

PJT명	최종 프로젝트 기획 및 기술 설계	
단계	[소프트웨어 아키텍처 설계]	
진행일자	2025.12.05	
예상 구현 시간	필수기능	8H
	추가기능	
	심화기능	

1. 목표

- 요구사항을 세분화하여 관리할 수 있다.
- 정의된 요구사항을 바탕으로 한 필요한 기술 요구사항을 정의할 수 있다.
- 사용자 관점의 UI/UX를 구성하고, 화면 간 이동 흐름을 시각화 할 수 있다.
- 다양한 다이어그램을 활용하여 프로젝트를 설계를 진행할 수 있다.

2. 준비사항

1) 제공 파일

- 없음

2) 사용 도구

- WBS & Gantt 차트 작성 도구
 - Excel, Google Sheets 등
- UML Diagram 작성 도구
 - Diagrams.net, SequenceDiagram.org 등
- 와이어프레임 작성 도구
 - Figma 등

3. 작업 순서

- 1) 팀원과 함께 필요 제출 산출물 내역을 확인하고, GitLab에 프로젝트를 생성한다.
 - 프로젝트 이름은 10-pjt로 지정한다.
 - 각 반 담당 강사님을 Maintainer로 설정한다.
- 2) 팀원과 함께 만들고자 하는 프로젝트의 사용자 관점에서의 핵심 시나리오를 구성한다.
- 3) 프로젝트 설계를 위한 기술 스택을 결정하고, 일정을 계획한다.
- 4) 프로젝트 설계를 진행하고, 완성된 문서와 다이어그램을 정리한다.
- 5) 제출물을 GitLab에 업로드 후 README를 작성한다.
- 6) README 작성이 완료되면 팀원과 새로운 아이디어를 논의하고 정리한다.
- 7) 제출 기한에 맞춰 모든 산출물이 GitLab에 업로드 될 수 있도록 한다.

4. 요구사항

본격적으로 프로젝트에 착수하기에 앞서 프로젝트 계획을 수립하고자 한다. 자신이 진행하고자 하는 프로젝트의 요구사항을 정리하고, 요구사항 구현을 위한 계획을 수립해보자. 이후 프로젝트 설계를 위한 여러가지 다이어그램을 작성해보자.

처음부터 모든것을 완벽하게 설계하기 보다는, 프로젝트 진행을 위해 필요한 내용들로 인식하고 작업하자. 내용이 완벽하지 않더라도 추후 업데이트를 통해 프로젝트의 모습을 그려나간다고 생각하고, 모든 요구사항을 마무리하지 못하여도 제출할 수 있도록 한다.

1) 프로젝트 기획

아이디어를 구체화하여 프로젝트의 방향성 및 계획을 수립하고자 한다.
다음 내용을 참고하여 시나리오와 요구사항을 정리하고, 프로젝트 진행 계획을 문서화해보자.

- 작성을 위해 사용할 도구는 자유롭게 결정할 수 있다.
- 프로젝트에 사용할 데이터셋을 바탕으로 아이디어를 구체화한다.
 - 필요하다면 기타 통계 자료를 활용하여도 좋다.

A. 아이디어 구체화 및 요구사항 정의

다음 내용을 참고하여, 이전에 기획한 프로젝트 기획서를 바탕으로 아이디어를 구체화해보고, 이를 바탕으로 요구사항을 정의해보자.

- 문제 정의
 - “왜 이 프로젝트가 필요한가?”를 바탕으로 문제 정의
 - 실제로 불편하거나 개선이 필요한 문제를 명확하게 정의
- 목표 설정
 - 여러가지 질문을 제시하고 답변해 봄으로써 아이디어를 구체화할 수 있음
 - 누구를 위한? 무엇을? 어떻게? 왜 필요한가? 이미 존재하는 서비스와의 차별점은?
 - SMART 원칙 활용 가능
 - **Specific**: 명확한 목표 설정
 - **Measurable**: 측정 가능한 수치를 바탕으로 목표 달성 여부 판단이 가능해야 함
 - **Achievable**: 현실적으로 달성 가능한 목표 설정
 - **Relevant**: 목표가 전반적인 목적과 비전과 연결되어야 함
 - **Time-bound**: 명확한 마감 기한이 있어야 함
- 구체적인 목표를 바탕으로 사용자 관점의 요구사항 분석
 - 기능적 요구사항: 시스템을 통해 수행 가능한 기능
 - 비기능적 요구사항: 품질, 성능, 보안 등
- 서비스 기획과 요구사항 목록을 최종 산출물로 제출할 것

● 작성 예시

구분	작성 내용
서비스명	SSAFY Library
서비스 개요	AI 기반 도서 분석과 창작 지원 통합 솔루션 서비스
주요 기능	API를 활용한 도서 조회 기능 도서 독후감 작성 기능 도서 독후감을 바탕으로 한 AI 생성 이미지 제작 기능
주요 페이지 구성	홈 / 게시판 / 프로필 / 도서 목록 / 게시글

번호	요구사항명	요구사항 상세	우선순위
기능적 요구사항			
F01	도서 목록 출력	주어진 도서 데이터에 따라 화면이 다르게 출력되도록 구현	높음
F02	검색 기능	사용자가 입력한 검색어를 바탕으로 특정 데이터만 출력되도록 구현	높음
F03	회원 가입 기능	서비스 사용을 위해 사용자가 자신의 정보를 등록하는 기능	높음
F04	로그인 기능	등록된 데이터를 바탕으로 사용자가 자신을 인증하는 기능	높음
F05	프로필 기능	사용자가 자신의 관심사를 표현할 수 있는 프로필을 만드는 기능	보통
...
비기능적 요구사항			
NF01	활성 사용자	최대 20명의 사용자가 동시에 안정적으로 활용 가능해야 함	높음
NF02	화면 전환	원할한 인터넷 환경에서 모든 화면 전환 작업은 6초 이내에 이뤄져야 함	높음
...

B. 진행 계획 수립

다음 내용을 참고하여, 정의된 요구사항을 기준으로 구체적인 일정을 계획하고 문서화해보자.

- Work Breakdown Structure (WBS)
 - 작업 분해 구조
 - 프로젝트의 전체 범위를 계층적이고 체계적인 방식으로 더 작고 관리하기 쉬운 구성 요소로 분해하는 도구
 - 계층적 구조를 가지고 있음
 - Level 1: 프로젝트 (최종 산출물)
 - Level 2: 주요 단계
 - Level 3: 대규모 작업
 - Level N: 작업 패키지, 더 이상 분해되지 않는 가장 작은 단위까지 분해
- Gantt Chart
 - 프로젝트 전체 일정을 확인할 수 있는 바 형태의 도구
 - WBS에서 분해된 작업 패키지에 예상 기간, 선행 관계 등을 고려하여 작성
 - 개발 뿐만 아니라 이미 진행한 아이디어 구상, 본 명세서에 포함된 기획 및 설계 요구사항도 포함하여 작성할 것

- 작성 예시 (Google Sheets 템플릿 활용)

WBS & Gantt 차트

프로젝트 이름

AI 활용 프로젝트

팀원

홍길동

[illegible]

C. 데이터 탐색

실제 서비스 제공을 위해서 필요한 데이터셋을 찾아서 정리해보자.

- 추천 시스템의 구현을 위한 목적성에 알맞은 데이터셋 수집
- 외부 API의 사용 가능성 여부 조사
 - 서로 다른 데이터셋을 사용할 경우, 데이터 정제 및 표준화를 고려
- 수집한 데이터의 활용 방안을 같이 기재
- 참고 가능한 데이터셋 제공 API 목록
 - TMDB API
<https://developer.themoviedb.org/reference/intro/getting-started>
 - IMDb Developer API
<https://developer.imdb.com/>
 - KMDb Open API
<https://www.kmdb.or.kr/info/api/apiList>
 - 공공데이터포털 API
<https://www.data.go.kr/tcs/dss/selectDataSetList.do>
 - 금융상품통합비교공시 API
<https://finlife.fss.or.kr/finlife/main/contents.do?menuNo=700029>
 - 알라딘 API
https://docs.google.com/document/d/1mX-WxuoGs8Hy-QalhHcvuV17n50uGI2Sg_GHofgiePE/edit?tab=t.0
 - 국가 자료 종합 목록
<https://www.nl.go.kr/NL/contents/N31101030400.do>
 - 네이버 도서검색 API
<https://developers.naver.com/docs/serviceapi/search/book/book.md>

2) 프로젝트 설계

구체적인 프로젝트의 기획이 완료되었으면 완성된 프로젝트의 청사진을 그려보자. 프로젝트 구조를 묘사하는 다양한 다이어그램을 알아보고, 화면을 설계하고, 화면의 기능을 위한 서버 엔드포인트를 정리해보자.

- 설계를 통해 얻고자 하는 것
 - 구조적 명확성 확보를 통한 효율적 개발 환경
 - 구현 전 단계에서 위험과 결함을 사전에 알아보기
 - 서로 같은 목표를 가지고 있음을 시각적 문서로 확인
- 완전한 설계를 목표로 하지 않음을 유의할 것
 - 프로젝트 진행 중 발생하는 아이디어나 오류에 대하여 업데이트 가능

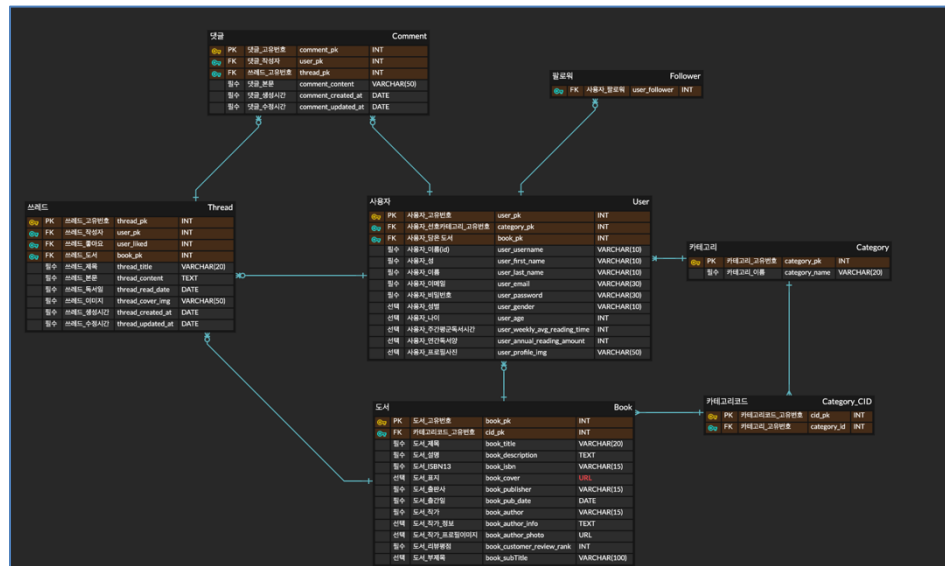
A. 다이어그램 작성

다음 다이어그램을 찾아보고, 프로젝트를 묘사하는 다이어그램을 작성해보자.

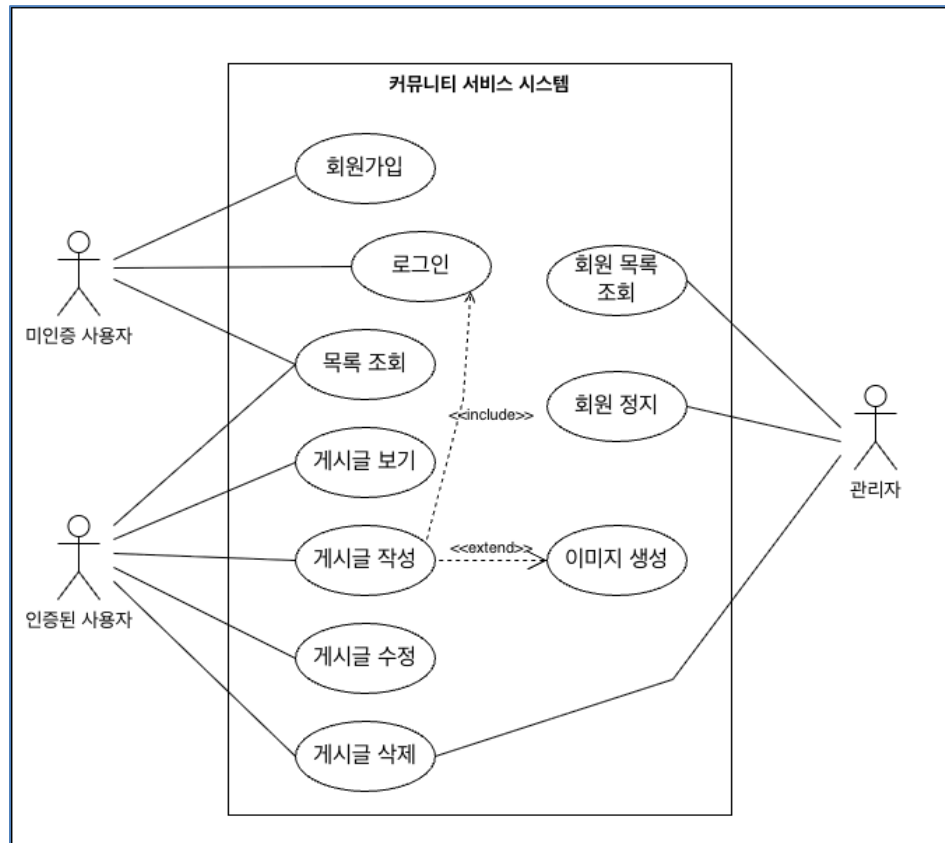
- Entity Relationship Diagram (ERD)
 - 개체-관계 모델, 개별 데이터의 속성을 정의하고, 개체별로 분류 후 각 개체 간의 관계를 명확히 표현한 다이어그램
 - 참고: <https://www.ibm.com/kr-ko/think/topics/entity-relationship-diagram>
- Use Case Diagram
 - 사용자와 시스템 사이 가능한 상호 작용을 시각적으로 묘사한 다이어그램
 - 각각의 사용자(Actor)가 목표를 달성하기 위해 시스템과 상호 작용하는 방식을, 사례(Use Case)를 기준으로 표현
 - 참고: <https://www.ibm.com/docs/ko/rational-soft-arch/10.0?topic=diagrams-use-case>
- 그 외 추가하고 싶은 Diagram을 자유롭게 조사 후 추가
 - Sequence Diagram 등

- 작성 예시

- ERD 예시



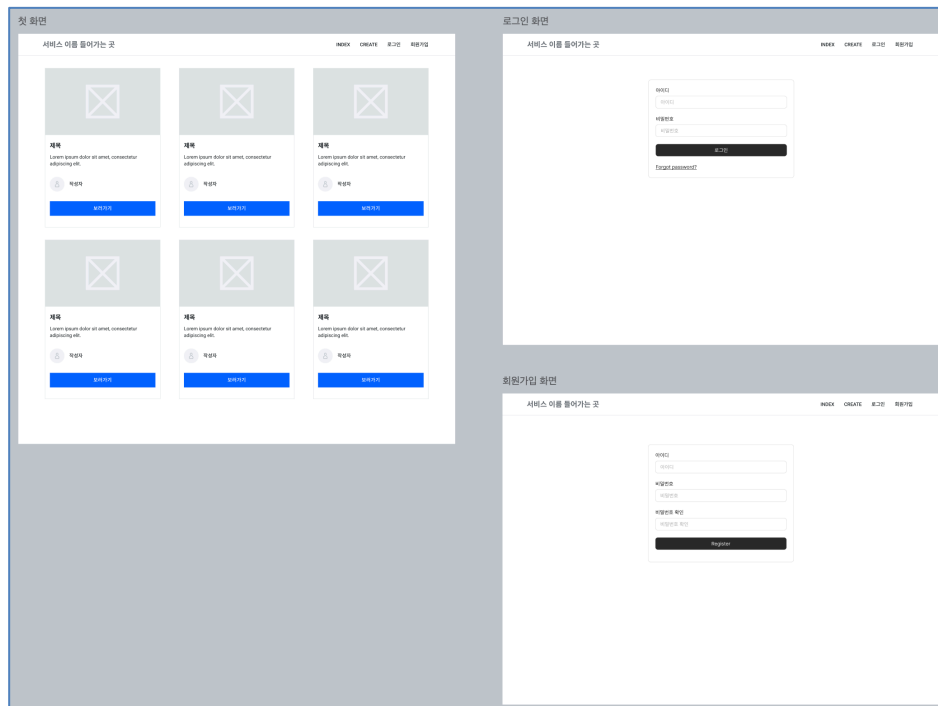
- Use Case Diagram 예시



B. UI/UX 설계

서비스 전체의 화면 구조를 설계하여 사용자 흐름을 정리해보자.

- 사용자가 상호작용할 수 있는 화면을 정리 및 나열
- 화면에서 화면 간 이동 흐름을 묘사
 - 참고: <https://www.figma.com/ko-kr/resource-library/what-is-wireframing/>
- 예시



C. API 명세 작성

필요한 화면을 기준으로 서버에서 제공해야 하는 엔드포인트 구조를 설계해보자.

- 각 기능별로 API를 분리하여 작성
- Method, URL, 설명, 요청/응답, 인증 필요 여부를 포함
 - AJAX 기술을 활용하는 것을 전제로 작성
 - RESTful 원칙을 준수하여 작성할 것
- OpenAPI Specification을 활용하여도 무방
 - 참고: <https://docs.tosspayments.com/resources/glossary/oas>

- 작성 예시

- Word 작성 예시

1. 금융상품 리스트 조회 API

항목	내용
Method	GET
Endpoint	/api/products/
설명	금융감독원 API 기반 정기예금 상품 목록 전체 조회
Request	없음
Response (예시)	[{ "id": 1, "bank_name": "싸피은행", "product_name": "싸피 예금", "base_rate": 3.2 }, ...]
인증 필요 여부	불필요 (공개 API)

1. 도서 목록 리스트 조회 API

항목	내용
Method	GET
Endpoint	/api/v1/books/
설명	보유 도서 데이터 전체 목록 조회
Request	없음
Response (예시)	[{ "id": 1, "title": "소년이 온다", "author": "한강", "isbn": "9788936434120" }, ...]
인증 필요 여부	불필요 (공개 API)

- Swagger UI 예시

금융상품

GET

/api/products/

금융상품 리스트 조회

금융감독원 API 기반 정기예금 상품 목록 전체 조회

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Successful operation <small>Media type</small> <div>application/json</div> <small>Controls Accept header.</small> <small>Example Value</small> <small>Schema</small>	No links

도서 조회

GET

/api/v1/books/

도서 목록 조회

보유 도서 데이터 전체 목록 조회

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Successful operation <small>Media type</small> <div>application/json</div> <small>Controls Accept header.</small> <small>Example Value</small> <small>Schema</small>	No links

5. 참고자료

- IBM: 개체 관계 다이어그램이란 무엇인가요?
<https://www.ibm.com/kr-ko/think/topics/entity-relationship-diagram>
- IBM: 유스 케이스 다이어그램
<https://www.ibm.com/docs/ko/rational-soft-arch/10.0?topic=diagrams-use-case>
- Figma: 와이어프레임 제작이란 무엇인가요?
<https://www.figma.com/ko-kr/resource-library/what-is-wireframing/>
- Toss: OAS (Open API Specification)
<https://docs.tosspayments.com/resources/glossary/oas>
- Swagger UI
<https://swagger.io/tools/swagger-ui/>

6. 결과

제출 기한은 프로젝트 진행 당일 18시까지이므로 제출 기한을 지킬 수 있도록 한다. 제출은 GitLab을 통해서 이뤄진다.

- 산출물과 제출
 - 단계별로 작성 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점을 상세히 기록한 README.md
 - 완성된 프로젝트 기획 / 설계 문서
 - 프로젝트 이름은 10-pjt로 지정 및 각자 제출
 - 한 명의 GitLab 계정에 Git 저장소 Push 및 작업
 - 나머지 인원은 제출 시 해당 저장소를 Fork 하여 제출
 - 각 반 담당 강사님을 Maintainer로 설정

- 끝 -