

---

# Open Source SW Contribution

---



## 중간 보고서

제출일	2023.05.09	담당교수	송인식 교수님
과목	오픈소스SW기여	프로젝트명	Dr.Bot
학번 / 이름			
32197256 박성우		32173575 이해주	

## 목차

1. 프로젝트명
2. 분석
  - 2.1. User Scenario
  - 2.2. Use Case Diagram
  - 2.3. 서비스 흐름도
3. 사용기술
4. 상세 설계
  - 4.1. 시스템 설계
  - 4.2. 프로그램 설계
5. 테스트 플랜
  - 5.1. 테스트 정의
  - 5.2. 개발 단계와 테스트 단계의 관계
  - 5.3. 단계별 수행내용
  - 5.4. 테스트 시나리오
    - 5.4.1. 단위테스트
    - 5.4.2. 통합테스트
    - 5.4.3. 시스템테스트

## 1. 프로젝트명

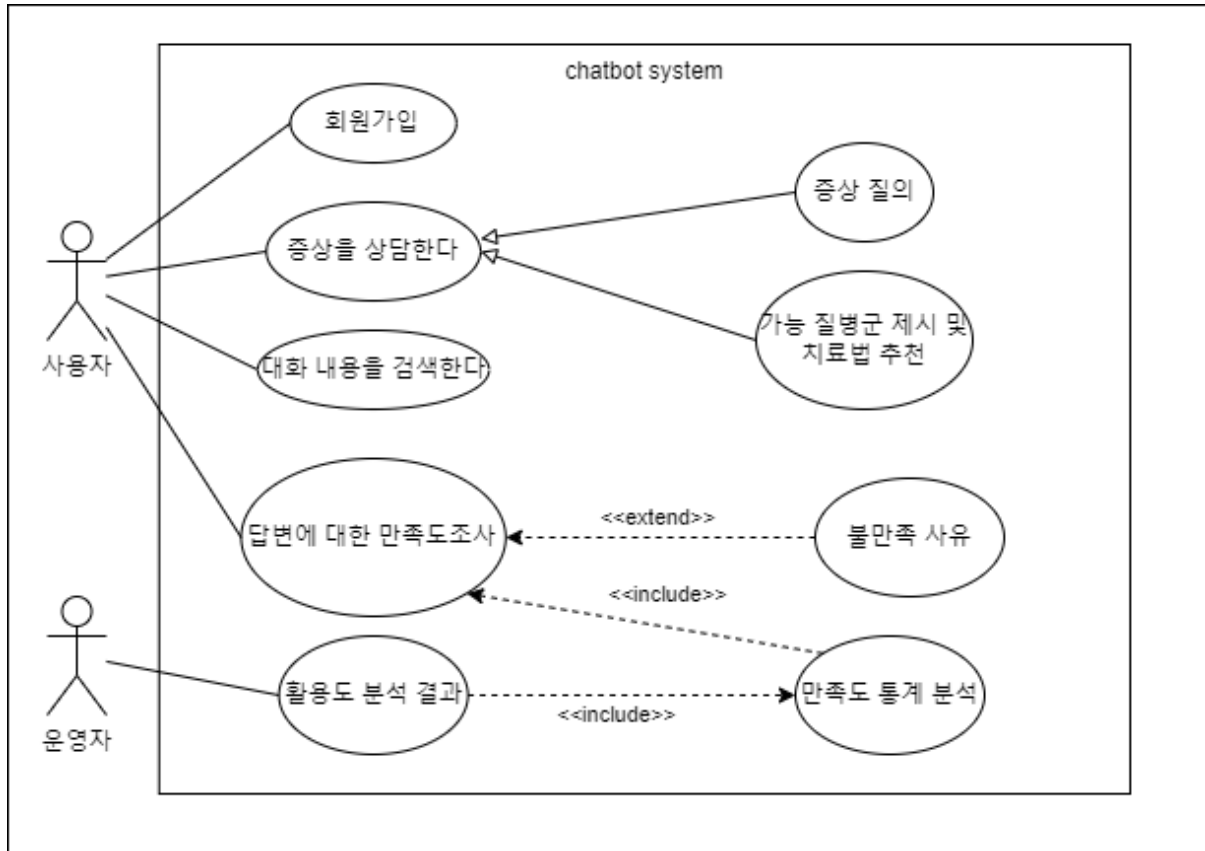
“의학정보 제공 챗봇 제작” (Dr.Bot)

## 2. 분석

### 2.1. User Scenario

구성요소	설명
사용자 그룹	의료서비스 관련 전문적인 답변이 필요한 스마트폰 이용자
시놉시스	1) A 씨는 원인 모를 복통으로 불편함을 호소하고 있으며 병원에 갈만큼 아프지만 현재 업무량이 많아 당장 병원을 갈 수 없는 상황이다. 따라서 간단하게라도 진단을 받고자하며 증상이 완화되기를 원한다.  2) B 씨는 장시간 두통으로 인해 약국 약을 복용하였으나 증상이 완화되지 않았다. 병원에 방문할 만큼 아픈 것은 아니지만 증상이 잘 낫지 않기 때문에 원인과 도움이 되는 해결법을 제공받고자 한다.
니즈	병원에 가지 않고 통증의 원인이 될 수 있는 요인들을 제공받고 증상 완화와 치유 촉진에 도움이 되는 방법들을 제안 받고자 한다.
불편사항	인터넷에서는 환자에게 주어진 상황에 맞는 통증 원인을 정확히 알아낼 수 없으며 해당 정보의 정확성을 알 수 없기 때문에 신뢰성이 떨어진다. 하지만 의료 챗봇을 이용한다면 사용자가 언급한 증상들을 기반으로 알맞는 정보를 제공할 수 있다.
대안마련	사용자가 알고자 하는 증상의 원인을 간단한 채팅(키워드)로 파악할 수 있으며 높은 신뢰성을 제공하며 시간이 없는 사람들은 병원에 가지 않고도 어느정도 자신의 문제를 파악하여 증상을 완화시킬 수 있다.

## 2.2. Use Case Diagram



액터명	역할	비고
사용자	<ul style="list-style-type: none"> <li>챗봇을 사용하여 요구사항을 수행하는 행위자를 말한다.</li> </ul>	
운영자	유지보수 등 시스템을 관리하는 행위자를 말한다.	

### 1. 개요

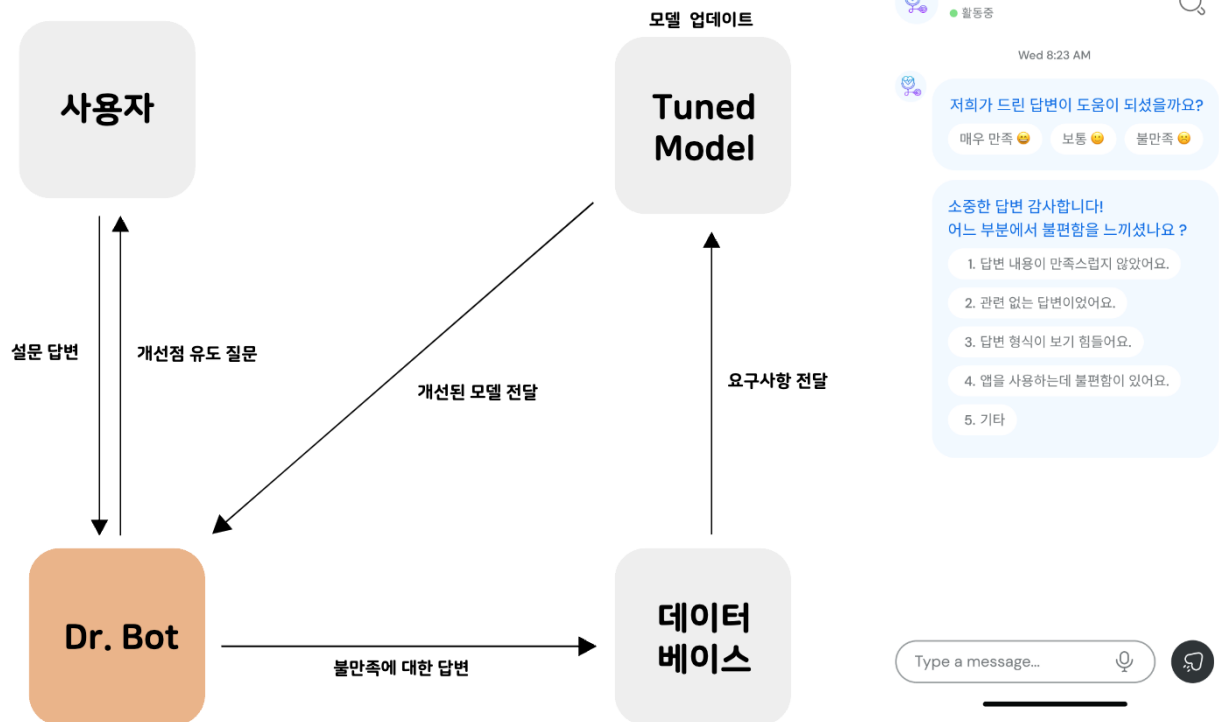
- 해양수산부 본부 사용자와 각 지방해양수산청 사용자는 바다넷을 통한 링크를 통하여 통합인증을 통하여 로그인한다.

### 2. 주요 액터(Primary Actor)

사용자, 운영자

<p>3. 전제 조건(Preconditions)</p> <p>- 어플리케이션이 실행된 상태여야 한다.</p>
<p>4. 종료 조건(Postconditions)</p> <p>- 사용자가 종료 명령어를 입력한다.</p>
<p>5. 기본 흐름(Basic Flow)</p> <p>사용자는 Dr.Bot에 로그인한다.(id/pwd)  사용자의 신체적 증상에 대해 상담한다.  Dr.Bot의 답변을 받는다.</p>
<p>6. 대안적 흐름(Alternative Flows)</p> <p>없음.</p>
<p>7. 특수 요구사항(Special Requirements)</p> <p>없음</p>
<p>8. 관련정보</p> <p>없음</p>

### 2.3. 서비스 흐름도



사용자가 앱을 다 사용하고 나면 플랫폼에서는 사용자에게 개선점을 위한 질문을 한다. 개선점을 위한 설문은 만족 / 보통 / 불만족으로 이루어지며 만족 혹은 보통일 경우에는 데이터베이스에 반영되지 않으며, 불만족일 경우에 추가적인 질문이 위 사진과 같이 이루어진다.

사용자가 “Dr.Bot” 앱을 실행하여 사용하면 매 답변의 마지막에 만족도를 조사하도록 설계할 예정이다. 이에 사용자가 불만족을 눌렀을 경우 개선점을 유도하는 질문을 피드백한다. 이후 사용자가 선택한 버튼을 DB에 저장하여 개선 방향성을 설정하고 그 방향성을 토대로 챗봇 모델을 Tuning 한다. 업데이트된 모델을 다시 제공함으로써 이전의 불편사항을 반영하여 개선되고 있음을 느낄 수 있도록 구현하고자 한다.

### 3. 사용기술

#### 가) LLM (Large Language Model)

대형언어모델은 GAN과 함께 AI분야에서 최근 폭발적인 발전을 이끈 신경망이다. 오픈AI의 GPT 시리즈, 구글의 팜(PaLM) 또는 메타의 라마(LLaMa) 등이 LLM으로 방대한 텍스트 데이터를 사용해 훈련된다. LLM은 고품질의 웹 문서, 책, 위키피디아의 기사들, 블로그 글과 깃허

브의 오픈소스 코드 등 공개된 데이터를 활용해 학습한다. 이를 토대로 텍스트가 주어지면 다음에 어떤 텍스트가 올지를 확률적으로 예측해 내고 그 결과 더 긴 텍스트가 생성되면 또 다시 다음 텍스트를 예측하는 과정을 되풀이한다.

#### 나) 미세조정 (Fine tuning)

GPT 시리즈와 같은 대형 AI는 범용성이 뛰어난 기초(foundational) 모델이기 때문에 특정한 AI도구 개발에 쓸 때 이용할 수 있다. 이때 미세조정이 필요하다. 기초 모델에 특정 데이터를 추가로 학습시켜 적은 비용으로 매우 특화된 기술을 가진 모델을 개발할 수 있다.

Chat GPT에서 사용하는 데이터는 2021년 까지이다. 2022년도부터의 질문에는 답하지 못한다. 따라서 원하는 format으로 원하는 답이 나오도록 유도하기 위해 fine-tuning 과정이 필요하다.

Fine-tuning절차로는 먼저 format을 맞추기 위한 데이터를 준비해야 한다. 그 다음 준비한 데이터를 토대로 모델을 생성하고 생성된 모델을 활용하여 챗봇을 구축한다. 이때 openAI playground에서 사용자가 직접 만든 모델을 업로드하거나 다른 사람이 업로드해 놓은 모델을 사용할 수 있다. Fine-tuning을 위해 준비해야 할 데이터로는 JSONL 형태의 데이터와 파일 안에는 prompt, completion의 형태로 구성해야 한다. Prompt는 원하는 프롬프트를 구성하고, completion은 원하는 답의 형태를 구성한다.

#### 다) 퓨샷러닝 (Few-Shot Learning)

의료 분야와 같은 충분한 데이터를 구할 수 없거나, 레이블 데이터셋 생성에 많은 시간적, 비용적 부담이 요구되는 분야에서 Few-Shot Learning(FSL) 학습방식을 도입하여 인간의 추론과정을 모방하여 제한된 데이터로도 별도의 증강 없이 다양한 태스크에서 높은 일반화 성능을 달성하여 한계점을 극복하고자 하는 학습방식이다.

#### 4. 상세설계

##### 4.1. 시스템 설계

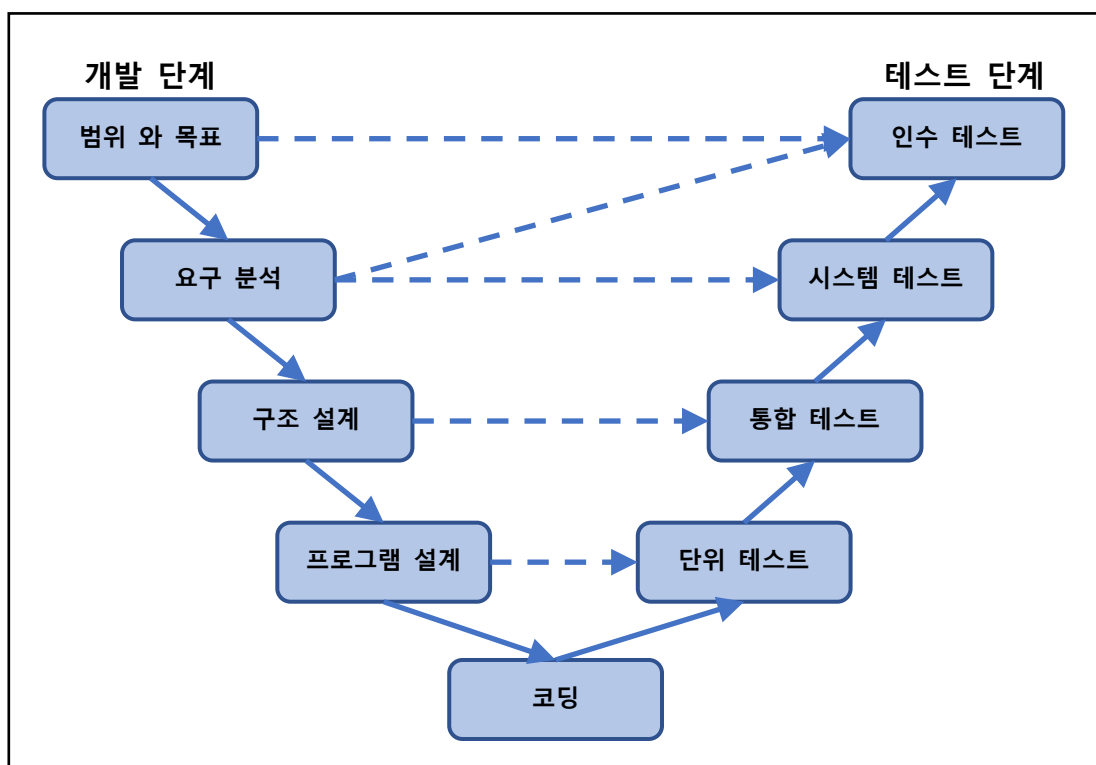
##### 4.2. 프로그램 설계

#### 5. 테스트 플랜

##### 5.1. 테스트의 정의

시스템이 정해진 요구를 만족하는지, 예상과 실제 결과가 어떤 차이를 보이는지 수동 또는 자동 방법을 동원하여 검사하고 평가하는 일련의 과정을 말한다.

##### 5.2. 개발 단계와 테스트 단계의 관계 (V 모델)



V 모델은 사용자의 요구사항을 정의하고 구체화, 상세화 하는 각 개발 단계에 따라 그에 상응하는 단계별 테스트를 수행하여 개발 산출물(소프트웨어)에 대한 요구사항 충족도를 검증하는 모델이다.



### 5.3. 테스트 단계별 수행 내용

테스트 단계	주요 수행내용
단위 테스트	단위 테스트는 코딩 직후 소프트웨어 설계의 최소 단위인 '모듈'이나 '컴포넌트'에 초점을 맞춰 테스트 사용자의 요구사항을 기반으로 기능성 테스트를 최우선으로 수행한다.
통합 테스트	통합 테스트는 단위 테스트가 완료된 모듈들을 결합하여 하나의 시스템으로 완성시키는 과정의 테스트 통합 테스트는 모듈 간, 통합된 컴포넌트 간의 상호 작용 오류를 검사한다.
시스템 테스트	개발된 소프트웨어가 해당 컴퓨터 시스템에서 완벽하게 수행하는지를 점검하는 테스트 환경적인 장애 리스크를 최소화하기 위해 실제 사용 환경과 유사한 테스트 환경에서 테스트 해야한다.
인수 테스트	인수 테스트는 개발한 소프트웨어가 사용자의 요구사항을 충족하는지에 중점을 두고 수행하는 테스트 인수 테스트는 개발한 소프트웨어를 사용자가 직접 테스트한다.

### 5.4. 테스트 시나리오

#### 5.4.1. 단위테스트

<https://docs.google.com/spreadsheets/d/1gdXJVQHtY6bSitcdp7CNkwtmWjTixxSsHud08weKp0U/edit?usp=sharing>

			테스트 결과 범례: pass - 성공, fail - 실패, N/A - 테스트 불가, 기타				
업무		테스트 시나리오			테스트 수행결과		
분류	테스트 ID	케이스 설명	테스트 데이터	예상결과	테스트결과	테스트일자	성명
backend	BE-001	이름 / 생년월일을 입력하면 DB 에 저장되는가?		이름 / 생년월일을 입력하면 DB 에 저장됨			
	BE-002	DB 에서 각 메시지에		DB 에 저장할때 각			

		고유키값을 부여하는가?		메시지에 고유키값이 부여됨			
	BE-003	user 와 assistant, 전체 message 로 분류 하여 저장되는가?		user 와 assistant, 전체 message 로 분류 하여 저장됨			
	BE-004	메시지 기록을 잘 불러오는가?		메시지 기록을 잘 불러옴			
	BE-005	API 에 설정된 persona 값이 적용이 잘 되었는가?	"자기소개해주 세요"	지정된 이름과 역할을 출력한다.			
	BE-006	만족도 조사 선택항목이 DB 에 저장되었는가?		만족도 조사 선택항목이 DB 에 저장됨			
frontend	FE-001	메시지 전송 버튼 클릭 시 이벤트가 잘 발생 되는가?		텍스트필드에 담긴 메시지가 뷰레이아웃에 잘 전송되어 나타난다.			
	FE-002	검색 버튼 클릭시 이벤트가 잘 발생 되는가?		이전 메시지 기록 검색을 할 수 있는 상태로 바뀐다.			
	FE-003						
	FE-004						
	FE-005						

#### 5.4.2. 통합테스트

<https://docs.google.com/spreadsheets/d/1h8-iBOBPA82o0tzE7rmhJ1feZxQwXmajl4oLujVt6CQ/edit?usp=sharing>

테스트 결과 범례: pass - 성공, fail - 실패, N/A - 테스트 불가, 기타								
업무			테스트 시나리오			테스트 수행결과		
분류	테스트 ID	테스트명	케이스 설명	테스트 데이터	예상결과	테스트결과	테스트일자	성명
실행		실행	Dr.Bot 앱을 실행한다	앱을 터치한다.	의사 아이콘이 화면에 출력되며 함께 앱이 실행된다.			
					환영 인사와 함께 입력란과 검색 아이콘이 화면에 출력된다.			
사용		질문	사용자가 본인의 증상에 대해 포괄적으로 질문한다.	"머리가 아파요"	증상 발현 주기, 통증정도, 정확한 통증위치 등을 되묻는다.			
		사용자 요구사항	사용자가 통증 완화에 도움이 되는 대처법 3 가지를 질문한다.	"손목 통증이 오래 지속되는데 통증 완화에 도움이 되는 방법 3 가지 알려줘"	통증 완화에 도움이 되는 대처법 3 가지를 제공한다.			
					증상이 지속되거나 통증이 악화될시 병원방문 진료를			

					권고한다는 답변을 출력한다.			
		사용자 요구사항	사용자가 증상 완화에 도움이 되는 약 3 가지를 질문한다.	"두통이 지속되서 불편한데 증상 완화에 도움이 되는 약 3 가지 알려줘"	증상 완화에 도움이 되는 약 3 가지와 복용시 주의 사항을 출력한다.			
					약물 사용에는 특히 주의가 필요함을 명시한다.			
		이외 정보입력	의료정보와 관련없는 내용을 질문한다.	"자동차 엔진 3 가지 알려줘"	답변하되 의료정보 제공이 본역할임을 제시한다.			
		검색	키워드로 대화 기록을 검색한다.	검색(돋보기) 아이콘을 클릭하여 키워드를 입력한다.	키워드에 해당하는 글씨가 하이라이트 처리되어 출력되며 해당 위치로 화면이 이동한다.			
		만족도 조사	사용자에게 만족도 조사를 버튼식으로 제공한다.		만족도 조사 말풍선을 화면에 출력한다.			

#### 5.4.3. 시스템테스트

목록	테스트
기능 테스트	<ul style="list-style-type: none"> <li>• 모든 버튼이 필요에 따라 잘 작동하는지 확인합니다.</li> <li>• 사용자에게 네트워크 오류가 있을 때 마다 적절한 오류메시지가 표시되는지 확인 합니다.</li> <li>• 모든 iOS버전에서 요구사항에 따라 프로그램이 작동하는지 확인합니다.</li> </ul>
성능 테스트	<ul style="list-style-type: none"> <li>• 클라이언트-서버 구성 설정이 필요한 최적 성능 수준을 제공하는지 확인합니다.</li> <li>• 프로그램의 응답 시간이 요구사항에 맞는지 확인합니다.</li> <li>• 요구에 맞는 응답을 하는지 확인합니다.</li> <li>• 과거 메시지 기록을 잘 불러오는지 확인합니다.</li> </ul>
사용성 테스트	<ul style="list-style-type: none"> <li>• 버튼이 필요한 크기인지, 큰 손가락에도 적합한지 확인합니다.</li> <li>• 아이콘이 자연스럽게 프로그램과 잘 어울리는지 확인합니다.</li> <li>• 배경/텍스트필드를 탭 했을 때 키보드가 잘 작동되는지 확인합니다.</li> <li>• 글꼴 크기가 충분한지 확인합니다.</li> </ul>
복구성 테스트	<ul style="list-style-type: none"> <li>• 예상치 못한 중단 / 충돌 시나리오 후 복구 상황의 유효성을 검사합니다.</li> </ul>
호환성 테스트	<ul style="list-style-type: none"> <li>• 모바일 장치의 크기, 해상도, 화면, 버전, 하드웨어에 따라 인터페이스가 같은지 확인합니다.</li> </ul>
점검 사항	<ul style="list-style-type: none"> <li>• 앱을 적절한 시간 안에 설치할 수 있는지 여부를 확인합니다.</li> <li>• 프로그램 시작화면 확인합니다.</li> <li>• 프로그램이 백그라운드에서 실행되는동안 충전 효과에 대해 확인합니다.</li> <li>• 프로그램이 수행되는 동안 배터리 소비량을 확인합니다.</li> </ul>