조건 검색

단순하게 단일 조건에 대한 검색 결과를 반환하는 것 뿐만이 아니라 다중 조건(금요일에 수업이 없으면서 오후수업만 존재하는 수업)을 복합적으로 고려해서 결과를 반환할 수 있어야 한다.

4.1.4 검색 결과 표시

반환된 강의 정보는 일관된 인터페이스로 표시되어야 하고 별도의 해석에 노력이 필요하지 않게 시간표와 강의 정보 표의 형식으로 출력해야 한다.

4.2 강의 추천 기능

4.2.1 개인 맞춤형 추천

본 시스템은 사용자가 개인 프로필 정보를 제공할시 이를 기반으로 최적화된 조건을 산정해서 최적화된 강의를 추천하는 기능을 제공한다.

4.2.2 자연어 기반 추천 요청

사용자의 자연어 기반 질문에 대해서 적절하게 대답하는 기능을 제공해야 한다. “학생 평가가 좋은 교양 과목을 찾아줘”, “내가 졸업하려면 어떤 수업을 들어야 해?”, “학점 받기 쉬운 수업을 추천해줘”와 같은 질문을 받으면 알맞는 조건을 설정해서 요구한 수업을 검색하고 반환해야 한다.

4.2.3 졸업 요건 맞춤 추천

각 학생마다 다른 졸업 요건을 모두 파악해서 필요한 수업 및 학점을 만족하기 위해 필요한 수업을 찾는 기능을 제공해야 한다. 사용자가 먼저 요청하지 않더라도 문의한 조건 안에서 시간표를 편성할 때 졸업 요건을 충족하는 수업 위주로 우선적으로 편성하는 기능을 제공해야 한다.

4.2.4 진로 및 관심 분야 맞춤 추천

사용자가 입력한 진로와 관심 분야를 가지고 연관성이 높은 수업을 추천하는 기능을 제공해야 한다.

사용자가 “나는 인공지능에 관심이 있고 졸업 이후에도 자연어 처리와 언어 모델을 연구하고 싶어”와 같은 입력을 하면 이를 토대로 수업계획서나 강의하는 교수님 연구 분야와 같은 데이터와 결합해서 추천해야 한다.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- 평균 응답 시간은 3초 이내로 제한된다.
- 200명 이상의 동시 접속 사용자에게 안정적인 서비스를 제공해야 하며, 수강신청 기간에는 서버 스케일링을 통해 성능을 유지한다.
- 질의응답 응답은 Streaming 방식으로 제공되어 사용자 체감 속도를 개선한다.

5.2 Safety Requirements

- 시스템 중단 또는 서버 장애 발생 시 사용자는 정보 유실 없이 다시 질의할 수 있어야 하며, 재접속 시 이전 대화 내용 일부를 확인할 수 있도록 로그 기록 시스템을 구축한다.
- 테스트 환경과 운영 환경을 분리하여 개발 중 오류로 인한 운영 장애를 방지한다.

5.3 Security Requirements

- 모든 통신은 HTTPS 프로토콜로 암호화되어야 하며, 사용자 인증에는 JWT(Json Web Token) 기반 토큰 인증 방식을 사용한다.
- 사용자의 민감한 정보(성적, 학번 등)는 저장하지 않으며, 서버 로그에도 기록되지 않도록 한다.

5.4 Software Quality Attributes

- **사용성(Usability)**: 간결한 UI와 자연어 기반 인터페이스로 초보자도 쉽게 사용할 수 있어야 한다.
- **신뢰성(Reliability)**: 수강신청 시즌 중에도 중단 없이 서비스를 제공해야 하며, 서비스 중단 발생 시 자동 복구 메커니즘을 내장한다.
- **유지보수성(Maintainability)**: 모듈형 구조를 통해 각 기능의 독립적 업데이트 및 버그 수정이 용이해야 한다.
- **이식성(Portability)**: 다양한 운영체제와 웹 브라우저 환경에서도 일관된 사용자 경험을 제공할 수 있어야 한다. 및 브라우저에서 동일한 경험을 제공할 수 있어야 한다.

5.5 Business Rules

- 사용자는 반드시 학교 계정으로 로그인해야 하며, 학교 구성원이 아닌 경우 접근 권한이 제한된다.
- 학생 개인 정보는 맞춤형 추천 외의 용도로 사용될 수 없다.
- 과목 정보 및 강의 평가는 주기적으로 업데이트되며, 사용자는 이를 최신 상태로 유지하는 데 동의한 것으로 간주한다.

Appendix A: Relevant Diagrams

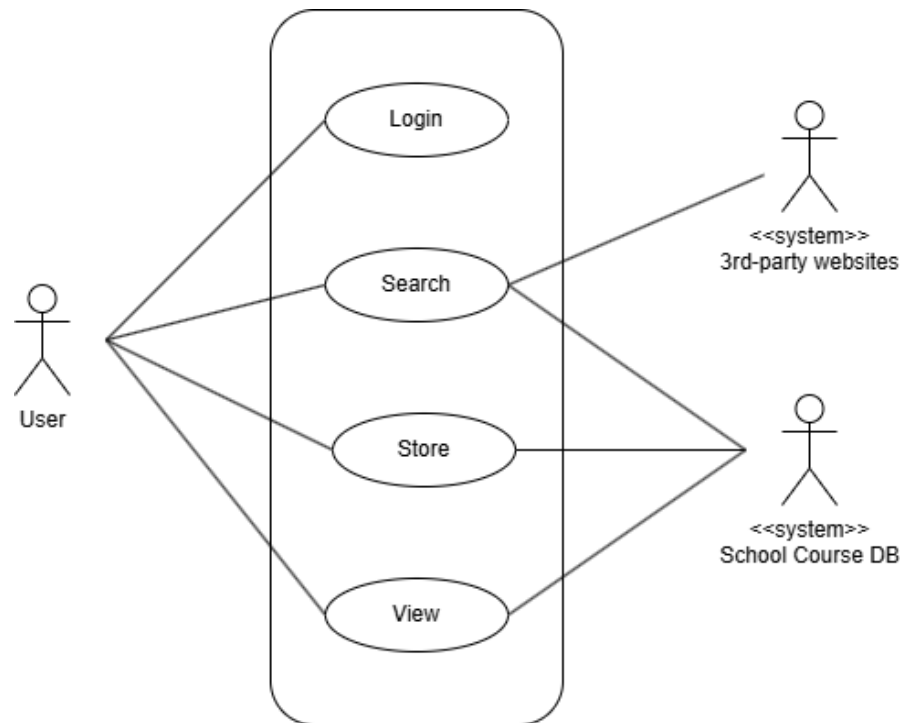


Figure 5 Use case diagram

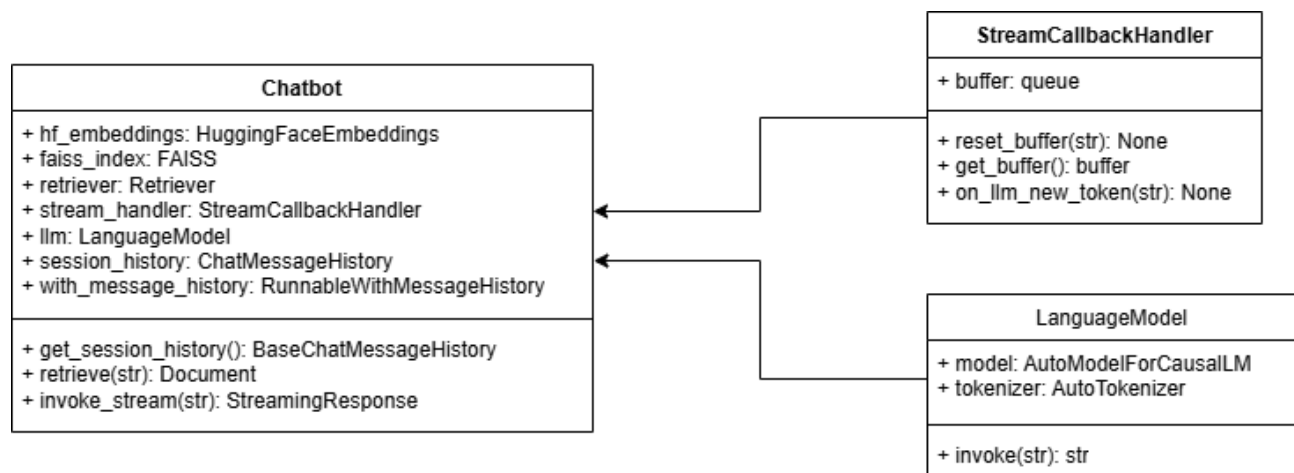


Figure 6 Class diagram of Chatbot Server

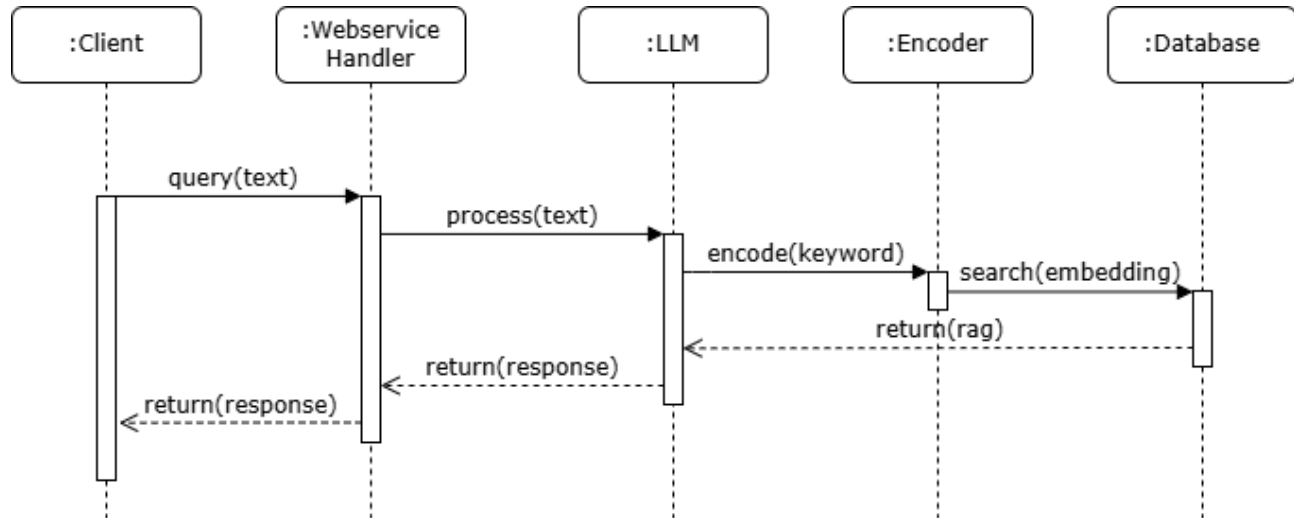


Figure 7 Sequence Diagram for Query Process

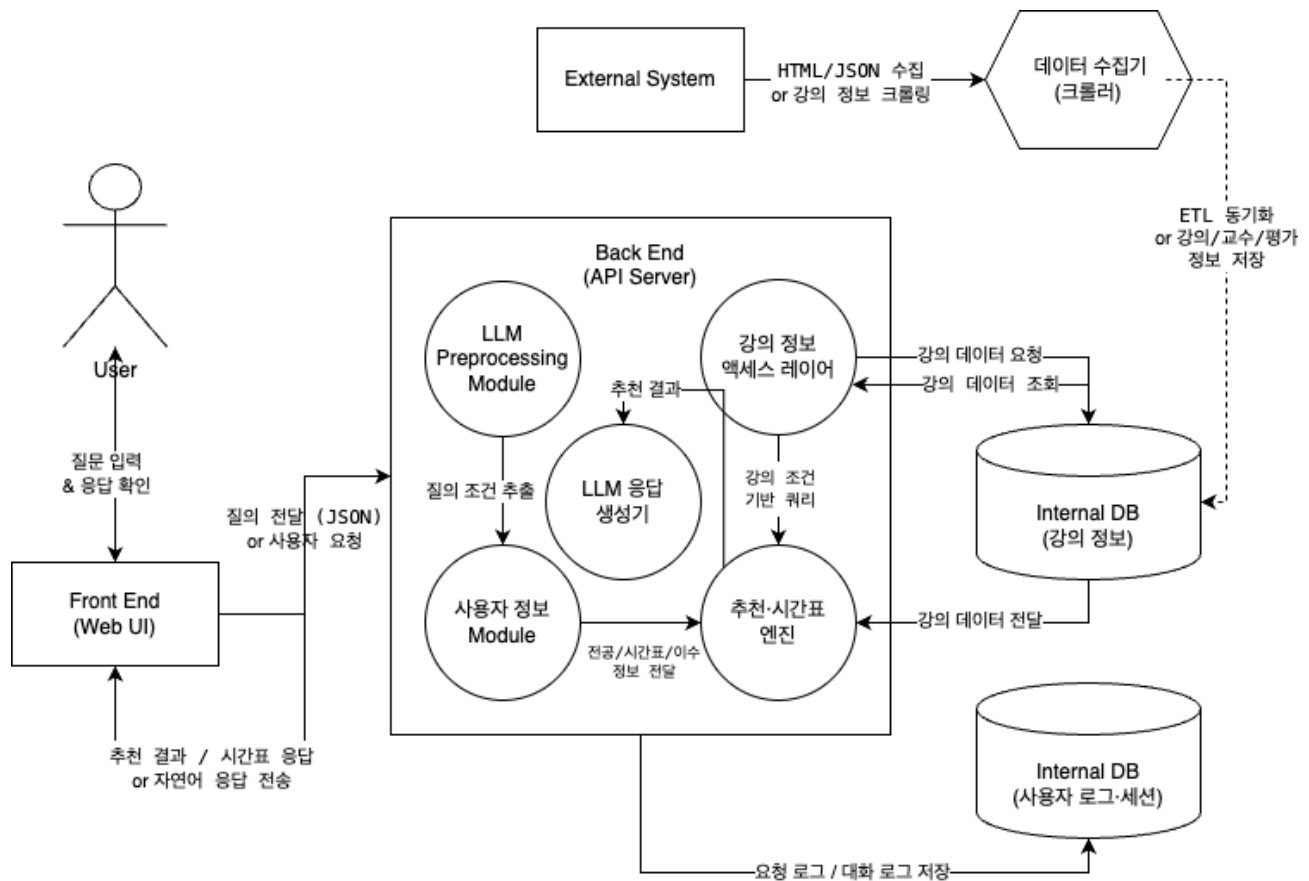


Figure 8 Data Flow Diagram