

# 생성형AI 활용 개발 실무

Lecture 04 생성형AI 활용 개발

황선희



동양미래대학교  
DONGYANG MIRAE UNIVERSITY

# 생성형AI활용 경진대회 안내

---

- **프로젝트 계획 발표 (발표시간: 5~10분)**
  - 10월 16일, 10월 23일
  - 자료 제출 마감 (10월 16일 수업시간 전까지)
- **프로젝트 결과물 발표 (발표시간: 5~10분)**
  - 11월 20일, 27일
  - 자료 제출 마감 (11월 20일 수업시간 전까지)
- **경진대회 (발표시간: 5~10분)**
  - 11월 27일 오후 5시~6시, 우승후보자 대상 발표
  - 최우수상 30만원(1명), 우수상 20만원(2명), 장려상 10만원(5명)
- **발표 순서는 임의로 선정될 예정**

# 2023년도 경진대회 소개

- ChatGPT 기반 SW개발 프로젝트
  - 소프트웨어 개발 과정 전체에 ChatGPT를 활용하여 프로그램을 제작

**PM(기획자) 업무영역**

( W H M )

## ChatGPT의 활용

서비스 네임

MRD 작성

ChatGPT를 이용해 상담이나 진단 도우미로 활용하여 이를 관련 질문에 대답하고 응답을 기반으로 사용자를 적절한 의사나 병원으로 연내하는 서비스의 목적을 정리해줘

의료 상담 및 진단 도우미로 활용되는 서비스에 적합한 이름을 선택해 놓는 전문성, 안전성, 친근한 등을 고려하는 것이 좋습니다. 아래는 ChatGPT를 기반으로 한 의료 상담 및 진단 도우미 서비스에 어울리는 몇 가지 이름 제안입니다:

- MediVoteChat:
  - 의료(Medical)와 투표(Vote)를 결합하여 의료 상담에서 시제본을 도움을 주는 서비스를 나타냅니다.
- SymptomGuide:
  - 증상(Symptom)과 안내(Guide)의 결합으로, 증상 기반의 안내를 강조합니다.
- DocAssistHub:
  - 의사(Doctor)와 도움(Hub)이 핵심(Hub)으로, 의료 상담을 효과적으로 지원하는 중심성을 의미합니다.
- HealthQChat:
  - 의료 상담과 질병(Quality)에 초점을 맞춘 서비스입니다.

Message ChatGPT... ChatGPT can make mistakes. Consider checking important information.

서비스 네임

MRD 작성

**기획/개발자 업무영역**

시나브로

## 기획/개발자 업무영역

기획력 상승을 위한 문장 끝말잇기 및 AI 이미지 생성 플랫폼

팀 명	시나브로	학 과	컴퓨터소프트웨어공학과
팀 원	김유진		

1. MRD (Market Requirements Document)

1.1 서비스 소개

항목	내용	비고
개요	- 단어 혹은 문장에 대한 끝말잇기 - 사용자가 입력한 단어 혹은 문장을 기반으로 시를 짓거나 그림을 그리는 활동 - 활용 예 : 단어 또는 문장, 이미지	입력 : 단어 또는 문장 출현 : 단어 또는 문장, 이미지
개발 목적	- 언어 능력을 향상시킬 목적으로 단어 혹은 문장을 기반으로 시를 짓거나 그림을 그리는 활동 - AI 이미지 생성 기능을 통해 입체적인 문장에 대해 학습 능력 강화, 시각적인 축면에서 창의성 증진, 창의적 사고를 발전시키기 위함	언어 능력을 향상시킬 목적으로 단어 혹은 문장을 기반으로 시를 짓거나 그림을 그리는 활동 - AI 이미지 생성 기능을 통해 입체적인 문장에 대해 학습 능력 강화, 시각적인 축면에서 창의성 증진, 창의적 사고를 발전시키기 위함
타겟 고객	어린이 및 청소년, 교육기관, 다국어 사용자 및 디문화 환경	

1.2 시장 분석

- 아동 및 청소년의 언어 능력 저하  
언어능력은 축서방언 일정하게 연관 되어있다. 하지만, 문화체육관광부와 축자 개발 연구 보고서에 따르면 청소년기의 축서량은 급격히 하강하는 추세를 보이고 있다. 이러한 현상은 초등학교 5학년을 지나면서부터 귀찮다는 정서가 겹겹 커지기 때문이다. 이 귀찮다는 정서의 가장 큰 원인은 과제, 시험 연계 폭주감, 축서의 강요로 나타난다. 이 뿐만 아니라 5세 미이하에서도 언어 능력 저하 현상이 일

**개발자 업무영역**

시나브로

단어 또는 문장을 입력하세요

요리를 먹는 것은 언제나 즐겁고 감사한 일이다!

창의력을 활용해 창작 활동을 즐기고 기록해보세요.

창의력을 활용해 창작 활동을 즐기고 기록해보세요.

## 아이디어 기획 및 시장조사 (MRD, PRD)

\*Market requirement document  
\*Product requirement document

## 개발문서 작성 (기획/결과)

## 소스코드 개발 (프론트/백엔드, API활용 등)

# ChatGPT를 활용한 작곡서비스 개발

**특징:** 사용자가 자연스럽게 대화하듯 플랫폼과 상호작용하면서 음악 작곡에 대한 개인적인 요구사항을 전달할 수 있습니다.

- **잘 하는 영역**
  - 자연어 이해 및 생성에 강한 모델로, 사용자와의 자연스러운 대화를 가능하게 합니다.
  - 다양한 음악 스타일 및 톤을 이해하고 표현할 수 있습니다.
- **한계점**
  - 음악 작곡은 음악 이론, 감성 등 다양한 영역을 포괄하는데, ChatGPT가 모든 영역에 대해 전문적인 지식을 가지고 있지는 않습니다. 따라서 모델의 출력물은 가끔 예측할 수 없는 결과를 보일 수 있습니다.

8마디  $\frac{3}{4}$ 박자 f key의 밝은 느낌의 왈츠곡을 만들어줘!



X:1

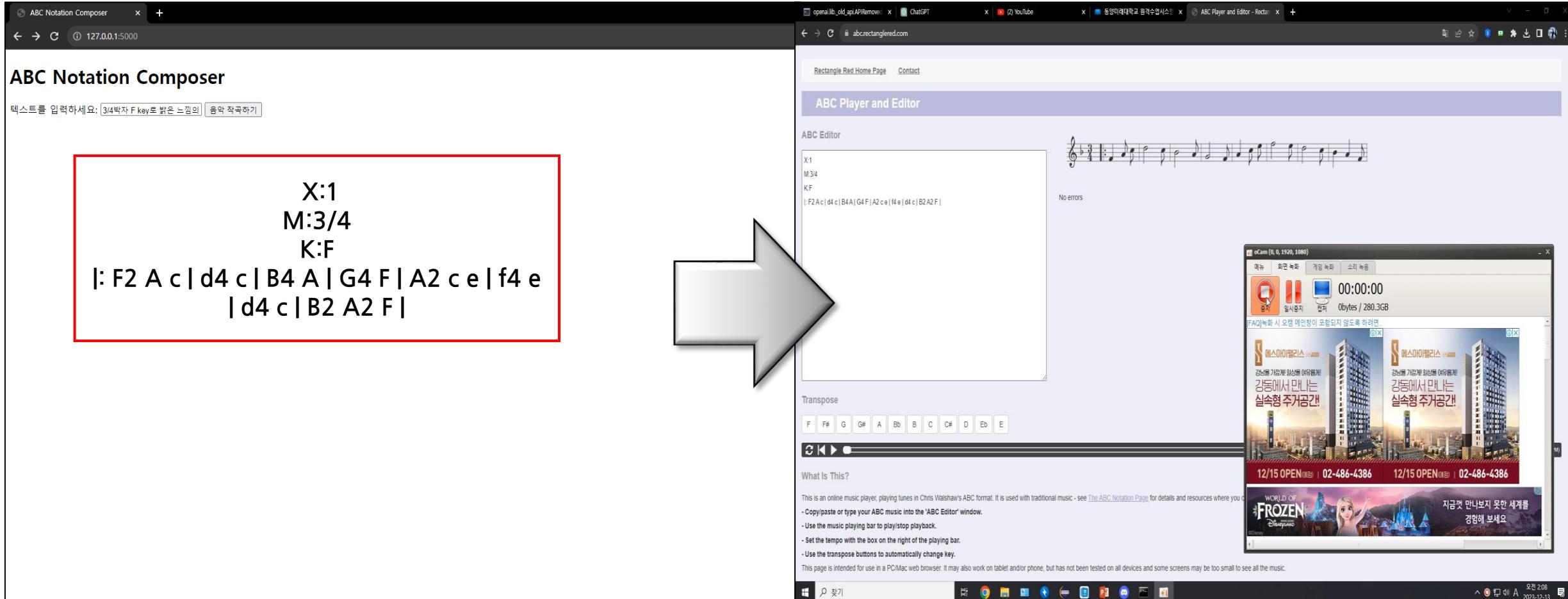
M:3/4

K:F

I: F2 A c | d4 c | B4 A |  
G4 F | A2 c e | f4 e | d4 c  
| B2 A2 F |



# 작곡 및 음악 연주 시연 예시



텍스트 기반 작곡 서비스

코드기반 음악생성 기술 활용



# ChatGPT를 활용한 비트코인 자동매매 SW개발

- ChatGPT의 few-shot learning 기능을 활용하여, 사용자가 입력한 거래 조건과 시장 데이터를 바탕으로 최적의 거래 전략을 도출합니다. 이를 통해 사용자는 복잡한 시장 분석 없이 자동매매를 설정할 수 있습니다.
- 프로그램의 사용자 가이드 작성 시, ChatGPT를 통해 자연스럽고 이해하기 쉬운 설명을 제공하여 사용자 경험을 향상시킵니다.
- 사용자의 피드백과 질문을 ChatGPT에 입력하여, 자동매매 프로그램의 기능 개선과 사용자 지원을 위한 인사이트를 얻습니다.

"사용자가 설정한 비트코인 거래 조건은 다음과 같습니다:

1. 24시간 동안의 가격 상승률이 5% 이상일 때 매수.
2. 매수 가격보다 10% 상승했을 때 매도.
3. 매수 가격보다 5% 하락했을 때 손절매.

그리고 현재 시장 데이터는 다음과 같습니다:

- 비트코인은 지난 24시간 동안 6% 상승했습니다.
- 현재 시장 추세는 강세 패턴을 보이고 있습니다.

이 조건과 추세를 고려할 때, 어울리는 거래 전략을 생성해주세요."



# 비트코인 자동매매 SW 개발내역 시연영상



# ChatGPT를 활용한 사투리 교육용 게임 서비스 개발

- 배우고 싶은 지역을 선택한 뒤, 그 지역에 대한 퀴즈를 맞춰가는 앱 서비스를 제작
  - 최근 소멸 중인 ‘제주 사투리’를 보며, 사투리를 보존하기 위해 기획함
  - 퀴즈를 맞춰가며, 포인트를 쌓게 되면 그 지역의 특산물을 포인트로 교환할 수 있음
  - 음성출력을 통해 음성으로 사투리를 들을 수 있게 함

제주도 방언을 표준어로 알려줘  
“ 혼저옵서 / 아방 / 어멍 / 도세기 / 감저 “

제주도 방언에서 표준어로 번역하면 다음과 같습니다:

1. 혼저옵서 - 어서오세요
2. 아방 - 아버지
3. 어멍 - 어머니
4. 도세기 - 돼지
5. 감저 - 고구마

위의 제주도 방언 표현들이 표준어로 무엇을 의미하는지 알려드렸습니다.



# ChatGPT 활용예시 (소스코드 개발)

- 주어진 데이터 및 작동 구조를 기반으로 html 및 JavaScript 코드 개발



You

학습할 수 있는 페이지인데 카드 형식으로 구현되어 있고 앞에는 제주 방언, 카드 클릭하면 표준어가 들 수 있도록 하고 싶어. 데이터들은 "흔저옵서 - 어서오세요 / 올데강, 흔저오십서 - 오셨습니까, 어서 오십시오. 감자 - 고구마 / 지실 - 감자 / 모물 - 메밀 / 어욱 - 억새 / 태역 - 잔디" 정도로 될 수 있게 어떻게 해야 할까?



ChatGPT

카드 형식의 페이지를 구현하려면 HTML, CSS, 그리고 JavaScript를 사용하여 카드의 앞면에 제주 방언을, 뒷면에 표준어를 표시하도록 만들 수 있습니다. 클릭 시 카드가 뒤집히는 효과를 추가하여, 사용자가 카드를 클릭하면 표준어가 나타나도록 합니다.

## HTML 구조

html

Copy code

```
<!DOCTYPE html>
<html>
<head>
    <title>사투리 학습</title>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
    <div class="card-container">
        <div class="card" onclick="this.classList.toggle('flipped');">
            <div class="card-front">흔저옵서</div>
            <div class="card-back">어서오세요</div>
        </div>
    </div>
</body>
```

## JavaScript 코드

javascript

Copy code

```
document.addEventListener('DOMContentLoaded', function() {
    var cards = document.querySelectorAll('.card');

    cards.forEach(function(card) {
        card.addEventListener('click', function() {
            card.classList.toggle('flipped');
        });
    });
});
```

이 코드는 페이지 로드 시 모든 '.card' 요소에 클릭 이벤트 리스너를 추가합니다. 카드가 클릭되면, 해당 카드의 클래스 목록에 'flipped' 클래스를 토글합니다. 이 토글 동작은 카드가 뒤집히는 효과를 만듭니다.

## HTML에 JavaScript 추가

html

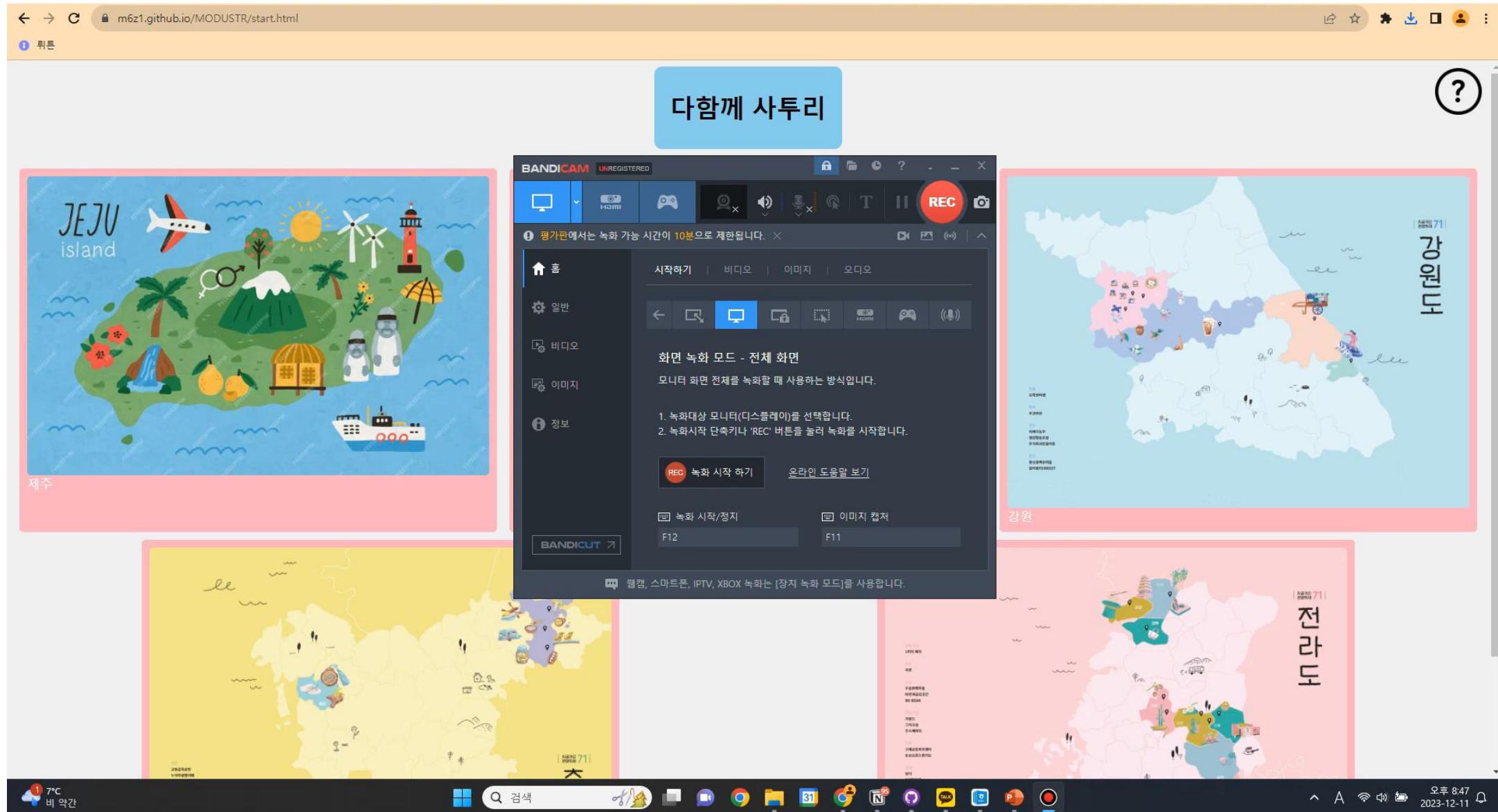
Copy code

```
<script>
    document.addEventListener('DOMContentLoaded', function() {
        var cards = document.querySelectorAll('.card');

        cards.forEach(function(card) {
            card.addEventListener('click', function() {
                card.classList.toggle('flipped');
            });
        });
    });
});
```



# ChatGPT를 활용한 사투리 교육용 게임 시연영상



# 일곱 용의 노래

*GPT와 Stable Diffusion으로 리소스를 생성해 개발한 텍스트 RPG*

## 개요

- ChatGPT와 Stable Diffusion으로 만든 Unity 게임
- ChatGPT는 게임 개발자의 역할을 맡을 수 있을까?

## 개발 목적

- 오늘날 사장된 장르인 텍스트 RPG 장르를 ChatGPT로 손쉽게 개발
- ChatGPT가 게임 개발자로서 얼마나 효용성이 있는지 확인

## 타겟 고객

- 어린이, 청소년, 성인 게임 이용자
- 핸드폰으로 가벼운 게임을 즐기는 게임 이용자들

# 일곱 용의 노래

GPT와 Stable Diffusion으로 리소스를 생성해 개발한 텍스트 RPG

## Chat GPT는 어디까지 게임 개발자의 역할을 할 수 있을까?

1인 개발을 하며, 최대한 많은 역할을 GPT가 수행하며 개발



게임 기획



아트 디자인



시나리오



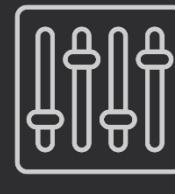
프로그래밍



서버



보안



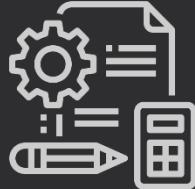
사운드



QA

# 일곱 용의 노래

GPT와 Stable Diffusion으로 리소스를 생성해 개발한 텍스트 RPG



## 게임 기획

**You**

던전을 탐색하며 등장할 수 있는 이벤트들을 기획하자.

다양한 몬스터가 출현할 수 있으며, 보물을 획득할 수도 있고, 가끔 나오는 쉼터에서 휴식도 할 수 있으면 좋겠다.

**ChatGPT**

던전 탐험 이벤트 기획:

- 다양한 몬스터 출현:**
  - 이벤트 설명: 던전을 탐험하면 다양한 몬스터가 등장합니다. 각 몬스터는 고유한 특성과 공격 스킬을 가지고 있으며, 플레이어는 전투를 통해 경험치와 아이템을 얻을 수 있습니다.
  - 플레이어 선택자:
    - 공격하여 전투
    - 회피 시도
    - 던전을 벗어남
- 보물 찾기:**
  - 이벤트 설명: 던전 내에서 특별한 보물이 숨겨져 있습니다. 플레이어는 보물을 찾기 위해 던전을 탐험하거나 퍼즐을 풀어야 합니다.
  - 플레이어 선택자:
    - 퍼즐 해결
    - 보물 위치 찾기 시도
    - 다른 방향으로 이동
- 쉼터 발견:**
  - 이벤트 설명: 가끔 던전 내에서 안전한 쉼터가 나타납니다. 플레이어는 쉼터에서 휴식을 취할 수 있으며, 체력과 마나가 회복됩니다.
  - 플레이어 선택자:
    - 휴식 취하기
    - 주변 탐험

던전 시스템을 기획하는 ChatGPT



던전 진행률: 2%

의간의 살기가 느껴진다. 멀리서 몬스터들이 날 지켜 보고 있는 느낌이 든다.

▶ 탐색한다.  
▶ 돌아간다.



핀팡이(?) 등장했다!

체력: 100  
공격력: 40  
스피드: 8

▶ 전투 시작!  
▶ 도망치다.



기획대로 구현된 게임의 던전 탐색 시스템

# 일곱 용의 노래

GPT와 Stable Diffusion으로 리소스를 생성해 개발한 텍스트 RPG



## 아트 디자인 with Stable Diffusion

**You**  
Stable Diffusion으로 일러스트를 뽑고 싶어. 프롬프트를 추천해줘.

로브를 두른 남자가 조그맣게 서있고, 등엔 칼 하나가 채워져 있어.  
바위 절벽 위에 서서 저 멀리 판타지 풍경을 바라보고 있어.

기본적으로 넣을 프롬프트  
(masterpiece, best quality), landscape, no humans

기본적으로 넣을 네거티브 프롬프트  
(worst quality, low quality:1.3), text, letter, letterboxed

추천할만한 프롬프트  
fantasy, 1boy, knight, hill

**ChatGPT**  
Stable Diffusion으로 원하시는 일러스트를 뽑으려면 다음 프롬프트를 사용해보세요:

"로브를 두른 남자가 조그맣게 서서, 등엔 칼 하나가 채워져 있습니다. 그는 바위 절벽 위에 서서 저 멀리의 환상적인 풍경을 바라보고 있습니다. 이 작품은 최상의 품질(masterpiece, best quality)을 갖춘 풍경(landscape)을 담은 일러스트레이션입니다. 작품 내에는 인간형태가 없습니다."

이 프롬프트는 원하는 요소와 품질을 강조하며, 불필요한 요소를 배제하도록 도와줄 것입니다. 또한, Fantasy와 1boy, knight, hill과 같은 추가 프롬프트를 포함하여 작품의 분위기나 주제를 더 강조할 수 있습니다.

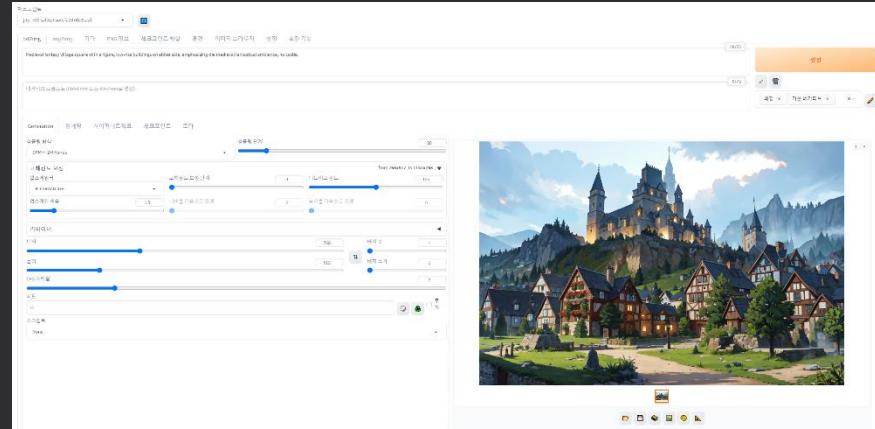
**You**  
영어로 바꿔어주고, 주어와 동사가 없는 짧은 문구로 만들어줘.

**ChatGPT**  
로브를 입은 남자, 칼, 바위 절벽, 환상적인 풍경.

**You**  
영어로 바꿔줘.

**ChatGPT**  
Man in robe, sword, rocky cliff, fantastical landscape.

일러스트 프롬프트를 생성하는 ChatGPT



해당 프롬프트를 그대로 사용해 생성한 이미지

# 일곱 용의 노래

GPT와 Stable Diffusion으로 리소스를 생성해 개발한 텍스트 RPG



## 시나리오

**You**

한 10~12줄 정도의 프로그 내용을 소설처럼 멋있는 문체로 적어줘.

검과 마법의 대륙 '이루티아'에서 일어나는 일이며,  
이 나라엔 신으로 섬겨지는 일곱 마리의 용이 존재해.  
그런데, 그 중 하나인 흑룡 바하무트가 갑작스럽게 날뛰기 시작해서  
이루티아 대륙의 북쪽이 완전 파멸되기까지 했어.

그 결과, 이루티아 대륙 남쪽의 마을 '엘버라'에 존재하는  
엘버라 기사단의 모든 기사들에게 남쪽에 가서 바하무트를 토벌하라는 임무가 주어져.  
때문에 게임의 주인공(기사단의 엘리트 기사, 플레이어) 역시  
바하무트를 토벌하기 위해 남쪽으로 출발하는 이야기야.

**ChatGPT**

이루티아, 검과 마법이 교차하는 대륙. 이곳에는 일곱 마리의 신비한 용이 모습을 드러내며 존재한다.  
그러나 평화로운 나라는 흑룡 바하무트의 갑작스러운 날뛰기로 인해 위협을 받게 되었다. 이루티아 대륙의 북쪽은 바하무트의 무자비한 파멸을 피하지 못하고 망가져갔다.

이로 인해 엘버라 마을의 기사들은 모두 남쪽으로 가서 이 악을 퇴치하라는 명령을 받게 되었다. 이 중에서도 특히 엘버라 기사단의 엘리트 중 한 명, 우리 이야기의 주인공은 바하무트를 물리치기 위해 떠난다. 그의 모험은 이루티아 대륙의 위기에 맞서 신비로운 여정을 시작하게 된다.

**You**

그럼 이제 무기 상점에 들어갔을 때 텍스트를 마련해줘. 전제적으로 대장간 느낌이 풀풀 나며, 석탄을 뒤집어쓴듯한 주인이 찾는 무기가 있느냐고 물어보는거야. 브로드 소드 / 레이피어 / 골드 소드 세 개를 구매할 수 있어.

**ChatGPT**

무기 상점에 벌을 딴던 당신은 또가운 화로에서 번쩍이는 불빛과 금속의 냄새가 가득한 대장간의 느낌이 확연하게 느껴진다. 석탄으로 뒤집어쓴 듯한 대장간이 당신에게 다가와 말한다.

"어서오세요, 모험가님! 어떤 무기를 찾으시나요? 우리 대장간에는 최고의 무기들이 기다리고 있답니다."

그의 눈에는 숙련된 무기 장인의 눈빛이 빛나고 있다.

"혹시 원하시는 무기가 있으나요? 저희 대장간에서는 브로드 소드, 레이피어, 그리고 골드 소드를 판매하고 있답니다."

- 브로드 소드를 구매한다 (가격: 30골드)
- 레이피어를 구매한다 (가격: 25골드)
- 골드 소드를 구매한다 (가격: 40골드)
- 돌아간다

시나리오와 소설을 라이팅하는 ChatGPT

**You**

게임 아이템에 두 줄짜리 설명을 적을거야.

순서대로 약한 검에서 강한 검으로 순서가 올라가.

**문체 예시)**

**녹슨 검**  
다 닳아 녹슬어버린 검.  
이가 다 빠져 못 쓸 것 같다.

**기사의 검**  
제국의 위대한 기사들이 배급받은 검.  
그 도신은 매우 아름답다.

**ChatGPT**

1. **녹슨 검**
  - 다 닳아 녹슬어버린 검. 이가 다 빠져 못 쓸 것 같다.
2. **브로드 소드**
  - 강력한 검술이 담긴 대형 검. 탁월한 파괴력을 자랑한다.
3. **레이피어**
  - 빠른 공격과 예리한 칼날을 지닌 검. 적의 방어를 뚫고 가는 데 특화되어 있다.
4. **골드 소드**
  - 황금으로 장식된 고급 검. 그 찬란한 모습만큼 강력한 성능을 자랑한다.
5. **아이스 세이버**
  - 얼음의 힘이 담긴 검. 적을 얼려버리는 차가운 일격을 날린다.
6. **프로즌 블레이드**
  - 얼음의 힘이 깃든 대검. 주변을 얼려버려 상대의 행동을 제압한다.
7. **칠흑검**
  - 검은 어둠의 힘이 깃든 검. 적을 둔화시켜 어둠 속에서 침투할 기회를 제공한다.

Few-Shot으로 아이템의 설명과 스토리를 작성

# 일곱 용의 노래

GPT와 Stable Diffusion으로 리소스를 생성해 개발한 텍스트 RPG



## 시나리오



검과 마법이 교차하는 대륙 '이루티아'.  
이곳엔 신으로 숭겨지는  
일곱 마리의 용이 존재한다.

그러나 여신년 120년,  
일곱 용 중 하나인 흑룡 바하무트가  
원인 불명으로 갑작스럽게 날뛰기 시작했고,  
이루티아 대륙의 북쪽은 바하무트에게  
무자비한 파멸을 맞이했다.

이로 인해 엘바라 마을의 '엘바라 기사단'의  
모든 기사에게, 대륙의 남쪽으로 떠나  
바하무트를 토벌하라는 임무가 주어진다.

그렇게 엘바라 기사단의 엘리트 기사,  
글리디우스 역시 바하무트를 토벌하기 위해  
모험을 시작하게 된다.

**게임 시작**

ChatGPT가 써준 라이팅대로 텍스트 작성



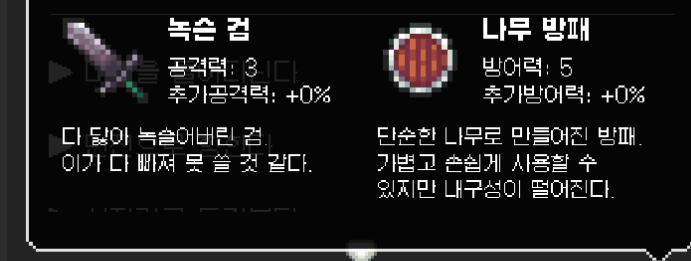
무기 상점에 발을 디딘 당신은 뜨거운 화로에서 번쩍  
이는 불빛과 금속의 냄새가 기득한 대장간의 느낌이  
확연하게 느껴진다. 석탄으로 뒤집어쓴 듯한 대장이  
당신에게 다가와 말한다.

"어서오세요, 기사님! 어떤 무기를 찾으시나요? 우리  
대장간에는 최고의 무기들이 기다리고 있답니다."

그의 눈에는 축령된 무기 장인의 눈빛이 빛나고 있다.

"혹시 원하시는 무기가 있나요? 저희 대장간에서는 브  
로드 소드, 레이피어, 그리고 골드 소드를 판매하고 있  
답니다."

- ▶ 브로드 소드를 본다.
- ▶ 레이피어를 본다.
- ▶ 골드 소드를 본다.
- ▶ 돌아간다.



ChatGPT가 생성한 아이템 설명

# 일곱 용의 노래

GPT와 Stable Diffusion으로 리소스를 생성해 개발한 텍스트 RPG



## 프로그래밍

**You**

좋아. 이제 다른 기능을 만들자.

주인공 캐릭터를 만들 때,  
힘, 민첩, 건강, 지혜, 매력 총 5종류의 스테이터스를 설정할거야.

각각의 스테이터스를 모두 자유롭게 내리거나, 올릴 수 있어.  
(스테이터스 하나에 낮추기 / 높이기 버튼이 전부 존재)

유저는 총 35의 스테이터스를 자유롭게 분배할 수 있으며,  
기본적으로 모든 스테이터스에 5씩 들어있어.

즉,  
힘:5  
민첩:5  
건강:5  
지혜:5  
매력:5  
남은 수치:10

이 맨 처음 캐릭터를 생성할 때 표시될 사항이야.

이제 코딩해야 할 C# 스크립트의 명세에 대해 설명할게.

게임의 인스펙터에 아래와 같은 오브젝트가 존재해.  
힘 수치 (TMpro)  
민첩 수치 (TMpro)  
건강 수치 (TMpro)  
지혜 수치 (TMpro)  
매력 수치 (TMpro)  
남은 수치 (TMpro)  
그리고, 스테이터스 5개에 낮추기 / 높이기 버튼 총 10개

우선 [SerializeField] 로 6개의 TMpro 텍스트 오브젝트를 받을 변수를 만들어줘.

유저는 각 스테이터스의 낮추기 / 높이기 버튼을 눌러 스테이터스를 조정할 수 있어.  
어떤 스테이터스의 수치를 1높였다면, 남은 수치가 1내려가고,  
어떤 스테이터스의 수치를 1낮췄다면, 남은 수치가 1올라가지.

어떤 스테이터스를 0 미만으로 낮출 수는 없어.  
그리고, 어떤 스테이터스든 20 이상으로 높일 수가 없어.

위 스테이터스 5개가 담긴 int 배열을 생성해주고, 기본값을 담고,  
스텟 낮추기 / 높이기 메서드를 만들어줘.

인자로 int 형태의 배열의 인덱스를 받자.

예를 들어 DownStatus(0) 을 호출했다면,  
힘 스테이터스가 1내려가고,  
남은 수치가 1올라가는거지.  
배열 인덱스는  
힘:0  
민첩:1  
건강:2  
지혜:3  
매력:4

로 하자.

DownStatus(string)  
UpStatus(string)

그리고 각 버튼을 눌렀을 때, 6개의 TMpro 텍스트 오브젝트에  
화면에 값이 갱신되는 메소드도 만들어주고,  
DownStatus나 UpStatus가 호출되었을 때 화면에 값을 갱신하는 메소드도 호출되게 해줘.

복잡한 기능을 구현하기 위해 한 질문에

세부적인 명세를 모두 서술하여 크게 질문

**ChatGPT**

이해했습니다. 아래는 설명한 내용을 기반으로 한 C# 스크립트의 예시입니다.

```
cssharp
using UnityEngine;
using TMPro;

public class CharacterCreation : MonoBehaviour
{
    // 인스펙터에서 설정할 TMpro 텍스트와 버튼 오브젝트
    [SerializeField] private TextMeshProUGUI strengthText;
    [SerializeField] private TextMeshProUGUI agilityText;
    [SerializeField] private TextMeshProUGUI healthText;
    [SerializeField] private TextMeshProUGUI wisdomText;
    [SerializeField] private TextMeshProUGUI charmText;
    [SerializeField] private TextMeshProUGUI remainingPointsText;

    [SerializeField] private GameObject[] decreaseButtons; // 낮추기 버튼들
    [SerializeField] private GameObject[] increaseButtons; // 높이기 버튼들

    // 초기 스테이터스 값
    private int[] stats = { 5, 5, 5, 5, 5 }; // 힘, 민첩, 건강, 지혜, 매력

    private int remainingPoints = 10; // 초기 남은 포인트

    private void Start()
    {
        UpdateUI();
    }

    // 특정 스테이터스를 감소시키는 메서드
    private void DownStatus(int index)
    {
        if (stats[index] > 0 && remainingPoints < 20)
        {
            stats[index]--;
            remainingPoints++;
            UpdateUI();
        }
    }

    // 특정 스테이터스를 증가시키는 메서드
    private void UpStatus(int index)
    {
    }
}
```

바로 이해하고 스크립트 작성

# 일곱 용의 노래

GPT와 Stable Diffusion으로 리소스를 생성해 개발한 텍스트 RPG



## 프로그래밍

이건 아이템 매니저 코드로, 각 아이템 이름 / 고정 상승수치 / 퍼센트 상승수치 / 설명을 내가 전부 적어놨어.

이제 `itemInfo`라는 이름의 스크립트를 하나 새로 작성할거야.  
`ItemManager itemManager => ItemManager.Instance;`로 위 아이템 매니저 싱글톤을 가져와줘.  
`InfoManager infoManager => InfoManager.Instance;`로 무기 / 방어구 이미지 배열도 가져오자.

그리고, `UpdateItemInfo()`라는 public 메서드가 호출되면,  
아래의 리소스들을 내가 장착한 무기 / 방어구에 맞게 적어주면 돼.

### 1. 무기 외형 / 방어구 외형

`[SerializeField]`로 `WeaponImage` / `ArmorImage` 변수를 만들어놓으면, 내가 참조할게.

`InfoManager`라는 클래스에,  
`[SerializeField] public Sprite[] weaponImages;` // 무기 이미지 배열  
`[SerializeField] public Sprite[] armorImages;` // 방어구 이미지 배열  
로 무기 이미지와 방어구 이미지를 전부 배열로 할당해뒀어.

`SaveManager.Weapon`, `SaveManager.Armor`로 장착한 코드를 배열 인덱스로서 반영해줘.

### 2. 무기 이름 / 방어구 이름

`[SerializeField]`로 TMPro의 텍스트 오브젝트를 무기와 방어구 각각 만들어뒀으니 내가 참조할게.  
`ItemManager`의 `WeaponData` / `ArmorData`를 참고해서 반영해줘.

### 3. 무기 스텟 / 방어구 스텟

`[SerializeField]`로 TMPro의 텍스트 오브젝트를 무기와 방어구 각각 만들어뒀으니 내가 참조할게.

무기는 아래의 형식대로 적어줘.

공격력 + fixedIncrease

추가공격력 + percentIncrease

단, `percentIncrease`가 1이이라고 되어있는 건 10%라고 적히게 해줘.

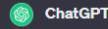
아마 100을 곱한 뒤, 100을 빼면 되겠지?

방어구도 방어력, 추가방어력으로 똑같이 해줘.

### 4. 무기 설명 / 방어구 설명

`[SerializeField]`로 TMPro의 텍스트 오브젝트를 무기와 방어구 각각 만들어뒀으니 내가 참조할게.  
`ItemManager`의 `WeaponData` / `ArmorData`를 참고해서 반영해줘.

한 번 ChatGPT가 생성한 스크립트는,  
해당 Chat에선 GPT가 반 영구적으로 기억  
아키텍처를 계속해서 키워나갈 수 있음



ChatGPT

이해했습니다. 아래는 `ItemInfo` 스크립트의 예시입니다.

```
csharp
using UnityEngine;
using TMPro;

public class ItemInfo : MonoBehaviour
{
    private ItemManager itemManager => ItemManager.Instance;
    private InfoManager infoManager => InfoManager.Instance;

    [SerializeField] private SpriteRenderer weaponImage;
    [SerializeField] private SpriteRenderer armorImage;
    [SerializeField] private TMP_Text itemNameText;
    [SerializeField] private TMP_Text itemStatsText;
    [SerializeField] private TMP_Text itemDescriptionText;

    public void UpdateItemInfo()
    {
        // 1. 무기 외형 / 방어구 외형
        int weaponIndex = SaveManager.Weapon;
        int armorIndex = SaveManager.Armor;

        if (weaponIndex >= 0 && weaponIndex < infoManager.weaponImages.Length)
        {
            weaponImage.sprite = infoManager.weaponImages[weaponIndex];
        }

        if (armorIndex >= 0 && armorIndex < infoManager.armorImages.Length)
        {
            armorImage.sprite = infoManager.armorImages[armorIndex];
        }

        // 2. 무기 이름 / 방어구 이름
        string weaponName = itemManager.weaponDatas[weaponIndex].name;
        string armorName = itemManager.armorDatas[armorIndex].name;

        itemNameText.text = $"무기: {weaponName}\n방어구: {armorName}";

        // 3. 무기 스텟 / 방어구 스텟
        int weaponFixedIncrease = itemManager.weaponDatas[weaponIndex].fixedIncrease;
        float weaponPercentIncrease = itemManager.weaponDatas[weaponIndex].percentInc
        int armorFixedIncrease = itemManager.armorDatas[armorIndex].fixedIncrease;
```

명령 프롬프트에 넣지  
않았음에도 불구하고,  
이전에 자신이 스크립팅한  
코드를 참조하여 프로그래밍

# 일곱 용의 노래

GPT와 Stable Diffusion으로 리소스를 생성해 개발한 텍스트 RPG



서버



보안

1. `GameDataLoad()` 메소드 설정:

```
csharp
public void GameDataLoad()
{
    // ... 기존 로직 ...

    for (int i = 0; i < 3; i++)
    {
        userGameData.knights[i] = int.Parse(gameDataJson[0][ $"knight{i + 1}" ].ToString());
        userGameData.magics[i] = int.Parse(gameDataJson[0][ $"magic{i + 1}" ].ToString());
        userGameData.heals[i] = int.Parse(gameDataJson[0][ $"heal{i + 1}" ].ToString());
    }

    // ... 기존 로직 ...
}
```

게임 백엔드 서버와 JSON 형태의 데이터를  
주고받는 메서드를 구현하는 ChatGPT

You

```
using TMPro;
using UnityEngine;

public class PotionShopInfo : MonoBehaviour
{
    InfoManager InfoManager => InfoManager.Instance;

    [SerializeField] private TextMeshProUGUI smallCostText;
    [SerializeField] private TextMeshProUGUI bigCostText;

    private int smallCost = 100;
    private int bigCost = 500;

    private void OnEnable()
    {
        // Charm에 따른 비용 조정
        int charmBonus = SaveManager.Charm; // 상수값으로 조정

        // 실제 비용에 Charm 반영
        smallCost -= charmBonus;
        bigCost -= charmBonus;

        // UI에 출력
        smallCostText.text = $"Cost: {smallCost}";
        bigCostText.text = $"Cost: {bigCost}";
    }
}

이 PotionShopInfo 메서드,
필드와 메서드, 클래스 등의 이름들을 전부 랜덤한 문자열로 생성해서 난독화해줘.
```

ChatGPT

C# 코드의 난독화는 일반적으로 성능 저하나 코드 가독성 감소와 관련이 있습니다. 또한 난독화된 코드는 디버깅이 어려워질 수 있습니다. 이러한 이유로 실제 프로덕션 코드에서 난독화를 사용하는 것은 권장되지 않습니다. 난독화는 주로 보안 목적이나 프로텍션의 일환으로 사용되기도 하지만, C#의 경우 일반적으로 난독화가 강력한 이점을 제공하지 않습니다.

그러나 난독화된 코드를 생성해보겠습니다. 이 코드를 사용하기 전에 주의해야 합니다. 또한 난독화된 코드는 코드 품질과 가독성을 저하시킬 수 있으므로 주의해서 사용해야 합니다.

원하는 스크립트의 필드, 메서드, 프로퍼티, 파라미터, 클래스명 등  
코드를 랜덤한 문자열로 난독화하여 생성해주는 ChatGPT

```
public void L1(int Mm)
{
    int Nn = 0;
    int Oo = 0;

    if (Mm == 0)
    {
        Nn = Gg;
        Oo = Ss;
    }
    else if (Mm == 1)
    {
        Nn = Hh;
        Oo = Tt;
    }

    if (Pp(Mm, Nn, Oo))
    {
        Qq(Mm, Nn, Oo);
    }
    else
    {
        UM.Debug.Log("Re Ss Tt.");
    }
}

private bool Pp(int Mm, int Nn, int Oo)
{
    int Uu = Cc.Vv;
    int Ww = (Mm == 0) ? Cc.Xx : Cc.Yy;

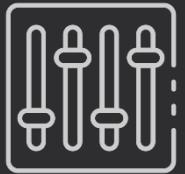
    return Uu >= Nn && Ww < Oo;
}

private void Qq(int Mm, int Nn, int Oo)
{
    Cc.Vv -= Nn;

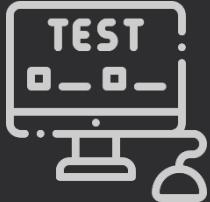
    if (Mm == 0)
    {
        Cc.Xx++;
    }
    else if (Mm == 1)
    {
        Cc.Yy++;
    }
}
```

# 일곱 용의 노래

GPT와 Stable Diffusion으로 리소스를 생성해 개발한 텍스트 RPG



사운드



QA

**You**

배경음악을 작곡하고 싶은데, 멜로디를 추천해줘.  
너무 빠르지 않게, bpm은 110정도로,  
리듬조로 작곡하며, 멜로디 제이름과 코드를 같이 작성해줘.

**ChatGPT**

멜로디 추천:

```
mathematica
E | G A B A G | F E D C D E | G A B A G | F E D C D E |
```

Copy code

이 멜로디는 리듬조에서 파생된 조성이며, 감각적이고 모험적인 느낌을 주기에 적합합니다. 110 BPM  
으로 재생할 경우 일정한 리듬과 조화를 유지할 수 있을 것입니다.

코드 진행:

- 첫 부분: Am
- 두 번째 부분: Dm G
- 세 번째 부분: Am
- 네 번째 부분: Dm G Am

이렇게 되면, 전체적으로 Am - Dm - G - Am으로 진행되는 반복 패턴이 형성됩니다. 이는 일반적으로  
판타지적이고 모험적인 분위기를 나타낼 수 있는 코드 진행입니다.

머니코드로 작곡을 해주는 ChatGPT

SU

You

'고요숲' 던전에서 한 번 탐색을 할 때 드는 시간은 10초야.  
탐색을 한 번 할 때마다, 40% 확률로 아무일도 일어나지 않고, 20% 확률로 쉴 수 있어.

중요한 것은, 30% 확률로 몬스터를 조우하며, 10% 확률로 보물을 찾는다는거야.

몬스터와 전투하여 승리하는 데엔 20초정도 소요되며,  
대개 40골드 ~ 80골드 사이의 골드를 벌어.

보물을 찾으면 거의 즉시 200골드 가까이 벌 수 있어.

한 20분동안 고요숲에서 탐색을 계속해서 하면,  
플레이어가 어느 정도의 골드를 벌 수 있을까?

**플레이어의 평균 골드 수급량을  
알기 위해서, 던전 탐색의  
명세를 넣은 후 원하는 시간 동안  
골드 수급량을 계산  
결과에 따라 밸런스 조절**

SU

ChatGPT

탐색 시간은 10초로 고정되어 있으며, 각 이벤트의 확률을 고려하여 플레이어가 한 번의 탐색에서 얼마나 골드를 벌 수 있는지 계산할 수 있습니다.

- 아무일도 일어나지 않는 경우: 40%
  - 아무 일도 일어나지 않으므로 골드는 증가하지 않습니다.
- 쉴 수 있는 경우: 20%
  - 쉴 때에도 아무 일도 일어나지 않으므로 골드는 증가하지 않습니다.
- 몬스터와 조우하는 경우: 30%
  - 몬스터와의 전투에 소요되는 시간은 20초이며, 골드 범위는 40골드에서 80골드입니다.
  - 따라서 한 번의 전투로 얻을 수 있는 골드는 평균  $(40 + 80) / 2 = 60$ 골드입니다.
- 보물을 찾는 경우: 10%
  - 보물을 찾을 때 걸리는 시간은 거의 즉시이며, 골드 범위는 200골드 가까입니다.

따라서 한 번의 탐색에서 플레이어가 얻을 수 있는 평균 골드는 다음과 같습니다:

$$(0.4 \times 0) + (0.2 \times 0) + (0.3 \times 60) + (0.1 \times 200) = 33 \text{ 골드}$$

20분(1200초) 동안 10초마다 한 번씩 탐색한다면, 플레이어가 얻을 수 있는 총 골드는:

$$33 \text{ 골드}/\text{탐색} \times (1200 \text{ 초}/10 \text{ 초}/\text{탐색}) = 3960 \text{ 골드}$$

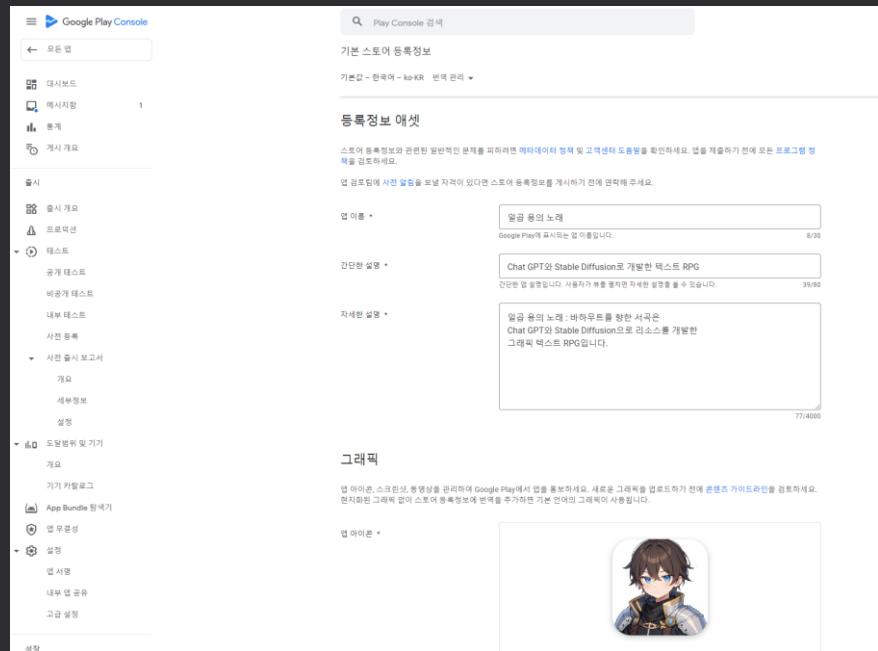
따라서 플레이어는 20분 동안 고요숲에서 탐색하면 약 3960골드를 얻을 수 있습니다.

# 일곱 용의 노래

GPT와 Stable Diffusion으로 리소스를 생성해 개발한 텍스트 RPG

## 개발 결과물

<https://github.com/Suldangoo/A-Songs-of-the-Seven-Dragons.git>



플레이스토어 출시 및 런칭 준비

## 시연 영상

이세진

# 오늘도 나는 먹는다

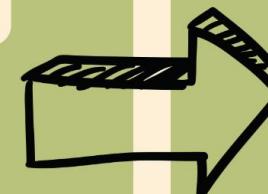
이세진과 다이어트, 이세진



# 프로젝트 개요

감정기록과 특정 영양소 섭취

평소 감정과 특정 영양소 섭취



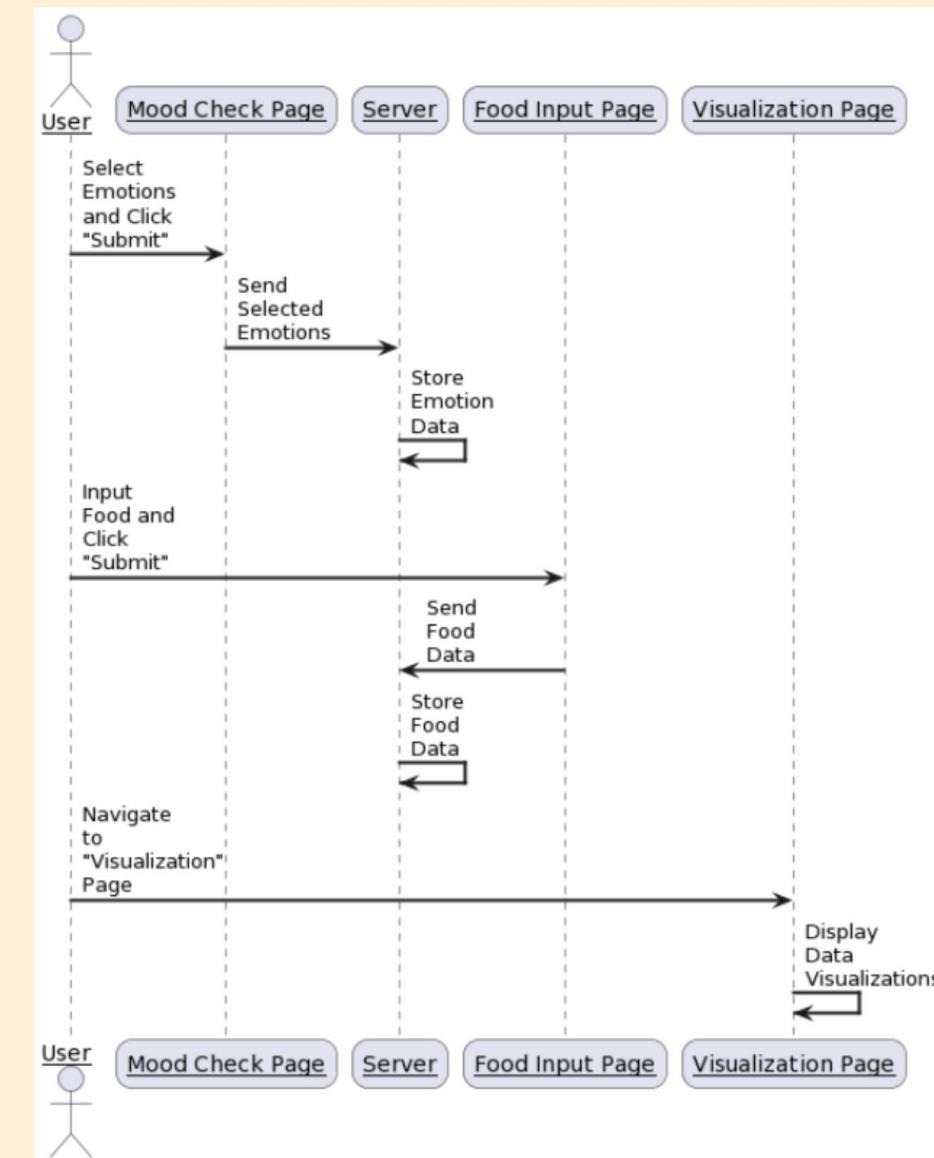
건강한 식단을 위한  
길을 제시

# 제안 기술의 특장점

항목	기존 기술의 문제	제안 서비스의 특장점
맞춤형 건강 제안	사전에 정의된 일반적인 건강 조언을 제공	사용자의 감정과 음식 데이터를 기반으로 맞춤형 건강 제안을 제공
데이터 시각화	주로 텍스트 형식의 조언을 제공	데이터를 시각화하여 사용자가 쉽게 데이터 추세를 파악
음식 및 감정 연계	서비스는 제한적	식습관과 감정의 관계를 시각화하고 분석



# 개발 기술 소개



# 개발 기술 소개



**exeljs #1**

감정, 섭취 음식  
데이터 수집

**pandas #2**

'식품 영양정보 데이터셋'  
데이터 가공

**recharts #3**

입력된 감정, 음식 데이터  
시각화

**recharts #4**

감정기록과 섭취 영양소의  
상관 관계

# GPT 활용방안

데이터 저장: Excel 파일에 저장:

- Node.js 서버 안에 엑셀 파일로 저장할 것.
- 엑셀파일이 없으면 생성하고, 있으면 해당 파일에 데이터를 저장할 것.
- 파일에 데이터를 입력받을 때에는 입력받은 시각도 저장할 것.
- Node 서버 만드는 방법부터 실행까지 과정을 자세히 설명할 것.

1. 프로젝트 초기화
2. 서버 코드 작성
3. 서버 실행 방법
4. React 데이터 조회

# GPT 활용방안

데이터 저장해줘~

	A	B	C	D	E	F
1	Mood	Food	Timestamp			
2		negative		2023. 12. 13. 오후 12:45:33		



# GPT 활용방안

A	B	C	D	E	F
Mood	Food				
negative		12. 13. 오후 12:45:33			

데이터를 저장해줘~



## data.xlsx 파일에 데이터 저장해줘

- A1에 Mood, B1에 Food, C1에 TimeStamp 컬럼이 있다.
- Mood데이터가 저장되면 Food값은 null을 넣는다.
- Food 데이터가 저장되면 Mood 값은 null을 넣는다.



# 개발 결과물

[Oh-Na-Muk / README.md](https://github.com/MarieSJLEE/Oh-Na-Muk)

MarieSJLEE now

14 lines (9 loc) · 334 Bytes

Preview Code Blame ⋮ ⋮

## Oh-Na-Muk

2023 소프트웨어공학 GPT 활용 프로젝트 '오늘도 나는 먹는다(Oh-Na-Muk)' 저장소입니다.

### node version

v16.20.2

### settings

```
$ npm install
$ npm start
```

url <https://github.com/MarieSJLEE/Oh-Na-Muk>

## Oh-Na-Muk-Back

2023 소프트웨어공학 GPT 활용 프로젝트 '오늘도 나는 먹는다(Oh-Na-Muk)' 서버 저장소입니다.

A	B	C	D	E
Mood	Food	Timestamp		
positive		2023. 12. 13. 오후 7:04:01		
positive		2023. 12. 13. 오후 7:04:04		
positive		2023. 12. 13. 오후 7:04:05		
	샐러드	2023. 12. 13. 오후 7:04:19		
negative		2023. 12. 13. 오후 7:28:59		

MoodCheck, FoodInput을 통해 엑셀에 사용자 데이터 수집

A	B	C	D	E
DB군	식품명	식품대분류	에너지(kcal)	대표영양소
수산물	가다랑어	수산물	127	단백질(g)
수산물	가다랑어	수산물	0	단백질(g)

농축산물, 수산물, 음식 데이터셋을 기준하여 식품.xlsx 생성 > 음식과 대표 영양성분을 매치

### node version

v16.20.2

### settings

```
$ npm install
$ npm start
```

url: <https://github.com/MarieSJLEE/Oh-Na-Muk-Back>



# 구현사항 시연

오늘도 나는 먹는다

Mood Check

Food Input

Visualization

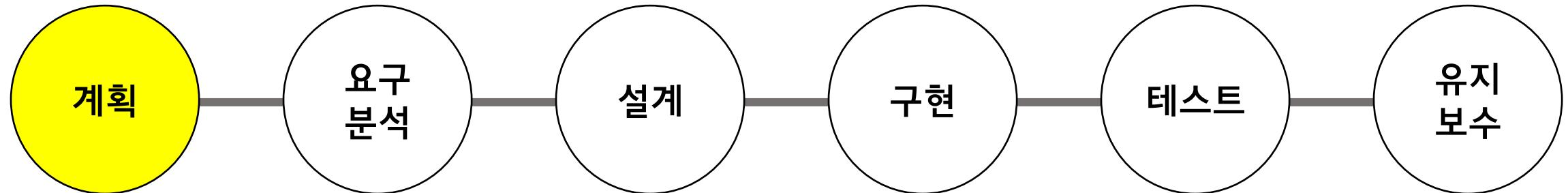
## Mood Check

How are you feeling?



Submit

# 소프트웨어 개발과정



## • 계획 (Planning)

- 목적: 소프트웨어 개발의 기본적인 방향과 목표를 설정하고, 프로젝트의 범위를 명확히 정의하는 단계
- 주요 활동: 프로젝트 목표 설정, 필요한 자원(인력, 예산 등)의 결정, 타임라인 및 마일스톤 계획, 위험 분석 및 해결 계획 수립, 개발 범위 및 주요 기능 정의
- 결과물: 프로젝트 계획서, 범위 정의서, 예산 계획서, 일정표

# 일정 계획

- 프로젝트 관리도구를 활용한 일정 계획
  - 간트 도표(Gantt chart) : 다양한 설계 활동을 시간축상에 표시한 수평 막대 그래프

ID	작업이름	시작	완료	기간	2023. 5				2023. 6		
					1~8	9~16	16~23	23~31	7~14	14~21	22~29
1	프로젝트 시작	2023-05-01	2023-05-01	1d							
2	고객 요구사항 이해	2023-05-01	2023-05-10	10d							
3	문제 기술 명료화	2023-05-05	2023-05-13	9d							
4	연구 수행	2023-05-12	2023-05-17	6d							
5	기능 요구사항 분석	2023-05-15	2023-06-02	19d							
6	대안 생성	2023-05-28	2023-06-06	10d							
7	시험결과 보고	2023-06-17	2023-06-24	8d							
8	대안 중 선정	2023-06-20	2023-06-25	6d							
9	설계프로세스 문서화	2023-06-23	2023-07-03	11d							
10	프로젝트 종료	2023-07-01	2023-07-04	3d							

[그림 2-1] 간트 도표

# 일정 계획

- 작업분해구조(Work Breakdown Structure : WBS) : 설계 프로젝트를 완수하기 위해 수행해야 할 모든 과업을 계층적으로 나타낸 관리도구



[그림 2-2] 작업분해구조(WBS)

# 일정 계획

- 팀 일정표(team calendar) : 각 작업이 완료돼야 할 최종 기한과 시간 프레임을 강조하며, 설계 팀이 사용할 수 있는 모든 시간을 정리하는 관리도구

2023년 5월						
월요일	화요일	수요일	목요일	금요일	토요일	일요일
1	2	3	4	5	6	7
				오후 5:00 원형 제작		
8	9	10	11	12	13	14
	오후 7:00-8:15 팀 회의			오전 11:00 개념 검증 기한		
15	16	17	18	19	20	21
오전 11:00 대략 개요 기한	오후 7:00-8:15 팀 회의			오후 5:00 주제기술문 개요 기한		
22	23	24	25	26	27	28
오전 11:00 발표 개요 기한	오후 7:00-8:15 팀 회의	오전 11:00 슬라이드 기한		오후 5:00 최종보고서 초고 기한		
29	30	31	6/1	2	3	4
	오후 7:00-8:15 팀 회의		오후 5:00 최종보고서 제출 기한			

[그림 2-3] 팀 일정표

# 일정 계획

- **선형책임도표(Linear Responsibility Chart : LRC ) : WBS의 각 과업을 성공적으로 완수 할 책임자와 과업을 완료하기 위해 참여해야 할 사람들을 식별하는 관리도**

선형책임도표	가온	나훈	다미	라나	튜터	외부 전문가	기호
1.0 주제 브레인스토밍	1						1=일차 책임
1.1 문제 기술 명료화	1	2	2	2			2=보조/작업
1.2 연구 수행	1	2		2		4	3=자문 필수
2.0 기능 요구사항 분석	2	2	1	2	5	3	4=자문 가능
3.0 대안 생성				1			5=검토
4.0 대안 평가	5	1	2	2			6=최종승인
5.0 선호하는 설계 선정	1	2			5	4	
6.0 설계결과 문서화		1					
7.0 프로젝트 관리	1						
7.1 주간회의	1	2	2	2			
7.2 프로젝트 계획 개발	1	2	2	2			
7.3 진척도 추적	1					5	
7.4 진도보고서	1						

[그림 2-4] 선형책임도표(LRC)

# 비용 계획

- **인건비 산정**

- 맨먼스(Man/Month): 한 명의 개발자가 한 달 동안 일한 작업량을 의미
- 맨먼스를 기준으로 프로젝트에서 몇 명의 개발자가 몇 개월 동안 프로젝트에 참여해야 하는지를 계산하여 총 개발 비용을 산정
- 맨먼스 = 참여율 × 참여기간

1명, 100%, 12개월 참여 시 → 12MM  
2명, 20%, 3개월 참여 시 → 1.2MM  
5명, 40%, 2개월 참여 시 → 4MM

- 인건비 = 참여율 × 참여기간 × 노임단가

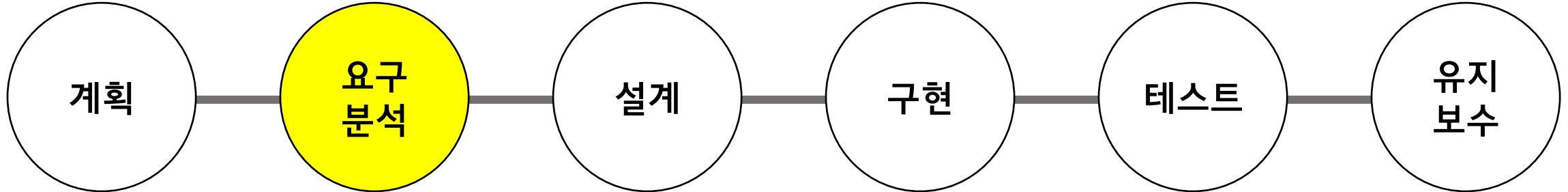
1명, 100%, 12개월, 4백만 → 4,800만  
2명, 20%, 3개월, 8백만 → 960만  
5명, 40%, 2개월, 6백만 → 2,400만

# 비용 계획

---

- 인건비 (개발비)
- 개발용역비 (외부 기관과 공동 수행하는 경우)
- 하드웨어 구입비 (PC, 네트워크 장비 등)
- 네트워크 구축 설치비
- 운영체제, DB 등 소프트웨어 구입비
- 유지보수비 (장비 재구매 비용, AS비용 등)
- 기타 개발에 필요한 비용

# 소프트웨어 개발과정



## • 요구 분석 (Requirements Analysis)

- 목적: 시스템이 어떤 기능을 가져야 하는지, 사용자 및 비즈니스 요구 사항을 명확히 정의하는 단계
- 주요 활동: 고객과의 인터뷰 및 회의, 시스템의 기능적 요구사항(무엇을 해야 하는가) 및 비기능적 요구사항(성능, 보안 등) 수집, 요구사항 명세서(SRS, Software Requirements Specification) 작성
- 결과물: MRD, PRD 등 요구사항 명세서

# 요구사항(Requirements) 문서

---

- 제품의 전략과 요구사항을 명확히 정의하기 위한 MRD와 PRD
- MRD (Market Requirements Document) - 시장 요구사항 문서
  - MRD 작성의 목적은 특정 시장에서 고객의 요구와 경쟁 상황, 시장의 기회 등을 식별하여 제품 개발의 방향과 전략을 정의하는 것
  - 주요항목: 타겟 시장의 설명, 고객의 문제점 및 요구사항, 시장의 크기와 성장 예측, 경쟁 분석, 제품의 위치 및 가치 제안 등
- PRD (Product Requirements Document) - 제품 요구사항 문서
  - PRD는 MRD를 기반으로 제품의 구체적인 기능과 기술적 요구사항을 정의하고, 개발팀이 제품을 구현할 때 참조하는 핵심 문서
  - 주요항목: 제품의 기능적 요구사항, 사용자의 경험과 인터페이스에 관한 요구사항, 성능과 다른 시스템과의 통합에 대한 요구사항, 제품의 제약사항 등

# MRD (Market Requirements Document)

---

- 도서 대출 시스템 개요
  - 개발 목적: 대학 도서관에서 학생 및 교직원이 효율적으로 도서를 대출하고 반납할 수 있도록 시스템을 구축하여 사용 편의성을 향상시키는 것
  - 주요 사용자(타겟 고객): 학생, 교직원, 도서관 사서
- 시장 분석 및 기회
  - 기존의 도서 대여 시스템은 수작업으로 운영되는 경우가 많고, 이는 비효율성을 초래함
  - 사용자가 도서 상태를 실시간으로 확인하기 어렵고, 대여 및 반납에 많은 시간이 소요됨
  - 대여 기록과 반납 기한 관리가 불편해 연체 및 벌금 부과가 제대로 이루어지지 않음
  - 스마트 도서관 시스템을 구축하여 자동화된 대여 및 반납, 벌금 관리 등 사용자 편의성을 높일 수 있음
- 경쟁 제품 및 차별화 요소
  - A 도서관의 관리 시스템 - 사용자 친화적 UI/UX 보유
  - B 도서관 시스템 - 반납 기한 초과 시 자동 벌금 부과 시스템

# PRD (Product Requirements Document)

---

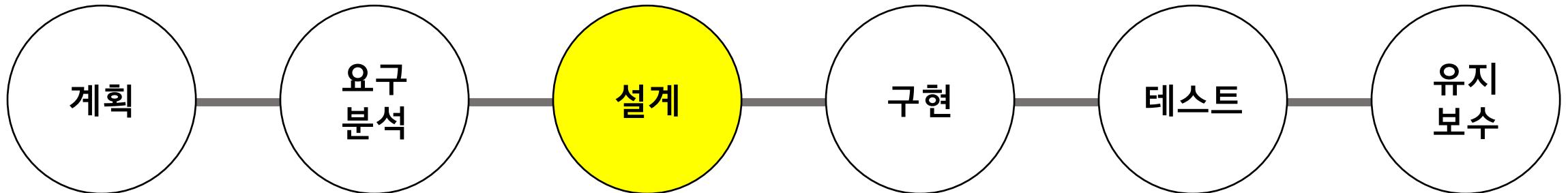
- 도서 대출 시스템 개요
  - 개발 목적: 사용자(학생 및 교직원)가 도서관에서 효율적으로 도서를 대출 및 반납할 수 있도록 자동화된 시스템을 개발하여 제공
  - 주요 기능: 도서 검색, 대출 및 반납 관리, 대여 기록, 벌금 부과 관리, 모바일 지원
- 기능 요구 사항
  - 도서 검색: 사용자가 도서관에 소장된 도서를 검색할 수 있음, 필터 기능 (저자, 제목, 출판사 등) 제공, 실시간으로 도서의 대출 가능 여부를 확인 가능
  - 도서 대여: 사용자가 도서를 대여할 수 있으며, 대여 기간이 자동으로 설정됨 (기본 14일), 사용자가 최대 3권까지 동시에 대여 가능, 대출 완료 시 사용자는 모바일 알림을 받음
  - ...
- 비기능 요구 사항
  - 성능: 시스템은 500명 이상의 동시 사용자가 사용할 수 있어야 하며, 대출 및 반납 과정은 2초 이내 처리
  - 보안: 사용자의 개인정보 및 대여 기록을 안전하게 보호하는 암호화 기능을 제공해야 함
  - 확장성, 유지보수성 ...

# PRD (Product Requirements Document)

---

- 사용자 인터페이스
  - 직관적인 UI/UX를 통해 사용자가 손쉽게 도서를 검색하고 대여할 수 있어야 함
  - 도서 검색 시 필터링을 통해 쉽게 결과를 좁힐 수 있는 기능을 제공
  - 대여 및 반납 상태, 벌금 정보를 쉽게 확인할 수 있는 모바일 알림 제공
- 제약조건
  - 법적 제약: 사용자의 개인정보 보호와 관련된 법률 준수 (예: GDPR 등)
  - 기술적 제약: 시스템의 성능을 보장하기 위해 500명 이상 동시 접속이 가능하도록 인프라를 구축해야 함
  - 비용: 시스템 구축 및 유지보수를 위해 할당된 예산 내에서 개발을 완료해야 함
- 품질기준
  - 안정성: 시스템 다운타임이 연간 0.1% 이내로 유지되어야 함 (99.9%의 가용성)
  - 성능: 대출 및 반납 처리 속도는 평균 2초 이내여야 함
  - 보안성: 사용자 정보가 암호화되고 데이터 누출이 발생하지 않아야 함
  - 사용성: 사용자 만족도 설문에서 85% 이상의 긍정적 평가를 목표로 함

# 소프트웨어 개발과정



## • 설계 (Design)

- 목적: 요구 분석 단계에서 정의된 사항을 기반으로, 시스템의 구조와 기능을 설계하는 단계
- 주요 활동: 시스템 아키텍처 설계, 데이터베이스 설계, 모듈 설계, UI/UX 설계
- 결과물: 시스템 설계서, 아키텍처 다이어그램, 데이터베이스 설계서, UI/UX 프로토타입

# 시스템 모델링(System Modeling)

---

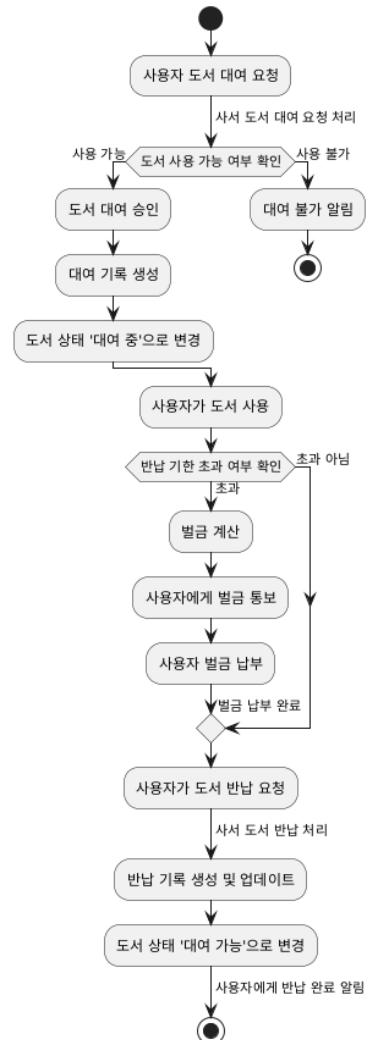
- 시스템 모델링은 시스템 요구사항을 도표기반의 모델링 기법을 활용해 모델을 만드는 것
  - UML(Unified Modeling Language): 시스템을 모델링 하기 위한 13가지 표기법들의 집합 (High/Low level 모델링)
- High-Level 모델링 - 시스템의 전체적 구조와 기능 표현
  - Activity Diagram: 비즈니스 프로세스, 워크플로우, 시스템 동작 흐름 등 표현
  - Use case Diagram: 시스템의 주요기능과 사용자 관점에서 상호작용 표현
- Low-Level 모델링 - 시스템의 구체적인 구현 세부사항 표현
  - Sequence Diagram: 시스템 내 객체 간 상호작용과 메시지 전달순서 표현
  - Class Diagram: 시스템 데이터의 구조와 구조간 관계 표현
  - State Diagram: 객체의 생명주기 동안 상태 변화에 대해 표현

# Activity Diagram

- 순서도와 형태가 비슷하며, 개발자와 사용자가 쉽게 이해하고 작성할 수 있는 디어그램
- 시스템이나 프로젝트, 업무 흐름, 활동 순서, 계산 과정을 단계적으로 작성할 수 있으며, 일의 흐름이나 화면의 흐름을 표현할 때 사용함

[표 7-2] 액티비티 디어그램의 사물을 표현하는 요소들 (계속)

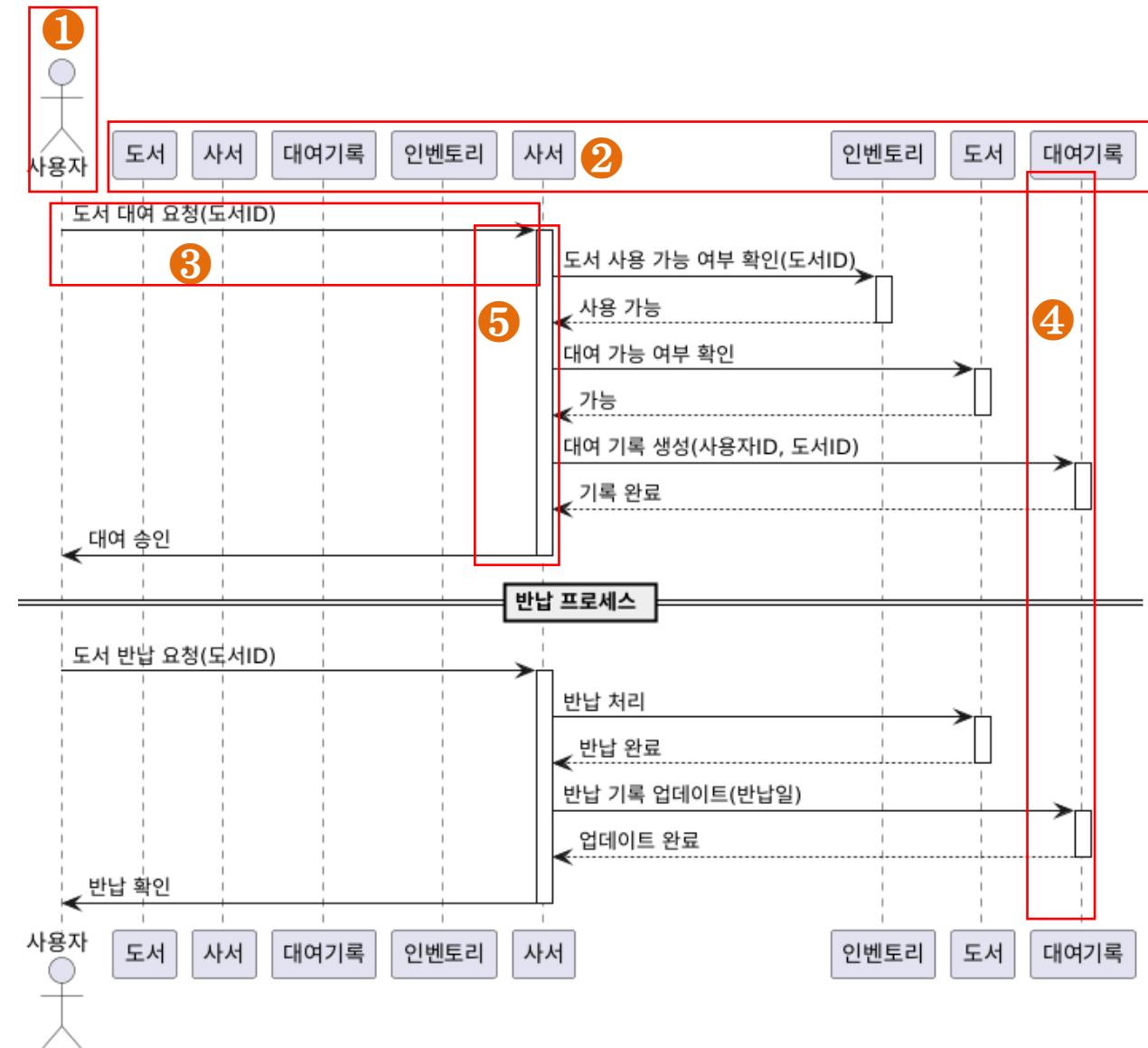
이름	그림	역할
시작점 (Initial Node)	●	• 액티비티를 호출하여 흐름의 시작을 알리는 노드
종료점 (Final Node)	○	• 동작이나 과정의 끝을 알리는 노드
액션 (Action)	Action	• 액티비티 내에서의 한 단계 • 더 작은 단위로 분해 불가
액티비티 (Activity)	Activity	• 활동 모델 내의 구조 표현 • 액션과 하는 일들로 구성



# Sequence Diagram

- 시스템 세부사항들의 진행 과정과 시간적 흐름을 표현할 수 있는 다이어그램
- 팀에서 탐색한 주제에 맞게 프로젝트가 잘 진행되고 있는지 활동의 흐름을 확인할 수 있으며, 활동을 시간 순서에 따라 자세히 표현 가능

- 액터(actor) : 시스템을 사용하는 사람이나 외부 시스템을 의미
- 객체(object) : 시스템 내에서 객체와 객체, 객체와 액터가 상호작용하는 개체를 의미
- 메시지(message) : 객체 간의 상호작용을 의미한다. 수평의 화살표 방향으로 진행되는 것을 의미
- 생명선(lifeline) : 객체가 존재할 수 있는 시간을 의미한다. 객체가 소멸되면 생명선도 함께 소멸됨
- 활성화(activation) : 객체가 활성화된 기간을 의미



# UML 그리기

- Draw.io에서 직접 그리기
  - 손으로 도형을 직접 작성하여 그리기
  - <https://draw.io/>
- ChatGPT를 활용하여 UML 그리기
  - 만들고자 하는 기능에 대해 설명하고, 해당 기능을 포함하는 PlantUML 코드 작성하기
  - 작성된 코드를 Visualize 도구 활용하여 시각화
  - <https://www.plantuml.com/plantuml/uml/>



도서대여 시스템 UML을 그리려고 해. PlantUML 코드로 작성해줘.  
대출, 반납, 조회 기능이 포함되도록 작성해줘.  
액티비티 디어그램과, 시퀀스 디어그램을 그려줘



아래는 도서 대여 시스템의 액티비티 디어그램과 시퀀스 디어그램을 위한 PlantUML 코드입니다. 대출, 반납, 조회 기능이 포함되어 있습니다.

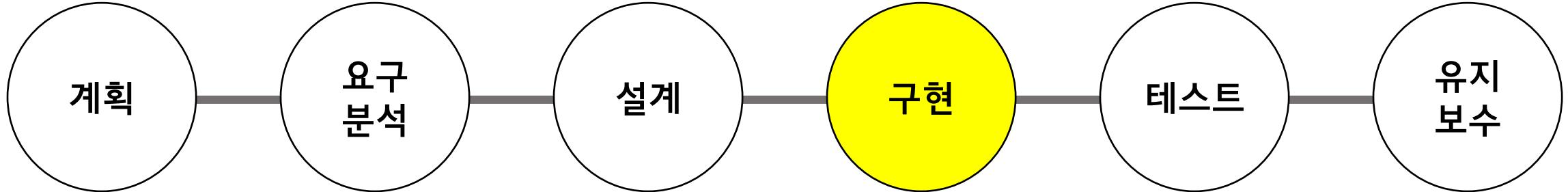
## 1. 액티비티 디어그램 PlantUML 코드

plantuml

```
@startuml
[사용자]
start
:도서 조회;
if (도서가 있는가?) then (예)
    :도서 선택;
    if (대출할 것인가?) then (예)
        :대출 요청;
        :대출 처리;
        :대출 완료;
    else (아니오)
●
```

코드 복사

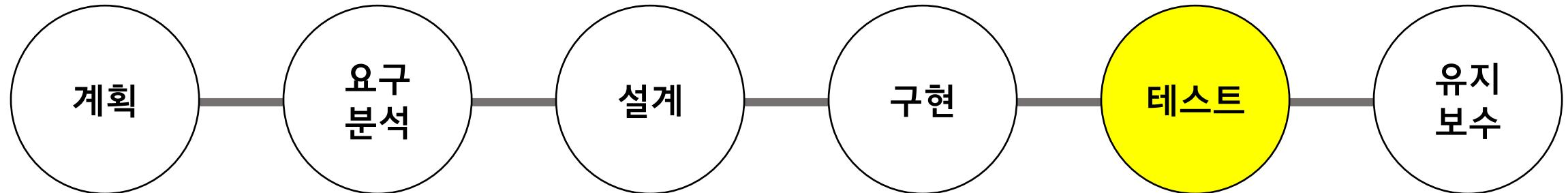
# 소프트웨어 개발과정



## • 구현 (Implementation)

- 목적: 설계된 내용을 바탕으로 실제 코드를 작성하고, 시스템을 개발하는 단계
- 주요 활동: 모듈별 코딩 및 개발, 데이터베이스 구현, 외부 시스템 또는 API 통합, 코드 리뷰 및 품질 관리
- 결과물: 개발된 소프트웨어 모듈, 소스 코드, 실행 파일

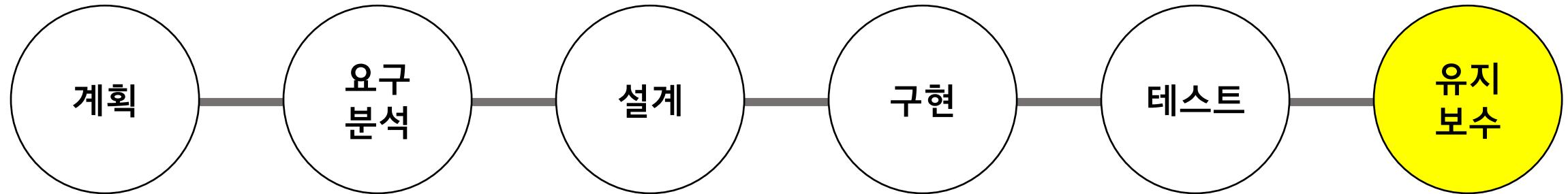
# 소프트웨어 개발과정



## • 테스트 (Testing)

- 목적: 개발된 소프트웨어가 요구사항을 충족하고, 오류가 없는지 확인하는 단계
- 주요 활동: 단위 테스트(각 모듈이 독립적으로 올바르게 동작하는지 테스트), 통합 테스트(모듈 간의 상호 작용을 테스트), 시스템 테스트(전체 시스템이 요구사항을 만족하는지 확인)
- 결과물: 테스트 보고서, 결함 보고서, 테스트 계획서

# 소프트웨어 개발과정



## • 유지보수 (Maintenance)

- 목적: 소프트웨어가 배포된 이후, 지속적인 유지보수 및 업데이트를 통해 시스템의 안정성과 성능을 보장하는 단계
- 주요 활동: 버그수정, 시스템 업데이트, 성능 향상, 보안 패치, 고객지원
- 결과물: 패치 릴리즈, 버전 관리, 유지보수 계획서

# Github Copilot

- 학생 등록 시, 무료로 사용 가능
  - <https://github.com/education/students>
- VS Code와 연동하여, 주석만으로 소스코드 제작

The screenshot shows a dark-themed instance of VS Code. The main editor area displays a Python file named `YoloV4.py`. The code implements a YOLOv4 object detection system, including functions for reading images, detecting objects, and drawing bounding boxes with labels. A conditional block at the bottom handles a test example by loading `cat.jpg`, creating a `yolo` object, and displaying the result. The bottom right corner of the editor shows the status bar with "Python 3.10.4 (conda)". Below the editor is the terminal window, which shows a log of network requests from IP 127.0.0.1, indicating interactions with a local web server. The bottom of the screen features the standard VS Code navigation bar with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, CODEWHISPERER, and REFERENCE LOG.

```
File Edit Selection View Go Run Terminal Help
YoloV4.py 2
D:\> python > software_engineering > lec7 > YoloV4py > ...
37     x = int(center_x - w / 2)
38     y = int(center_y - h / 2)
39     boxes.append([x, y, w, h])
40     confidences.append(float(confidence))
41     class_ids.append(class_id)
42     indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)
43     font = cv2.FONT_HERSHEY_PLAIN
44     for i in range(len(boxes)):
45         if i in indexes:
46             x, y, w, h = boxes[i]
47             label = str(self.classes[class_ids[i]])
48             color = self.colors[class_ids[i] % len(self.colors)]
49             cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)
50             cv2.putText(img, label, (x, y + 30), font, 3, color, 3)
51     return img
52
53 # using above class make a test example
54 if __name__ == "__main__":
55     # load image and convert to numpy array
56
57     img = cv2.imread("./cat.jpg")
58
59     # detect objects
60     yolo = YoloV4()
61     result_img = yolo.detect(img)
62
63     # show image
64     cv2.imshow("Image", result_img)
65     cv2.waitKey(0)
66     cv2.destroyAllWindows()

PROBLEMS ② OUTPUT DEBUG CONSOLE TERMINAL PORTS CODEWHISPERER REFERENCE LOG
127.0.0.1 - - [18/Oct/2023 01:52:34] "GET /uploads/catdog_detected.jpg HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2023 01:53:20] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2023 01:53:20] "GET /uploads/fruitdog_detected.jpg HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2023 01:53:20] "GET /favicon.ico HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2023 01:54:00] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [18/Oct/2023 01:54:00] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2023 01:54:00] "GET /uploads/fruitdog_detected.jpg HTTP/1.1" 200 -
* Detected change in 'D:\\python\\software_engineering\\lec7\\YoloV4.py', reloading
* Restarting with stat
* Debugger is active!
* Debugger PIN: 115-828-834
master 0 0 0 AWS
Ln 55, Col 18 Spaces: 4 UTF-8 CRLF Python 3.10.4 (conda) 52
```

# ChatGPT API 활용

---

- Python 코드 내 ChatGPT 활용을 원하는 경우(챗봇 개발 등), API를 활용해야 함 (유료)
- OpenAI API key 필요
  - <https://platform.openai.com/account/api-keys>
  - 우측 상단의 Personal -> View API Keys에서 확인

# ChatGPT API 활용

- 다음과 같은 순서로 코드 작성

1. 라이브러리 import
2. Prompt 생성
3. Openai의 ChatCompletion.create 메소드에 모델명과 프롬프트 전달

```
import openai

openai.api_key = ''

prompt = []
user_input = input('키보드 입력 : ')
prompt.append({'role': 'user', 'content': user_input})
response = openai.ChatCompletion.create(model='모델명 작성', messages=prompt)
response_text = response.choices[0].message.to_dict()['content']
print("\n"+ "답변: " + response_text )
```

# ChatGPT API 활용

- 응답 예측코드

```
response = openai.ChatCompletion.create(model='모델명 작성', messages=prompt)
```

- 모델명 확인

- <https://platform.openai.com/docs/models/embeddings>

- 주요 파라미터

- temperature: 0~2 사이의 값으로 낮으면 일관성 중요시, 높으면 창의력 중요시
- max\_tokens: 생성되는 텍스트의 최대 길이 지정 (제한하기 위해 사용)
- top\_p: 상위 몇%의 답변을 할지 정하도록 유도
- n: 생성할 답변의 수

# ChatGPT API 활용

- Use Case별 파라메터 추천

Use Case	Temperature	Top_p	Description
Code Generation	0.2	0.1	Generates code that adheres to established patterns and conventions. Output is more deterministic and focused. Useful for generating syntactically correct code.
Creative Writing	0.7	0.8	Generates creative and diverse text for storytelling. Output is more exploratory and less constrained by patterns.
Chatbot Responses	0.5	0.5	Generates conversational responses that balance coherence and diversity. Output is more natural and engaging.
Code Comment Generation	0.3	0.2	Generates code comments that are more likely to be concise and relevant. Output is more deterministic and adheres to conventions.
Data Analysis Scripting	0.2	0.1	Generates data analysis scripts that are more likely to be correct and efficient. Output is more deterministic and focused.
Exploratory Code Writing	0.6	0.7	Generates code that explores alternative solutions and creative approaches. Output is less constrained by established patterns.

<https://community.openai.com/t/cheat-sheet-mastering-temperature-and-top-p-in-chatgpt-api-a-few-tips-and-tricks-on-controlling-the-creativity-deterministic-output-of-prompt-responses/172683>

# ChatGPT API 활용

- 챗봇 개발
  - 지속적인 대화가 이루어질 수 있는(이전 대화를 기억하는) 형태로 작성하는 방법

```
import openai

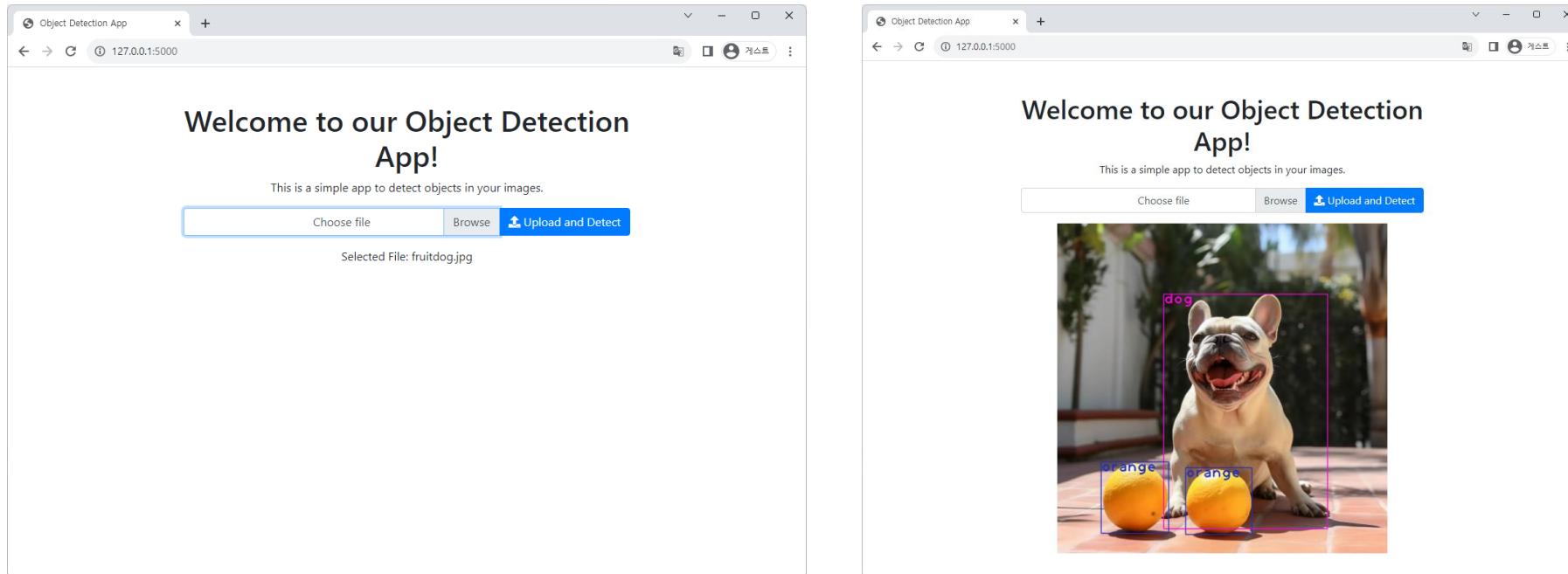
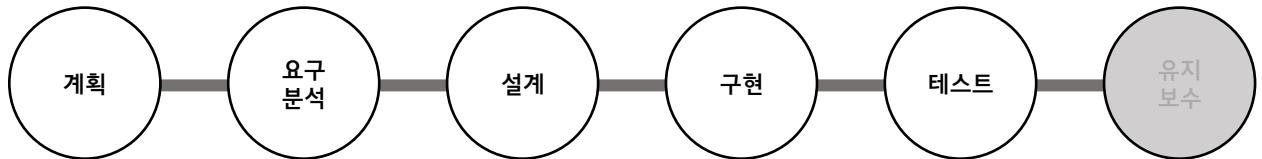
openai.api_key = ''

prompt = []

while True:
    user_input = input('키보드 입력 : ')
    prompt.append({'role': 'user', 'content': user_input})
    response = openai.ChatCompletion.create(model='모델명 작성', messages=prompt)
    prompt.append(response.choices[0].message.to_dict())
    response_text = response.choices[0].message.to_dict()['content']
    print("\n"+"답변: " + response_text )
```

# 실습 - 생성형 AI 활용 SW 개발

- 주제: 데이터(파일)을 분석하는 SW제작 (데이터는 자유롭게 선정, 개발은 AI 활용)
- 세부: 유지보수를 제외한 소프트웨어 개발
- 팀: 개인 또는 2인으로 구성



예시: Python, Flask, html, OpenCV, YOLO 활용 이미지 분석 웹페이지

# 실습 1. 개발 계획 세우기

## • 개발항목 정의

- ChatGPT등을 활용하여 작성할 것, 실습 파일에 입력 Prompt에 작성한 내용과 생성결과, 직접 수정한 부분을 표시할 것

## • 일정 계획수립 및 비용산정 (제품 판매를 가정하고 비용 산정)

2023년 5월						
월요일	화요일	수요일	목요일	금요일	토요일	일요일
1	2	3	4	5	6	7
				오후 5:00 월정 제작		
8	9	10	11	12	13	14
	오후 7:00-8:15 팀 회의			오전 11:00 개념 검증 기한		
15	16	17	18	19	20	21
오전 11:00 대략 개요 기한	오후 7:00-8:15 팀 회의			오후 5:00 주제기술문 개요 기한		
22	23	24	25	26	27	28
오전 11:00 발표 개요 기한	오후 7:00-8:15 팀 회의	오전 11:00 슬라이드 기한		오후 5:00 최종보고서 초고 기한		
29	30	31	6/1	2	3	4
	오후 7:00-8:15 팀 회의		오후 5:00 최종보고서 제출 기한			

[그림 2-3] 팀 일정표

선행책임도표	기온	나훈	다미	라나	튜터	외부 전문가	기호
1.0 주제 브레인스토밍	1						1=일차 책임
1.1 문제 기술 명료화	1	2	2	2			2=보조/작업
1.2 연구 수행	1	2		2		4	3=자문 필수
2.0 기능 요구사항 분석	2	2	1	2	5	3	4=자문 가능
3.0 대안 생성				1			5=검토
4.0 대안 평가	5	1	2	2			6=최종승인
5.0 선호하는 설계 선정	1	2			5	4	
6.0 설계결과 문서화		1					
7.0 프로젝트 관리	1						
7.1 주간회의	1	2	2	2			
7.2 프로젝트 계획 개발	1	2	2	2			
7.3 진척도 추적	1				5		
7.4 진도보고서	1						

[그림 2-4] 선행책임도표(LRC)

## 실습 2. MRD 및 PRD 작성

---

- 개발하고자 하는 항목에 대해 MRD와 PRD를 작성하기
  - MRD 주요항목(사업팀 관점 작성): 타겟 시장의 설명, 고객의 문제점 및 요구사항, 시장의 크기와 성장 예측, 경쟁 분석, 제품의 위치 및 가치 제안 등
  - PRD 주요항목(Project Manager 관점 작성): 제품의 기능적 요구사항, 사용자의 경험과 인터페이스에 관한 요구사항, 성능과 다른 시스템과의 통합에 대한 요구사항, 제품의 제약사항 등
- ChatGPT를 활용하고, 실습 파일에 입력 Prompt, ChatGPT 생성결과, 수정사항을 작성

## 실습 3. 설계도(UML) 그리기

---

- ChatGPT를 활용하여 액티비티 다이어그램과, 시퀀스 다이어그램을 제작 (PlantUML코드)
- 시각화 웹페이지를 활용하여 이미지 파일 저장하기
  - <https://www.plantuml.com/plantuml/uml/>
- 실습 파일에는 다이어그램 이미지만 작성

## 실습 4. 구현 및 테스트

- 생성형 AI 도구를 활용하여 소스코드 구현하기
- 구현 완료 된 경우, 테스트 케이스를 만들어서 해당 시나리오에 잘 작동하는지 확인
- 실습 파일에 소스코드 생성을 위해 사용한 입력 Prompt와 결과 예시 5건 이상 작성
- 주제 선정에 어려움이 있는 경우, 이미지 또는 csv등의 table 데이터를 분석하는 코드를 작성할 것 (예시 화면의 내용을 구현해도 됨)

