

# 개인맞춤 AI 복약관리 애플리케이션 개발



202045804 김기윤

202055508 강태진

202055639 김재형

지도교수 박진선

---

# 목 차

1. 서론 .....	1
1.1 연구 배경 .....	1
1.2 기존 문제점 .....	1
1.3 연구 목표 .....	2
2. 연구 배경 .....	4
2.1 Elasticsearch .....	6
2.2 Redis .....	6
2.3 Weaviate .....	6
2.4 FCM .....	7
2.5 AES-GCM .....	7
2.6 TLS .....	7
2.7 Nginx .....	8
2.8 React .....	8
2.9 Next.js .....	8
2.10 RAG .....	9
2.11 Prompt Engineering .....	9
2.12 Orchestration AI .....	10
2.13 YOLO v8 .....	10
2.14 Jetpack Compose .....	11
2.15 Lifecycle - LiveData .....	11
2.16 ViewModel .....	12
2.17 MVVM .....	12

---

2.18 Coroutine	12
2.19 Hilt-Dagger	13
2.20 OkHttp	13
3. 연구 내용	14
3.1 의약품·건강기능식품 공공데이터 가공	14
3.1.1 의약품 공공데이터 파서	14
3.1.2 구어체 변환을 위한 의약품 데이터 블럭화	16
3.1.3 Batch API	16
3.1.4 건강기능식품 공공데이터 파서	19
3.2 사용자 정보 및 의약품 DB 구성	23
3.3 개인맞춤형 DUR	27
3.4 암호화 및 비식별화	30
3.4.1 AES-GCM	30
3.4.2 TLS	30
3.4.3 Nginx	30
3.4.4 산업체 멘토 의견 반영	31
3.5 자동완성 및 검색	31
3.5.1 Elasticsearch	31
3.5.2 DUR Tagging	32
3.5.3 이미지 데이터 및 YOLO v8	32
3.6 약품 정보 및 맞춤형 안내	33
3.6.1 구어체 변환(GPT API)	33
3.6.2 Prompt Engineering	33
3.6.3 개인 맞춤 안내	34
3.7 복약 관리	35
3.7.1 복약 완료율 시각화	35
3.7.2 FCM을 활용한 복약 알림	35

---

3.8 문진표 및 QR 촬영	36
3.8.1 React	36
3.8.2 Next.js	36
3.9 피어링 및 계정 전환	37
3.9.1 친구 맺기(페어링)	37
3.9.2 가족 관리(계정 전환)	38
3.10 AI 챗봇	38
3.10.1 Orchestration	39
3.10.2 자연어 처리(NLP)	39
3.10.3 RAG	39
3.10.4 DUR Tool	39
4. 연구 결과 분석 및 평가	40
4.1 개발 일정	40
4.2 역할 분담	42
4.3 워크스페이스	42
4.4 API 명세서	42
4.5 서비스 아키텍처	48
4.6 수상 및 전시이력	48
5. 결론 및 향후 연구 방향	48
5.1 결론	48
5.2 향후 연구 방향	49
5.3 맺음말	49
6. 참고 문헌	50

김기운 강태진 김제형

최종보고서

필팁

완벽한 당신만의 약사님

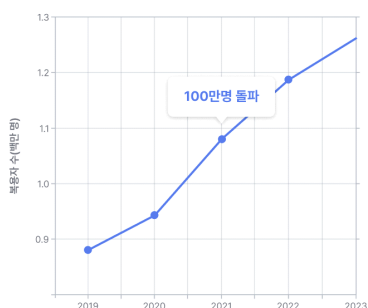
# Pilltip



# 1. 서론

## 1.1 연구 배경

대한민국은 빠른 고령화와 함께 만성질환을 가진 환자가 증가하면서 다약제 복용 인구가 꾸준히 늘고 있다. OECD 통계에 따르면 75세 이상 한국 노인의 70.2%가 5개 이상의 약물을 만성적으로 복용하고 있어 참여국 평균치(48.3%)를 크게 웃돌며 최고 수준을 기록했다. 이러한 다약제 복용은 약물 상호작용, 처방 오류, 부작용 등의 위험을 높이는 요인으로 작용한다. 그럼에도 불구하고 일상에서 환자들이 접하는 의약품 포장이나 처방전의 복약 안내 정보는 글자 크기가 매우 작고 전문용어가 많아 이해하기 어려운 형식이며, 획일적으로 제공되고 있다. 분명 건강을 위한 약품이지만 정작 복용자들은 필요한 정보를 쉽게 얻지 못하는 실정이다.



<연도별 다제약물 복용자 수 (백만 명)>  
한번에 10개 풀어...다제약을 환자, 작년 128만명 기록 1대일리광 기사 2024-07-15

2025년 7월 기준, 10개 이상 의약품  
60일 이상 복용 환자 136만명, 65세 이상 노인 평균 복용 수 5.3개

만성질환 환자 매우 증가, 다약제 복용자 2019년 약 84만명 ...  
2023년 기준 129만명까지 증가

그림 2 다약제 복용 현황

## 1.2 기존 문제점

그렇다면 우리가 약에 대해 잘 모르더라도 안심하고 복용할 수 있도록 도와주는 제도적 장치는 충분할까? 안타깝게도 현실은 그렇지 않다. 약물 간 상호작용이나 오남용을 막기 위한 의약품안전사용서비스(DUR)가 존재하지만, 의료기관에서 DUR 시스템을 의무적으로 사용하도록 하는 법안은 여러 사정으로 10년 넘게 국회에서 계류 중이다. 결국 DUR 시스템이 있어도 현장에서 충분히 활용되지 못하고, 병원 간 처방 이력도 원활히 공유되지 않아 환자들은 여전히 복약 안전의 사각지대에 놓여 있다.

		2018년	2019년	2020년	2021년	2022년	2023년 상반기
전체 사고 건수		6,553	11,953	13,919	13,919	14,820	10,934
약물 관련 사고	건수	1,676	3,798	4,325	4,198	6,412	5,777
	비율	25.6%	31.8%	31.1%	31.9%	43.3%	52.8%

[의료기관 환자 안전사고 추이] - 출처 (연합뉴스\_2024-04-22)

그림 3 의료사고 중 약물 관련 사고 발생 비율

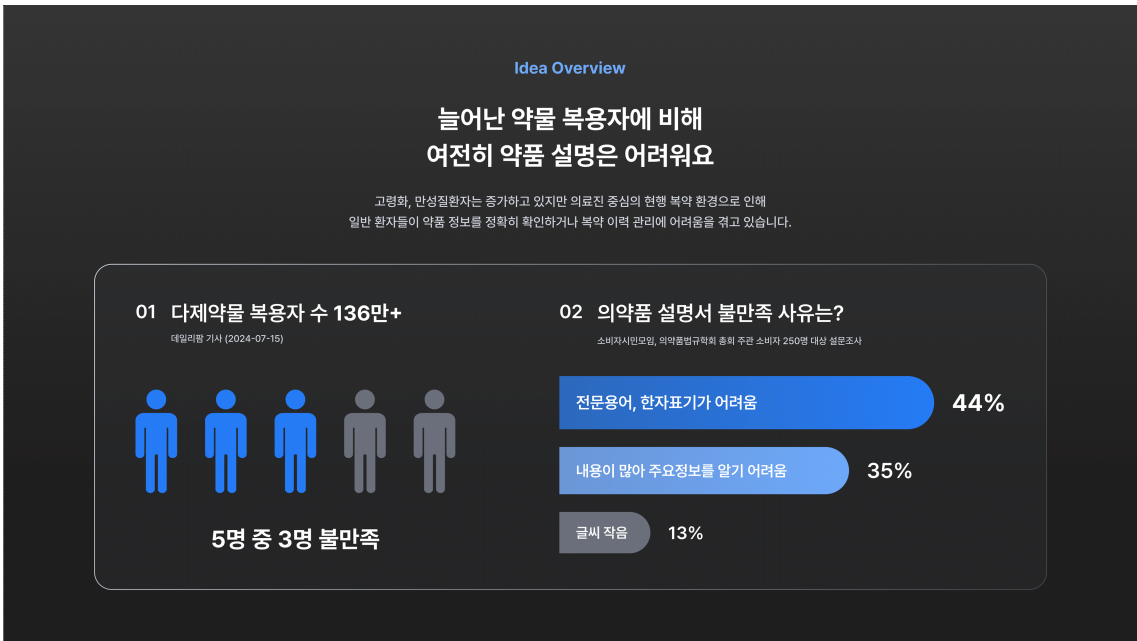


그림 4 의료지식이 없는 일반인에게는 어려운 약품 설명

또한, 해당 DUR 시스템은 전문 의료인력만이 다룰 수 있어 일반인의 접근성이 현저히 떨어지고, 복합제제에 대한 안전정보 부재 등 정보 구조 표준화 부족으로 인해, 단일 성분에 대해서만 금기 기준이 있고 복합성분 의약품에는 금기 기준이 마련되지 않아 2008~2009년 약 3,538건의 병용금기·연령금기 처방이 발생한 것으로 식약처에 보고되었다.

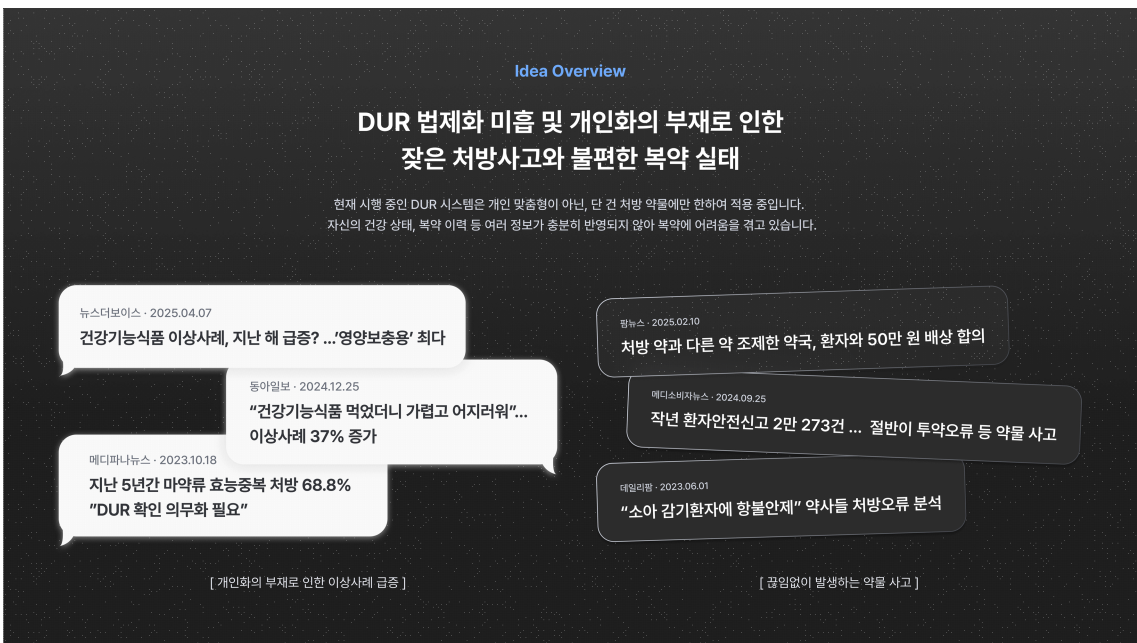


그림 5 획일화된 복약 안내와 최소한의 사회적 복약 안전망의 부재

DUR 사용이 법적 의무가 아니기에 일부 현장에서의 사용 누락으로 인해 완전한 연계가 이뤄지지 못하는 문제도 존재한다. DUR 점검 항목에는 의사 처방 의약품과 약국 조제의약품 위주의 정보만 반영되고, 건강기능식품이나 일반의약품 복용 정보는 포함되지 않는다. 그 결과 환자가 건강기능식품이나 영양보조제를 병용하여 발생할 수 있는 상호작용 위험을 DUR에서는 사전에 경고하지 못한다.

이에 더불어, 우리의 소중한 개인 의료정보마저 안전하지 못한 실정이다. 디지털 전환이 활발한 지금, 제도적 이슈로 여전히 사용되는 종이 처방전과 문진표 등에는 민감한 개인정보가 빼곡히 담겨 있지만, 관리 부실로 이를 폐기할 때 파쇄하지 않고 버리는 사례까지 종종 발생해 환자 정보가 유출되는 일이 반복되고 있다. 환자가 본인의 복용 이력과 건강 정보를 일일이 기억하여 문진표를 적어야 한다는 것도 큰 불편함 중 하나이다.



그림 6 기존 복약 서비스 분석

한편 시장조사를 통해 확인한 바에 따르면, 개인의 건강 상태와 복용 이력에 맞춘 개인맞춤형 복약 솔루션과 DUR을 통한 복약 안전관리, 개인정보 보호, 의료기관 연계 기능을 모두 갖춘 올인원 플랫폼은 시장에 존재하지 않는다. 즉, 앞서 언급한 복합적인 문제들을 한꺼번에 해결할 수 있는 포괄적 맞춤형 복약 관리 서비스의 부재가 현 상황이다.

### 1.3 연구 목표

상기 문제들은 전 국민이 겪고 있는 현실이며, 특히 고령층과 영유아가 가장 취약한 계층이다. 이들은 디지털 기기 활용이 서툰 경우가 많아, 보호자나 가족이 대신 정보를 확인하고 복약을 관리해주는 상황이 빈번하다. 이러한 배경을 고려해 본 프로젝트는 나와 가족의 건강관리에 관심이 많은 20-40대를 주 사용자층으로 설정하였다. 20-40대 젊은 성인들이 Pilltip을 통해 70-80대 부모님이나 어린 자녀의 복약까지 함께 관리함으로써, 직간접적으로 전 세대가 혜택을 볼 수 있는 개인맞춤 AI 복약 관리 서비스를 목표로 하고 있다.

#### Persona

### 나와 내 가족 자가 건강 관리에 관심이 많은 2040 유저

해당 유저들은 자가 건강 관리에 관심이 높은 만큼 건강기능식품과 영양제를 많이 복용하고 있어요.  
필팁은 이들의 복약 관리와 건강 관리 니즈를 충족시킬 수 있는 AI 기반 개인화된 복약 관리 서비스를 제공하고자 합니다.

**유형 01: 감기 걸린 임산부**  
**김필팁** | 31세 (여)  
" 임신 20주차 임산부예요!  
감기 기운이 있어서 집에 있는 감기약을 복용하려는데  
이부프로펜이 포함된 약을 복용해도 괜찮을까요? 궁금해요..!"

**Pilltip**  
이부프로펜은 임신 후기의 경우, 태아의 혈관 문제를 유발할 수 있어요.  
때문에 임산부인 김필팁님은 섭취에 주의가 필요해요.  
임산부에게 맞는 약이 필요하시면 산부인과에 방문하여 의사와 상담 후 복용해주세요!

**유형 02: 다중 복용 중인 직장인**  
**김부산** | 45세 (남)  
" 챙겨먹는 영양제가 많아지면서 다른 의약품이나 영양제와  
함께 복용해도 되는지 확인하기 어려워요. 제가 필수 영양소를  
잘 복용하고 있는 걸까요? 부작용이 생길까 불안해요...!"

**Pilltip**  
유저의 복용 환경을 고려해 완벽한 복약을 도와주는 전문가, 필팁을 추천드려요!  
필팁은 개인의 건강정보와 복용 이력을 바탕으로, 맞춤형 DUR 분석까지 반영해요.  
세상에 하나뿐인 개인맞춤형 건강기능식품 AI 돌인원 큐레이팅을 만나보세요!

그림 7 필팁 페르소나

본 프로젝트 "Pilltip(필팁)"은 사용자 중심의 맞춤형 AI 복약관리 환경을 구현하여 현행 문제들을 해결하고자 한다. 본 개발의 구체적인 목표는 다음과 같다.

- 첫째, 현대인의 높아진 건강 인식 수준과 급격한 고령화 및 만성질환 증가 추세, 개인정보 보호 요구에 부응하는 맞춤형 복약관리 서비스를 구현한다.
- 둘째, 개인의 복약 정보를 맞춤형으로 통합 관리하고 정보에 대한 주도권을 사용자가 확보함으로써 복약 순응도를 높이고 약물 사고 및 개인정보 유출을 예방한다.
- 셋째, 종이 처방전과 약품 패키지 등 아날로그 매체의 한계를 디지털 전환을 통해 극복하여 정보 전달 효율을 높인다.
- 넷째, AI를 활용한 복약 안내 및 관리의 자동화·개인화를 실현하여 사용자에게 정확한 맞춤형 정보를 제공하고, 이를 통해 디지털 헬스케어 혁신을 추구한다.

---

Pilltip을 통해 사용자들은 본인의 복용 정보를 한 곳에서 안전하게 관리하고, 개인에 최적화된 복용 가이드 및 알림 서비스를 받음으로써 안전하고 능동적인 복용 관리가 가능해질 것이다. 본 프로젝트는 앞서 제시한 문제들을 해결함과 동시에 개인맞춤형 AI 복용 관리 서비스를 제공하여 개인과 사회의 복용 안전성 향상, 개인정보 보호, AI를 활용한 디지털 대전환이라는 사회적 가치를 실현하는 것을 목표로 한다.

---

## 2. 연구 배경

### 2.1 Elasticsearch

Elasticsearch는 Apache Lucene을 기반으로 구축된 분산형 검색 및 분석 엔진으로, 대용량 데이터 저장소이자 벡터 데이터베이스로 활용되는 오픈 소스 시스템이다. JSON 형태의 문서를 색인하고, 역색인 구조를 통해 대용량 텍스트 데이터에 대한 실시간 검색을 지원하며, 분산 아키텍처를 통해 데이터와 쿼리를 여러 노드로 분산 처리함으로써 수평 확장성과 고가용성을 달성한다. 특히 Apache Lucene 기반의 역색인을 활용하여 풀텍스트 검색 시에도 밀리초 단위의 응답 속도를 제공하고, 클러스터 내 샤드와 복제본을 통해 장애 내성을 보장한다. 이러한 특성 덕분에 Elasticsearch는 로그 분석, 정보 검색, 보안 모니터링 등 대규모 데이터의 실시간 검색 및 분석에 널리 사용된다. 본 연구에서는 방대한 사용자 정보 및 약품 DB를 쿼리하기에, Elasticsearch로 검색 및 자동완성을 보다 빠른 속도로 지원함으로써 UX를 개선하였다.

### 2.2 Redis

Redis는 메모리 기반의 키-값 구조 데이터베이스로서, 고속의 읽기/쓰기를 제공하는 NoSQL 시스템이다. 일반적인 데이터는 메모리에 저장하고, 필요에 따라 디스크에 스냅샷이나 명령 로그 형태로 영구화할 수 있다. 문자열, 해시, 리스트, 셋, 정렬된 셋 등 다양한 자료 구조를 지원하여 캐싱, 세션 관리, 실시간 통계 등의 용도로 활용된다. 단일 스레드 이벤트 루프 기반으로 동작하며, 비동기 I/O와 I/O 다중화를 사용하여 초당 수백만 건의 요청을 처리할 수 있는 고성능을 지닌다. Redis는 마스터-슬레이브 복제 및 Sentinel을 통한 고가용성 구성, Cluster 모드를 통한 샤딩으로 수평 확장도 지원한다. 본 연구에서는 개인 맞춤형 DUR(Drug Utilization Review, 의약품안전사용서비스) 구현 시, MySQL에 저장된 DUR 정보를 쿼리를 통해 단순 호출하는 것이 아닌, 캐시 메모리에 가까운 Redis에 Key-Value 형식으로 정보를 저장하여 제품 검색과 동시에 태깅이 가능하도록 속도 향상을 꾀해 UX를 개선하고자 사용하였다.

### 2.3 Weaviate

Weaviate는 AI 응용을 위한 오픈 소스 벡터 데이터베이스로, 객체 데이터와 해당 벡터 임베딩을 함께 저장·관리함으로써 벡터 유사성 검색을 지원하는 시스템이다. 즉, 텍스트나 이미지 등을 기계 학습모델로 임베딩하여 얻은 고차원 벡터를 저장해 두고, 새로운 쿼리에 대한 유사한 벡터를 효율적으로 찾아낸다. HNSW와 같은 근사 최근접 탐색 알고리즘을 사용하여 수백-수천 차원의 벡터 공간에서도 실시간에 가까운 유사도 검색을 구현한다. Weaviate는 하

---

이브리드 검색(벡터 유사도 + 키워드 필터)과 스키마 기반 필터링도 지원하여, 백과사전식 지식 그래프나 문서 QA 등 의미론적 검색 시나리오에 적합하다. 본 연구에서는 RAG를 위한 벡터 DB로써, 하이브리드 검색을 통한 정확도 상승 및 자동 벡터화 기능을 활용하여 AI 챗봇의 정확한 정보 전달을 위해 사용하였다.

## 2.4 FCM(Firebase Cloud Messaging)

FCM(Firebase Cloud Messaging)은 Google Firebase에서 제공하는 크로스 플랫폼 푸시 메시징 서비스로, 안드로이드, iOS, 웹 등 다양한 클라이언트 앱에 알림 또는 데이터 메시지를 안정적으로 송신한다. 앱 서버에서 FCM 서버 API 호출 시 Google의 인프라가 특정 디바이스 토큰이나 토픽을 대상으로 모바일 기기의 FCM 클라이언트 SDK까지 푸시를 중개한다. 이를 통해 실시간 채팅, 공지, 상태 업데이트 등 다양한 시나리오에서 사용자 기기로 즉시 알림을 전달할 수 있다. FCM은 무상으로 무제한 메시지 전송을 지원하며, 알림 메시지와 데이터 메시지 두 가지 유형을 제공한다. 본 연구에서는 복약 알림 기능을 구현하고자, 안드로이드 플랫폼을 지원하는 오픈 소스인 FCM을 채택하였다.

## 2.5 AES-GCM

AES-GCM은 대칭키 블록 암호인 AES를 Galois/Counter 모드(GCM)로 동작시켜 데이터의 기밀성과 무결성을 동시에 보장하는 인증 암호 방식이다. 이 방식은 평문을 암호화하는 동시에 인증 태그를 생성하여, 전송 중 데이터가 위변조되지 않았음을 검증할 수 있게 한다. 또한 Initialization Vector(IV) 값을 사용해 같은 키를 사용하더라도 매번 다른 암호문이 생성되도록 하여 보안을 강화한다. GCM은 병렬 처리에 최적화되어 있어 높은 성능을 제공하며, TLS 1.3 등 다양한 보안 프로토콜에서 표준으로 채택되어 널리 활용된다. 본 연구에서는 사용자의 민감한 개인정보(예: 이름, 연락처, 건강 정보)를 안전하게 보호하는 것이 중요하다고 판단했다. 만약 데이터베이스가 유출되더라도 제 3자가 정보를 확인할 수 없도록, 모든 개인정보를 암호화하는 방법을 채택했으며, 이를 위해 기밀성과 무결성을 동시에 보장하는 AES-GCM 알고리즘을 사용했다. 특히 IV를 매번 다르게 생성하여 동일한 정보가 암호화되더라도 다른 암호문이 생성되게 함으로써 추측을 통한 해킹 역시 방지했다.

## 2.6 TLS

TLS(Transport Layer Security)는 인터넷 통신을 암호화하여 기밀성, 무결성, 인증을 보장하는 암호화 프로토콜이다. 웹의 HTTPS가 가장 대표적인 적용 사

---

레로, 클라이언트와 서버 간의 handshake 과정을 통해 통신에 사용할 대칭키를 협상한다. 이후 모든 데이터는 이 대칭키를 이용해 암호화되므로 제 3자가 통신 내용을 도청하거나 위변조하는 것을 막을 수 있다. TLS는 강력한 보안과 폭넓은 표준 채택으로 인해 오늘날 인터넷 통신 보안의 필수적인 기반 기술로 자리 잡았다. 본 연구에서는 모바일 앱 및 웹과 서버 간의 메시지 전송 과정에서 사용자의 민감한 정보가 송수신되는 점을 고려했다. 사용자 정보를 안전하게 보호하고자 공인된 기관에서 TLS/SSL 인증서를 발급받아 서버에 적용하고, 모바일 앱 및 웹과 서버 간의 모든 통신에 TLS를 적용하여 보안을 강화했다.

## 2.7 Nginx

Nginx는 이벤트 기반 아키텍처를 채택한 고성능 웹 서버이자 리버스 프록시 서버이다. 요청마다 프로세스를 생성하는 기존 방식과 달리, 비동기식 단일 쓰레드 이벤트 루프로 수많은 동시 요청을 효율적으로 처리한다. 이러한 특징으로, 적은 자원으로도 뛰어난 성능을 발휘하며, 정적 파일 서빙, 로드 밸런싱, 캐싱, TLS 암호화 등 풍부한 기능을 제공한다. Nginx는 특히 리버스 프록시로서 백엔드 애플리케이션 서버 앞단에서 요청을 분배하고 관리하는데 탁월해, 고성능이 요구되는 웹 서비스에 널리 사용되고 있다. 본 연구에서는 암호화된 통신 환경을 구축하기 위해 도메인에 TLS/SSL 인증서를 적용할 필요가 있었다. 또한, 사용자 요청을 안정적으로 서버에 전달하고 부하를 분산하기 위해 리버스 프록시가 필수적이라고 판단했다. 따라서 상기 역할을 효율적으로 수행하며, 높은 성능을 제공하는 Nginx를 사용했다.

## 2.8 React

React는 재사용 가능한 UI 컴포넌트를 활용하여 복잡한 사용자 인터페이스를 구축하는 데 사용되는 자바스크립트 라이브러리다. 핵심 개념은 선언적 프로그래밍으로, 애플리케이션 상태에 따라 UI 기술 시 React가 변경된 부분만 효율적으로 업데이트한다. 이를 위해 Virtual DOM을 사용해 실제 DOM 조작을 최소화하고 성능을 높인다. 이러한 컴포넌트 기반 접근 방식으로 React는 대규모 SPA(Single Page Application) 개발의 표준 도구로 자리 잡았다. 본 연구에서는 환자가 QR코드로 제출한 문진표를 실시간으로 웹에서 열람 가능하도록 React를 활용했다. 특히, 제출 시점으로부터 1시간의 열람 제한 시간을 두어야 했고, 이를 프론트엔드에서 구현하고자 JavaScript 기반인 React를 선택했다.

## 2.9 Next.js

Next.js는 React 기반 풀스택 웹 프레임워크로, 서버 사이드 렌더링과 정적

---

사이트 생성을 지원하여 React 애플리케이션의 성능과 SEO(검색 엔진 최적화)를 향상시키는 것을 목표로 한다. 파일 시스템 기반 라우팅, API 라우트, 자동 번들링 등 개발 편의 기능을 내장하여 효율적인 개발을 돕는다. Next.js는 페이지를 서버에서 미리 렌더링하여 초기 HTML을 완성된 형태로 전달하고, 이후 클라이언트에서 인터랙티브를 구성하는 방식으로 동작한다. 본 연구에서는 페이지 로딩 성능과 코드 안정성 향상을 목표로, React 기반 프로젝트 전반에 Next.js와 TypeScript를 적용했다. 특히, Next.js의 서버 사이드 렌더링 기능을 활용하여 페이지를 미리 렌더링함으로써 초기 로딩 속도를 최적화했다.

## 2.10 RAG

RAG(Retrieval-Augmented Generation)는 대규모 언어모델 생성 과정에 외부 지식원(문서 데이터베이스, 벡터 데이터베이스 등)에서 가져온 근거 문서를 동적으로 주입하여 사실성과 신뢰성을 높이는 아키텍처다. RAG는 크게 두 단계로 구성된다.

- 첫째, 사용자의 질의가 들어오면 검색기가 미리 구축된 외부 지식 기반에서 가장 관련성 높은 문서를 찾아낸다.
- 둘째, 생성기는 사용자의 원래 질의와 검색된 문서를 함께 참고하여 응답을 생성한다.

상기 방식으로, 모델 파라미터에 내장되지 않은 최신 또는 특정 도메인의 지식을 실시간 반영할 수 있으며, 응답의 근거를 함께 제시함으로써 투명성 및 추적성을 확보할 수 있다. 본 연구에서는 AI 챗봇의 할루시네이션 현상을 방지하고, 사용자에게 정확하고, 객관적인 정보만을 제공하는 것을 핵심 목표로 설정했다. 이를 위해 RAG를 도입하여, 챗봇이 생성한 답변의 근거를 외부 데이터베이스에서 찾아내도록 함으로써, 신뢰할 수 있는 정보를 기반으로 응답을 생성하도록 설계했다.

## 2.11 프롬프트 엔지니어링

프롬프트 엔지니어링은 대규모 언어모델이 개발자의 의도에 부합하는 결과를 생성하도록 입력 프롬프트를 정교하게 설계하고 최적화하는 방법론이다. 이는 모델 자체를 재학습(파인튜닝)하지 않고도 모델의 추론 능력과 안정성을 향상시키는 핵심 기술이다. 프롬프트는 단순히 질문을 던지는 것을 넘어, 모델에게 역할을 부여하거나, 단계별 사고 과정을 유도하고, 원하는 출력 포맷을 명시하는 등 다양한 지시를 포함한다.

예를 들어, Few-shot 프롬프팅은 몇 가지 예시를 제공하여 모델이 원하는 작업의 패턴을 학습하도록 돕고, CoT는 복잡한 문제 해결 과정을 중간 단계로 나누어 모델이 논리적으로 추론하게 한다. 이러한 기법들은 모델의 '컨텍스트

---

트 내 학습' 능력을 극대화하여, 복잡한 질의에도 일관되고 정확한 응답을 도출하게 한다. 프롬프트 엔지니어링은 모델의 '할루시네이션'을 줄이고, 특정 도메인에 대한 지식 활용을 개선하며, 보안 및 규제 준수 등 요구사항을 충족시키는 데 중요한 역할을 한다. 본 연구에서는 약품 설명 구어체 변환 및 맞춤 안내, 약품 상충 비교 등 사용자에게 정확하고 이해하기 쉬운 정보를 제공하는 것을 목표로 했다. AI 모델이 딱딱한 의학 정보를 직접 구어체로 '창작'하는 대신, 기존 정보를 활용하여 자연스러운 문장을 '재구성'하는 데 집중하도록 프롬프트 엔지니어링을 활용했다. 예를 들어, "사용자가 쉽게 이해할 수 있도록 구어체로 설명해 줘. 경어체를 사용하고, 어려운 의학 용어는 풀어 써줘."와 같은 구체적인 지시를 프롬프트에 포함시켰다. 이를 통해 문법적으로 어색하거나 부정확한 문장이 생성되는 것을 방지하고, 사용자가 필요한 정보를 쉽고 간편하게 습득할 수 있도록 AI의 응답 출력 품질을 극대화했다.

상기 접근법은 모델의 성능을 유의미하게 향상시키면서도, 추가적인 학습이나 비용 발생 없이 개발의 민첩성을 확보할 수 있었다.

## 2.12 Orchestration AI

Orchestration AI는 단일 LLM 호출을 넘어, 복합적인 작업을 안정적으로 실행하도록 돕는 설계 및 운영 계층이며, 프롬프트, 톨 호출, 메모리 관리, 로깅 등 다양한 요소를 통합하여 하나의 워크플로우로 연결한다. LangChain과 같은 프레임워크를 활용하면 LLM과 외부 API, 데이터베이스, 다른 에이전트 간의 작업을 선언적으로 구성하고 모니터링할 수 있다. 나아가 멀티 에이전트 시스템(MAS)을 통해 복잡한 문제를 여러 에이전트가 역할을 나누어 협력하고 해결하도록 지원한다. 본 연구에서는 AI 챗봇이 사용자에게 보다 정확한 정보를 제공하도록 RAG, DUR Tool 등을 적재적소에 사용해야 했다. 이를 위해 오케스트레이션 AI를 활용하여, 사용자의 요청에 따라 다양한 톨이 유기적으로 연동되고, 최적의 순서로 실행되도록 파이프라인을 구축했다.

## 2.13 YOLOv8

YOLOv8은 Ultralytics가 공개한 최신 실시간 객체 탐지 모델로, 단일 프레임워크 내에서 객체 탐지뿐만 아니라 인스턴스 분할, 자세 추정, 분류 등 다양한 작업을 수행하도록 설계된 멀티태스크 패밀리 모델이다. 이 모델의 핵심적인 아키텍처 개선점은 앵커 프리 방식과 디커플드 헤드를 채택했다는 점이다. 이를 통해 기존 모델에서 필요했던 복잡한 앵커 박스 설계 및 튜닝 과정을 없애고, 분류와 회귀 태스크를 분리하여 모델의 정확도와 추론 안정성을 동시에 높였다. 또한, C2f 모듈을 도입해 특징 맵의 재사용을 최적화하고

---

다중 스케일 피쳐 융합을 강화함으로써, 이전 세대인 YOLOv5에 비해 더 나은 성능-속도 균형을 달성했다. 본 연구에서는 알약 이미지를 활용한 검색 기능 구현 시, 이미지 속 알약 형상을 정확하게 식별하는 것이 중요하다고 판단했다. 특히, 알약의 색상, 색상, 식별 문자 등 다양한 특징을 정밀하게 탐지해야 했기에, 높은 정확도와 빠른 추론 속도를 모두 제공하는 YOLOv8 모델을 채택했다. 이를 통해 수많은 알약 이미지 데이터에 대해 모델을 학습시켜 인식률과 정확도를 극대화하고, 사용자가 사진을 찍어 알약 검색 시 실시간으로 결과를 제공하는 기반을 마련했다.

#### 2.14 Jetpack Compose

Jetpack Compose는 구글이 공식 제공하는 현대적 선언형 UI 킷으로, 네이티브 안드로이드 UI를 효율적으로 구축하도록 설계되었다. 기존의 XML 레이아웃 기반 명령형 UI와 달리, Compose는 선언형 프로그래밍 패러다임을 채택하여 개발자가 UI의 최종 상태 기술 시 프레임워크가 변경 사항을 자동으로 처리한다. 즉, UI를 함수 기반의 Composable 단위로 구축하고, 상태를 한 곳에서 관리하며 상태 변화에 따라 UI를 recomposition함으로써, 코드 양을 줄이고 UI 갱신에 따르는 오류를 감소시킨다. 이러한 핵심 개념을 통해 Compose는 UI 코드의 가독성과 유지보수성을 크게 향상시키며, 이러한 필요성으로, 많은 안드로이드 프로젝트가 기존 View 시스템에서 Compose로 전환하거나 병행 사용하고 있다. 본 연구에서는 선언형 구문과 Kotlin 기반 DSL을 통해 보일러플레이트 코드 및 반복 작업이 감소하고 UI 업데이트를 자동화하는 등의 이점이 있는 최신 방식인 Jetpack Compose 선언형 UI 킷을 UI 작성에 활용함으로써 UX 개선 및 애플리케이션 개발의 효율화를 추구하였다.

#### 2.15 Lifecycle - LiveData

Lifecycle은 안드로이드 컴포넌트의 생명주기 변화를 관찰하고 대응할 수 있도록 하는 아키텍처 구성요소이다. 그리고 LiveData는 Jetpack의 Lifecycle 라이브러리에 포함된 관찰 가능한 데이터 홀더 클래스로서, 일반 옵저버블과 달리 Lifecycle 인식 기능을 갖추고 있다. LiveData는 연결된 LifecycleOwner의 상태를 고려하여 STARTED 또는 RESUMED 상태의 옵저버에게만 데이터를 통지한다. 즉, LiveData에 등록된 옵저버는 해당 UI 컴포넌트가 화면에 보이는 동안에만 업데이트를 받고, 비활성 상태나 소멸 시에는 자동으로 통지가 중지된다. 이러한 특성 덕분에 LiveData를 사용하면 메모리 누수나 중지된 UI에 대한 잘못된 업데이트를 걱정할 필요 없이 안전하게 데이터 변화를 전달할 수 있다. LiveData/Lifecycle은 안드로이드 아키텍처 컴포넌트의 핵심 요소로서 널리 사용되며, 본 연구에서는 간결하고 안전한 UI 업데이트 메커

---

니즘이 필요한 상용 앱을 개발하고자 적극적으로 활용하였다.

## 2.16 ViewModel

ViewModel은 안드로이드 아키텍처 컴포넌트 중 하나로, 화면에 필요한 상태 데이터와 비즈니스 로직을 담는 객체이다. ViewModel 클래스는 UI 컨트롤러와 별개로 존재하여, 화면 회전 등의 구성 변경 시에도 데이터를 유지하고 계속 사용할 수 있게 설계되었다. 핵심 개념은 Lifecycle-aware라는 점으로, 특정 Activity나 Fragment에 ViewModel을 연결해 두면 해당 UI가 파괴되기 전까지 ViewModel 인스턴스가 메모리에 유지되어 데이터를 캐시한다. 이러한 특성 덕분에 ViewModel은 화면 회전 시 매번 네트워크나 DB로부터 데이터를 다시 불러오는 것을 방지하고, 복잡한 상태 저장 로직 없이도 UI 상태를 복원할 수 있게 한다. 또한 ViewModel 내부에는 UI에 표시할 데이터와 그 데이터를 처리하는 로직만을 포함하고, UI 요소에 대한 참조를 두지 않으므로써 UI와의 분리를 실현한다. 이로써 Activity나 Fragment는 View만 담당하기에 가벼워지고, ViewModel이 비즈니스 로직과 데이터 변환 작업 등을 캡슐화하여 앱 구조가 체계화된다. 특히 ViewModel은 LiveData나 Kotlin Flow와 결합하여 관찰 가능한 데이터 스트림을 노출하고, UI는 이를 관찰하도록 함으로써 MVVM 패턴의 구현을 돕는다. 본 연구에서는 견고하고 확장 가능한 안드로이드 애플리케이션을 구현하고자 Hilt와 함께 ViewModel을 활용하였다.

## 2.17 MVVM

MVVM(Model-View-ViewModel)은 소프트웨어 공학의 아키텍처 패턴으로, UI(View)와 비즈니스-데이터(Model) 사이에 ViewModel을 두어 분리를 극대화한 구조이다. Android 개발에서 MVVM 패턴은 Google이 권장하는 공식 아키텍처이며, Jetpack 아키텍처 컴포넌트를 활용하여 손쉽게 구현할 수 있다. 핵심 개념은 단방향 데이터 흐름으로, ViewModel이 노출하는 관찰 가능한 UI 상태(LiveData/Flow 등)를 View가 관찰하여 반영하고, View에서 발생하는 사용자 이벤트를 ViewModel에 전달함으로써 상호작용이 이뤄진다. 이때 ViewModel은 View에 대한 참조나 의존성이 없으며, View는 ViewModel을 참조하여 데이터만 받아 표현한다. 이러한 구조 덕분에 View와 Model 간의 결합도가 낮아져 유지보수성과 테스트 용이성이 크게 향상된다. 본 연구에서는 앱 개발 효율을 높이고, 사용자에게 더 안정적이고 응답성 좋은 앱 경험을 선사하고자 해당 패턴을 적용하였다.

## 2.18 Coroutine

코루틴은 Kotlin 언어에서 제공하는 경량화된 비동기 프로그래밍 프레임워크

---

로, 스레드 기반의 전통적인 방식보다 간결하고 효율적으로 동시 작업을 다룰 수 있도록 설계되었다. 코루틴은 하나의 디자인 패턴으로서, 함수 실행을 일시 Suspend하고 나중에 Resume할 수 있는 기능을 제공하여 논리적으로는 동기 코드 형태이지만 실제로는 비동기로 동작하도록 한다. 안드로이드에서 코루틴은 긴 작업을 메인 스레드를 블로킹하지 않고도 수행할 수 있게 해주며, 콜백 지옥을 피하고 예외 처리를 통합하는 등 코드의 가독성과 안정성을 높여준다. 본 연구에서는 복잡한 스레드 핸들링을 코루틴으로 편리하게 수행함으로써 코드 품질과 안정성을 높이고, 부드러운 UI 경험을 사용자에게 제공하고자 활용하였다.

### 2.19 Hilt-Dagger (Dependency Injection)

Dagger는 자바 및 안드로이드에서 쓰이는 대표적인 의존성 주입 프레임워크로, 컴파일 시점에 의존성 그래프를 분석하여 필요한 코드를 생성해주는 라이브러리이다. Hilt는 이 Dagger를 기반으로 구글이 안드로이드 앱 개발에 최적화하여 제공하는 DI 라이브러리로, Dagger 사용 시의 보일러플레이트 코드를 크게 줄여준다. DI의 핵심 개념은 객체 생성과 사용의 분리로, 클래스가 직접 그 의존 객체를 생성하지 않고 외부에서 주입받음으로써 모듈간 결합도를 낮추고 테스트 가능성을 높인다. Dagger는 @Inject, @Module, @Component 등의 애노테이션으로 이 작업을 자동화하며, Hilt는 더 나아가 안드로이드 구성요소 별로 DI 컨테이너를 제공하여 안드로이드 개발자가 DI를 쉽게 적용할 수 있도록 한 것이 특징이다. 예를 들어 Hilt를 사용하면 Application 클래스에 @HiltAndroidApp만 붙이면 애플리케이션 범위의 DI 컨테이너가 자동 생성되고, Activity에는 @AndroidEntryPoint만 붙여도 필요한 의존성들이 주입되어 사용 가능해진다. 본 연구에서는 의존성 주입을 통해 안드로이드 앱의 구조적 품질과 유지보수성을 높이고, DI 구현을 크게 단순화하면서도 컴파일 타임 안전성과 성능을 담보하기 위해 Hilt와 Dagger를 활용하였다.

### 2.20 OkHttp

OkHttp는 Square사에서 개발한 오픈소스 HTTP 클라이언트 라이브러리로, 안드로이드 및 자바 애플리케이션에서 신뢰성 있고 효율적인 네트워크 통신을 수행하기 위해 널리 쓰인다. OkHttp의 핵심은 HTTP 요청/응답 처리의 최적화에 있다. 이 라이브러리는 HTTP/1.1과 HTTP/2를 모두 지원하며, 하나의 연결로 다수의 요청을 효율적으로 주고받기 위해 커넥션 풀링과 응답 캐싱 등의 기능을 제공한다. 예를 들어 동일 호스트에 여러 요청을 보내면 HTTP/2 지원을 통해 단일 TCP 연결을 공유하거나, HTTP/2가 불가능한 경우에도 Keep-Alive 연결을 재사용하여 지연을 줄인다. 또한 네트워크 장애 대

---

응을 위해 내장된 재시도 전략이 있으며, TLS 설정도 최신 표준을 지향해 보안에 충실하다. OkHttp는 이러한 기능을 대부분 개발자로부터 감추고 자동으로 동작하게 하면서도, 동시에 간편한 API(fluent interface의 Request Builder 등)를 제공해 개발 편의성을 높인다. 본 연구에서는 안정적인 통신 유지 및 통신 간 암호화를 위해 OkHttp를 통신 전 과정에서 활용하였다.

### 3. 연구 내용

#### 3.1 의약품·건강기능식품 공공데이터 가공

##### 3.1.1 의약품 공공데이터 파서

의약품 안전 사용 서비스(DUR)는 일반 대중의 접근성이 낮아 실질적인 복약 안전 관리에 한계가 존재한다. 이에 따라 필팁은 공신력 있는 공공데이터를 기반으로 사용자의 건강정보와 약품 간 상충작용을 비교·판별하는 Pilltip만의 개인맞춤형 DUR 시스템을 구축하여, 일반적인 사용자가 정밀한 DUR 시스템을 보다 쉽게 활용하고, 맞춤형 서비스를 누릴 수 있도록 하였다.

이를 위한 데이터 수집은 식품의약품안전처의 '의약품 제품 허가 정보'와 '의약품 개요 정보'를 활용하여 약물별 제품명, 제조사, 성분함량, 효능, 용법, 주의사항 등 핵심 정보를 통합 확보하는 방식으로 진행하였다. 해당 공공데이터는 복약 설명 자동화를 위한 기반 자료로 적합하다고 판단하였으며, 핵심 정보를 일괄적으로 포함한다는 점에서 활용 가치를 갖는다. 약물 간 상호작용을 정밀하게 분석하기 위해 'DUR 품목정보'와 'DUR 유형별 성분 정보'를 모두 사용하였다. 이를 통해 기존에 품목 단위만으로는 탐지하기 어려웠던 특정 성분 단위의 금기 및 병용 주의사항까지 반영한 DUR 데이터를 구성하였다.

한편, 보건복지부와 식품의약품안전처 등 국가기관이 제공한 공공데이터라 하더라도 실제 분석·가공 과정에서는 불규칙한 줄바꿈, 비표준 성분명, 누락 항목 등 구조적 비일관성이 다수 확인되었다. 이러한 문제를 해소하지 않을 경우 데이터베이스 구축 시 불필요한 공백, 매핑 오류, 계산 착오가 발생할 가능성이 높다고 판단하여, 본격적인 시스템 설계에 앞서 데이터 정제와 표준화를 우선 수행하였다.

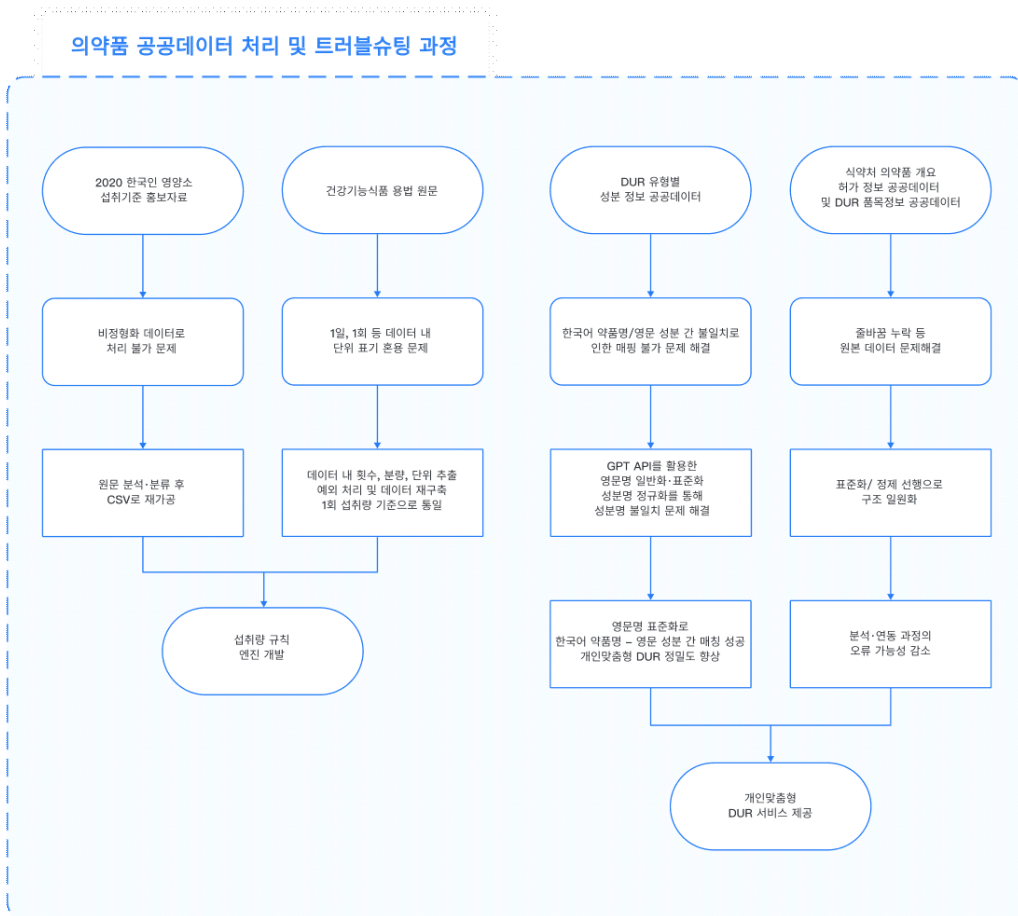


그림 8 <개인맞춤형 DUR 구축을 위한 의약품 및 성분 공공데이터 파싱 과정>

전처리 단계에서는 불필요한 공백과 특수기호를 제거하고, 약 이름에 주요 성분명이나 수출명이 추가되거나 불필요한 개행으로 명칭이 훼손된 사례를 교정하였다. 특히 괄호가 완결된 구간을 검출하여 해당 괄호 내부 내용을 제거하는 규칙을 적용함으로써 제품명 표기를 일관화하고, 명칭 기반 매핑의 신뢰성을 확보하였다.

의약품 성분 데이터에서는 숫자와 단위의 반복 표기, 숫자·단위 표기 방식의 비일관성, '이상', '역가'와 같은 특수 단위 존재 등이 확인되었다. 이에 따라 정규표현식 패턴 매칭으로 숫자와 단위를 분리 추출한 후, 동일 숫자가 중복되는 경우 하나를 삭제하여 중복 표기를 제거하였다. 숫자와 단위가 비일관적으로 작성된 사례는 숫자와 단위를 우선 분리한 뒤 각각에 대한 정제 규칙을 적용하여 형식을 일관화하였다. 특수 단위는 산출 과정에서 필요하지 않아 계산 단계에서는 제거하여 처리 복잡성을 낮추었다.

의약품 DUR 데이터는 품목별 DUR 데이터와 성분별 DUR 데이터의 두

범주로 구성되어 있다. 이 중 품목별 DUR 정보는 기록 자체의 일관성은 확보되어 있었으나, 기존에 저장한 의약품명과 DUR에 기재된 의약품명의 표기 규칙이 상이하게 나타났다. 두 명칭이 동일 품목을 지칭한다는 전제하에, 명칭 길이를 기준으로 한 포함 관계 매칭 규칙을 적용하였다. 구체적으로, 더 긴 명칭이 더 짧은 명칭을 포함하는 경우 동일 품목으로 간주하여 추가 매핑하고, 포함되지 않는 경우에는 동일하지 않은 것으로 판단하였다.

성분 DUR 데이터는 영문 표기 기준으로 제공되어 한글 기준 성분명과 직접 매핑이 어려웠다. 이에 GPT를 활용하여 영문 성분명을 일반화·표준화하고, 표준 한글 성분명과 자동 매핑을 구현함으로써 개인맞춤형 성분 DUR 데이터의 연계를 가능하게 하였다.

	A	B	C	D	E	F	G
1	성분명_한	성분명_영	정규화_영	DUR_매핑	일반명_영	DUR_정규화_매핑	
2	에페드린	ephedrine	ephedrine	FALSE	ephedrine	TRUE	
3	아미노필린	Aminophyll	aminophylli	FALSE	aminophylli	TRUE	
4	황산마그네	Magnesium	magnesium	FALSE	magnesium	TRUE	
5	테트라사이	Tetracyclin	tetracycline	FALSE	tetracycline	TRUE	
6	스트렙토미	Streptomyc	streptomyc	FALSE	streptomyc	TRUE	
7	이소니아지	Isoniazid	isoniazid	TRUE	isoniazid	TRUE	
8	옥시토신	oxytocin	oxytocin	TRUE	oxytocin	TRUE	
9	덱사메타손	dexametha	dexametha	TRUE	dexametha	TRUE	

그림 9 <영문 성분 DUR 데이터를 표준 한글 성분명으로 정규화·자동 매핑 결과 중 일부>

상기 과정을 통해 동일 성분임에도 표기 방식 차이로 발생하던 매핑 누락 문제를 해소하였다. 성분명 표기를 일관화함으로써 성분 기준 개인맞춤 DUR 적용의 정확도를 크게 향상시켰다.

### 3.1.2 구어체 변환을 위한 의약품 데이터 블럭화

각 의약품의 용법과 주의사항을 사용자 친화적인 구어체로 변환하기 위해 프롬프트 기반 GPT API를 적용하였다. 다만 전체 약물 약 44,000건을 그대로 변환할 경우 API 비용이 약 100~200달러로 추정되어, 비용 절감이 필요했다. 이를 위해 데이터를 면밀히 분석한 결과, 용법·주의사항 문장에 중복이 다수 존재함을 확인하였고, 문장 단위 추출 및 패턴별 재사용 전략을 도입하였다. 구체적으로 용법과 주의사항을 한 문장씩 분리 추출한 뒤 정규표현식을 이용해 대분류 번호를 제거하고, 정제된 문장을 각각의 리스트로 관리하였다. 이후 다른 약품 처리 시 동일 문장이 리스트에 존재하

면 해당 문장을 블록으로 간주하고, 블록 ID를 약품 ID와 매핑하여 재사용하도록 구성하였다. 이와 같은 블록화 절차를 통해 최종적으로 프롬프트 변환 대상은 용법 9,333문장, 주의사항 18,193문장으로 축소되었으며, 이 수량만으로 전체 용법·주의사항을 전부 커버할 수 있도록 하였다.

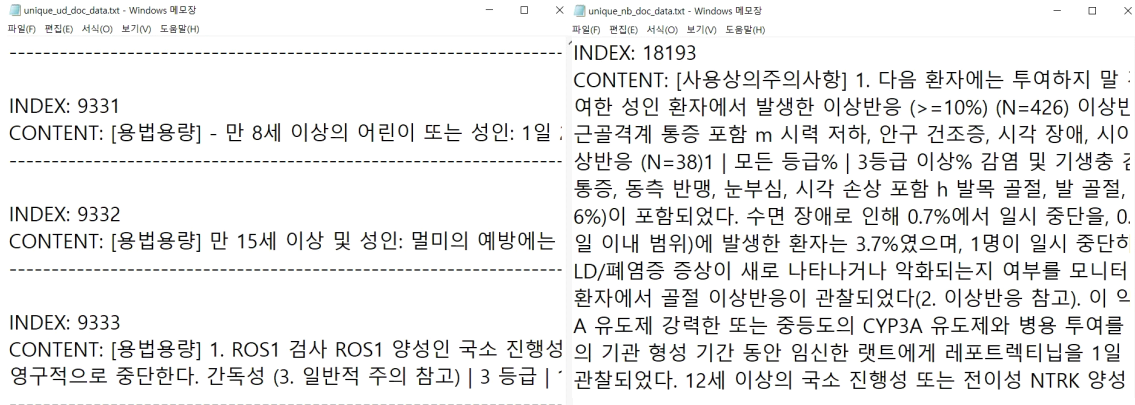


그림 6-7 <블럭화한 의약품 복용법 및 주의사항 중 일부>



그림 8-9 <블럭 매핑한 의약품 복용법 및 주의사항 중 일부>

### 3.1.3 Batch API

앞서 수행한 블록화를 통해 추출된 고유 문장만을 대상으로 프롬프팅하기 위해 OpenAI의 Batch API를 적용하였다. Batch API는 대량 요청을 일괄 제출할 수 있으며, 기존 실시간 API 대비 입·출력 토큰 단가가 절반 수준으로 절감된다. 결과물은 JSONL 형식으로 제공되고, 후처리 코드를 통해 TXT 등 내부 포맷으로 변환하여 활용하였다. 최대 24시간의 처리 지연이 발생할 수 있으나, 본 연구는 대규모 토큰 사용에 따른 비용 절감 효과를 우선시하여 Batch API를 선택하였다.

```

# Batch API 요청 형식
request = {
    "custom_id": f"{prompt_type}batch{batch_id}",
    "method": "POST",
    "url": "/v1/chat/completions",
    "body": {
        "model": "gpt-4o-mini",
        "temperature": 0.1,
        "max_tokens": max_tokens,
        "messages": [
            {"role": "system", "content": system_content},
            {"role": "user", "content": batch_prompt}
        ]
    }
}
batch_requests.append(request)

return batch_requests

def create_batch_file(self, requests: List[Dict], filename: str) -> str:
    """Batch API용 JSONL 파일 생성"""
    os.makedirs("batch_files", exist_ok=True)
    filepath = f"batch_files/{filename}"

    with open(filepath, 'w', encoding='utf-8') as f:
        for request in requests:
            f.write(json.dumps(request, ensure_ascii=False) + '\n')

    print(f"[배치 파일 생성: {filepath}]")
    return filepath

def upload_batch_file(self, filepath: str) -> str:
    """배치 파일을 OpenAI에 업로드"""
    print(f"[배치 파일 업로드 중: {filepath}]")

    with open(filepath, 'rb') as f:
        batch_file = client.files.create(
            file=f,
            purpose="batch"
        )

    print(f"[업로드 완료: {batch_file.id}]")
    return batch_file.id

```

그림10-11 <Batch API 코드 중 일부>

Batch API를 선택하여 활용한 결과, 변환 비용을 약 10~15달러 수준으로 절감하면서도 전체 데이터의 95% 이상을 효율적으로 구어체 변환할 수 있었다.

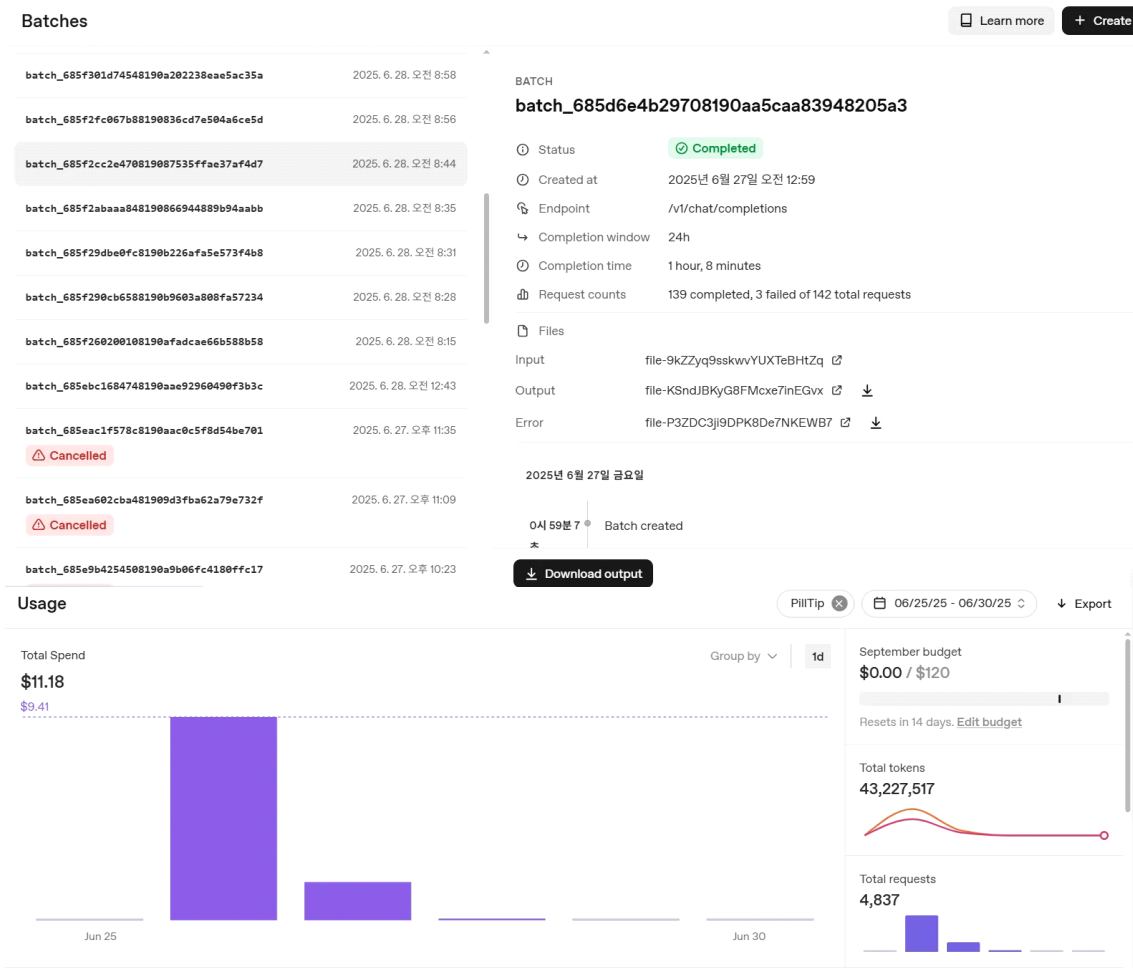


그림 17 <Batch api 결과 확인 및 청구 비용>

변환된 문장들은 후처리 코드를 통해 재조합하여, 제품별 정보를 모두 구어체로 일관되게 통합한 최종 데이터를 확보하였으며, 나머지 가공되지 않은 데이터는 동일 성분군의 주의사항을 활용해 보완하였다.

=== 중외5%포도당생리식염액(수출명:5%DextroseinnormalsalineInj.) 주의사항 ===

1. 경고

- 1) 포도당(당분) 함유 제제를 정맥주사(혈관에 직접 주사)하는 환자분들은 치아민(비타민 B1) 소모율이

2. 과량투여시의 처치

- 1) 과량으로 투여했을 경우 이뇨제(소변을 많이 배출하게 하는 약)를 투여해야 해요.

3. 다음 환자에는 신중히 투여할 것.

- 1) 신질환(신장에 문제가 있는 질환)으로 인한 신부전(신장이 제 기능을 하지 못하는 상태) 환자분께는
- 2) 심부전(심장이 제 기능을 하지 못하는 상태) 환자분께는 조심스럽게 이 약을 사용해야 해요.
- 3) 고장성 탈수증(체내 수분이 부족한 상태) 환자분께는 조심스럽게 이 약을 사용해야 해요.
- 4) 폐쇄성 요로질환(소변이 잘 나오지 않는 질환)으로 인한 요량감소(소변량이 줄어드는 상태) 환자분께
- 5) 당뇨병(혈당 조절이 어려운 질환) 환자분께는 조심스럽게 이 약을 사용해야 해요.
- 6) 고염소혈증(혈액 내 염소가 과다한 상태) 환자분께는 조심스럽게 이 약을 사용해야 해요.
- 7) 저나트륨혈증(혈액 내 나트륨이 부족한 상태, 120 mmol/L 미만) 환자분께는 조심스럽게 이 약을 사

4. 다음 환자에는 투여하지 말 것.

- 1) 수분과다상태(몸에 수분이 너무 많은 상태)인 환자분께는 이 약을 사용하지 말아주세요.
- 2) 저장성 탈수증(체내 수분이 부족한 상태) 환자분께는 이 약을 사용하지 말아주세요.
- 3) 저칼륨혈증(혈액 내 칼륨이 부족한 상태) 환자분께는 이 약을 사용하지 말아주세요.
- 4) 고나트륨혈증(혈액 내 나트륨이 과다한 상태) 환자분께는 이 약을 사용하지 말아주세요.

5. 보관 및 취급상의 주의사항

- 1) 실온에서 보관해 주세요.

6. 이상반응

- 1) 대량으로 급속하게 투여할 경우 뇌부종(뇌가 부풀어 오르는 상태), 폐부종(폐에 물이 차는 상태), 말초
- 2) 신생아(갓 태어난 아기)나 미숙아(조산으로 태어난 아기)에게 급속하게 투여할 경우 수중독(몸에 수분
- 3) 주입 정맥(주사한 혈관)에서 혈전증(혈관에 피가 굳어지는 상태)이 나타날 수 있어요.

7. 일반적 주의

- 1) 혈당(혈액 내 당의 양), 혈청 전해질(혈액 내 전해질의 양), 체액 평형(몸의 수분 균형)을 정기적으로

8. 임부, 수유부, 가임여성, 신생아, 유아, 소아, 고령자에 대한 투여

그림 18 <구어체 변환까지 완료된 최종 의약품 데이터 중 일부>

3.1.4 건강기능식품 공공데이터 파서

건강기능식품 정보의 경우, DUR 데이터가 주로 '~추출물' 성분 기준으로 제공되어 단순 제품 정보만으로는 성분 분석이 어려웠다. 이에 '건강기능식품 품목제조 신고사항 현황' 데이터에 추출물 정보가 포함되어 있음을 확인하고, DUR 연동을 위한 기반 데이터로 해당 자료를 채택하였다.

---

그러나 해당 데이터는 누락 항목, 불규칙한 작성 형식, 비표준 성분명 문제가 동시에 존재하였다.

STDR\_STND: 1.성상:이미 이취가없고 고유의 향미:  
2.비타민B1:표시량의(1.2mg/4,100mg)80~180%  
3.나이아신:표시량의(15mgNE/4,100mg)80~150%  
4.판토텐산:표시량의(2.5mg/4,100mg)80~180%  
5.비타민B6:표시량의(1.5mg/4,100mg)80~150%  
6.비타민B12:표시량의(2.4µg/4,100mg)80~180%  
7.비타민C:표시량의(100mg/4,100mg)80~150%  
8.아연:표시량의(8.0mg/4,100mg)80~150%  
9.대장균균 : 음성  
INDUTY\_CD\_NM: 건강기능식품전문제조업

그림 19 <성분명·성분함량·성분단위가 불규칙한 기존 공공데이터 중 일부>

이를 해소하기 위해 성분명·성분함량·성분단위를 정확히 분리 추출하는 정규표현식을 설계하였고, 특히 성분함량의 경우 다양한 표기 방식을 일관되게 인식하도록 7가지 패턴을 마련하였다. 동일 성분이 서로 다른 이름으로 기재되는 문제를 방지하기 위해 300여 개 성분명에 대한 통일·매핑 코드를 구현하여 데이터베이스의 일관성을 확보하였다.

성분 함량 표기에서도 문제가 확인되었다. 건강기능식품 정보 내 함량 값이 용법 기준과 일치하지 않게 '1일 섭취량'과 '1회 섭취량'이 혼용되어 작성된 사례가 존재하였다. 이는 동일 지표 간 비교 및 산출 과정에서 혼선을 야기할 수 있는 구조적 비일관성에 해당한다.

해당 문제를 해결하기 위해 우선 용법 원문에서 단위 표기 여부를 기준으로 데이터를 이분화하였다. 제시한 코드를 활용하여 단위가 명시된 사례와 명시되지 않은 사례를 자동 식별하고, 각각을 별도의 파일로 분리·저장하였다.

단위가 명시된 사례에서는 '~ml의 물을 함께 섭취'와 같은 표현이 용법의 중량 단위로 오인되어 예외가 발생하였다. 이를 방지하기 위해, 먼저 용법의 중량 단위가 기준량의 단위와 동일하거나 상호 변환 가능한 경우(예: mg↔g)로 분리하여 처리하였다. 이어서 용법의 중량 단위가 ml인 경우를 별도로 취급하여, 기준량 단위가 ml 이외의 다른 단위가 용법 내에 명시된 경우와 그렇지 않은 경우, 두 범주로 한 번 더 분기한 뒤 각각에 대해 계산을 수행하였다.

```

ratio_calculated_normal.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
item_id: 2
NTK_MTHD: 1일 3회, 1회 1스푼(3.2g)씩 물에 녹여 섭취 하
NTK_MTHD (정규화): 1일3회,1회1스푼(3.2g)씩물에녹여섭취
기준량: 9.6g
기준량 (정규화): 9.6g
용법 중량: 3.2g
1일 3회, 1회 1정
계산값: 3.2
비율: 3.2 : 9.6

item_id: 15
NTK_MTHD: 1일 1회, 1회 1정(1정 당 300 mg)을 물과 함께
NTK_MTHD (정규화): 1일1회,1회1정(1정당300mg)을물과함
기준량: 300 mg
기준량 (정규화): 300mg
용법 중량: 300.0mg

ml_exceptions_parsed.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
1일 1회, 1회 1정
ML 단위: [(200.0, 'ml')]
기타 단위: []
비율: 1 : 1
계산 방법: ML만 등장 - 1:횟수

item_id: 22
NTK_MTHD: 뚜껑을 개봉하여, 연질(500mg)+정제(600mg)를 9
기준량: 23600mg
기준량 (파싱): 23600.0mg
1일 1회, 1회 1정
ML 단위: [(20.0, 'ml')]
기타 단위: [(500.0, 'mg'), (600.0, 'mg')]
비율: 23600.0 : 23600.0
계산 방법: ML 제외 중량 합산

```

그림 20 <섭취량 비율 계산 과정 중 일부>

```

merged_ratio_data.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
ratio: 1

item_id: 15
ratio: 300.0/300.0

item_id: 18
ratio: 15.0/15.0

item_id: 22
ratio: 23600.0/23600.0

item_id: 23
ratio: 1/3

item_id: 24
ratio: 1.2/1.2

```

그림 21 <완성된 섭취량 비율 파일 중 일부>

상기 절차를 통해 모든 건강기능식품 데이터를 1회 섭취량 기준으로 통일 하였으며, 이를 바탕으로 1일 권장량과 상·하한 섭취량을 산출할 때 파싱 값으로 인한 오류가 발생하지 않도록 하였다.

또한 사용자의 자율적이고 건강한 섭취 판단을 지원하기 위해 보건복지부 '2020 한국인 영양소 섭취기준'을 활용하여 1일 권장 섭취량과 상·하한값 을 제공하였고, '건강기능식품 과잉섭취 방지 및 의약품 병용섭취 시 주의 사항' 전문가용 자료를 근거로 의약품-건강기능식품 간 병용 금기 여부를 판별할 수 있는 구조를 마련하였다. 다만 해당 자료들은 홍보용 형태로 제공되어 즉시 활용이 어려웠으므로, 원문을 분석·분류한 후 시스템에서 직 접 사용할 수 있도록 CSV 형식으로 재가공하였다.

성별	연령	평균 필요량	권장 섭취량	충분 섭취량	상한 섭취량	mg
영아	0-5개월			2		※ 1 mg NE(나이아신당량) = 1 mg 나이아신 = 60 mg 트립토판
	6-11개월			3		
유아	1-2세	4	6		10	
	3-5세	5	7		10	
남자	6-8세	7	9		15	
	9-11세	9	11		20	
	12-14세	11	15		25	
	15-18세	13	17		30	
	19-29세	12	16		35	
	30-49세	12	16		35	
	50-64세	12	16		35	
	65-74세	11	14		35	
여자	75세 이상	10	13		35	
	6-8세	7	9		15	
	9-11세	9	12		20	
	12-14세	11	15		25	
	15-18세	11	14		30	
	19-29세	11	14		35	

그림 22 <2020 한국인 영양소 섭취기준 재가공 데이터 일부>

성분명1	성분명2	금기사유	금기타입	비고
인삼	Azathioprin	인삼과 면역억제제의 동시 복용은	면역억제치료를 방해하는 것으로 보입니다.	
인삼	basiliximab	인삼과 면역억제제의 동시 복용은	면역억제치료를 방해하는 것으로 보입니다.	
인삼	cyclosporin	인삼과 면역억제제의 동시 복용은	면역억제치료를 방해하는 것으로 보입니다.	
인삼	daclizumab	인삼과 면역억제제의 동시 복용은	면역억제치료를 방해하는 것으로 보입니다.	
인삼	muromona	인삼과 면역억제제의 동시 복용은	면역억제치료를 방해하는 것으로 보입니다.	
인삼	mycophen	인삼과 면역억제제의 동시 복용은	면역억제치료를 방해하는 것으로 보입니다.	
인삼	tacrolimus	인삼과 면역억제제의 동시 복용은	면역억제치료를 방해하는 것으로 보입니다.	
인삼	sirolimus	인삼과 면역억제제의 동시 복용은	면역억제치료를 방해하는 것으로 보입니다.	
인삼	prednisone	인삼과 면역억제제의 동시 복용은	면역억제치료를 방해하는 것으로 보입니다.	
인삼	phenelzine	인삼이 페넬진(phenelzine)의 효과를 증가시켜	인삼과 페넬진 병용 섭취 시 불면증, 두통, 떨림 및 경조증이 나타날 수 있습니다.	
인삼	aspirin	항응고제, 항혈소판제 또는 소염진통제와	인삼을 병용하면 출혈의 위험을 증가	
인삼	cilostazol	항응고제, 항혈소판제 또는 소염진통제와	인삼을 병용하면 출혈의 위험을 증가	
인삼	clopidogrel	항응고제, 항혈소판제 또는 소염진통제와	인삼을 병용하면 출혈의 위험을 증가	
인삼	diclofenac	항응고제, 항혈소판제 또는 소염진통제와	인삼을 병용하면 출혈의 위험을 증가	
인삼	ibuprofen	항응고제, 항혈소판제 또는 소염진통제와	인삼을 병용하면 출혈의 위험을 증가	
인삼	naproxen	항응고제, 항혈소판제 또는 소염진통제와	인삼을 병용하면 출혈의 위험을 증가	
인삼	dalteparin	항응고제, 항혈소판제 또는 소염진통제와	인삼을 병용하면 출혈의 위험을 증가	
인삼	enoxaparin	항응고제, 항혈소판제 또는 소염진통제와	인삼을 병용하면 출혈의 위험을 증가	
인삼	heparin	헤파린(heparin)과 인삼은 모두 항응고를	증가시킵니다. 동시에 복용하면 출혈과 같은 위험한 상호작용이 있으므로 주의를 필요로 합니다.	
인삼	ticlopidine	항응고제, 항혈소판제 또는 소염진통제와	인삼을 병용하면 출혈의 위험을 증가	
인삼	warfarin	항응고제, 항혈소판제 또는 소염진통제와	인삼을 병용하면 출혈의 위험을 증가	
인삼	nifedipine	니페디핀(nifedipine)과 인삼을 함께 복용할	경우, 인삼이 니페디핀의 혈압강하 효과를 증가시킬 수 있습니다.	
인삼	raltegravir	인삼은 말테그라비브(raltegravir)와	같이 복용하였을 때 간효소의 수치가 상승하며, 혈청 말테그라비브(raltegravir) 농도가 높아졌	
인삼	fexofenadi	인삼과 페소페나딘을 동물실험에서	인삼과 페소페나딘(fexofenadine)을 병용하면 페소페나딘의 생물학적 이용 가능성을 감소시키는 것으로 밝혀졌습니다.	
인삼	chlorzoxaz	클로록사존(chlorzoxazone)을 포함한	약물은 간에서 분해가 되는데 인삼은 이러한 약물들의 분해를 촉진시킬 수 있습니다.	
인삼	theophylline	테오필린(theophylline)을 포함한	약물은 간에서 분해가 되는데 인삼은 이러한 약물들의 분해를 촉진시킬 수 있습니다.	
인삼	imatinib	인삼은 항암제 이마티닙(imatinib)과	병용 섭취 시 간 독성을 일으킬 수 있습니다.	
인삼	insulin	인삼을 인슐린과 같이 복용할 때	혈당을 낮추어 저혈당을 유발할 수 있어 인슐린 투여 시 주의를 필요로 하며 투여량을 조절할 필요가 있습니다.	
프로바이오티스	Amoxicillin	프로바이오티스와 항생제를 병용하면	프로바이오티스의 효과를 감소시키거나 프로바이오티스 개체 수를 줄일 수 있으므로 섭취 간격을 2시간 이상 두어야 합니	

그림 23 <건강기능식품 과잉섭취 방지 및 의약품 병용섭취 시 주의사항을 재가공한 csv 일부>

이를 통해 건강기능식품에 대해서도 권장 섭취량 산정과 의약품과의 병용 금기 여부를 판별할 수 있는 DUR 시스템을 구축하였다. 아래는 건강기능 식품 데이터 처리에 관련된 사항을 플로우차트로 도식화한 것이다.

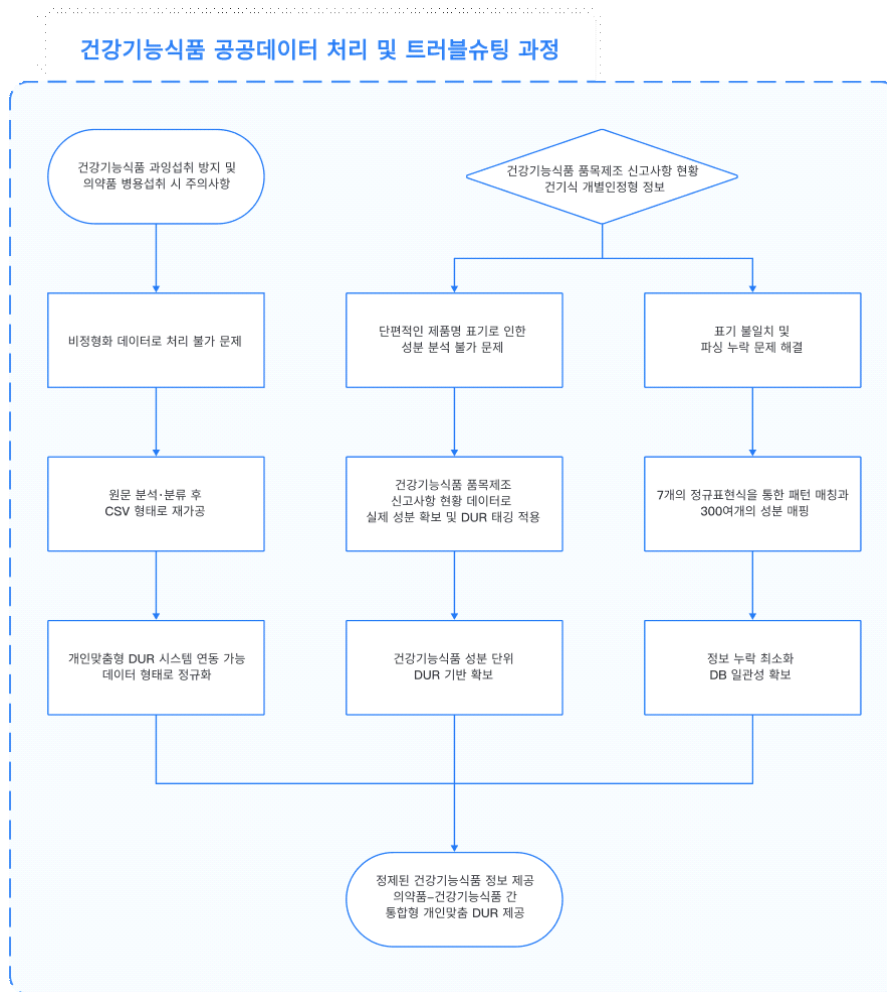


그림 24 건강기능식품 공공데이터 처리 플로우차트

### 3.2 사용자 정보 및 의약품 DB 구성

본 데이터베이스 설계의 핵심 원칙은 단일 기능 수행을 위해 과도한 데이터를 일괄 조회하는 상황을 방지하고, 요청 맥락에 따라 필요한 정보만을 최소 비용으로 가져오도록 하는 데에 있다. 이를 위해 의약품/건강기능식품(DB) 스키마를 기능 요구와 조회 경로 중심으로 구조화하였다.

우선, 제품 기본 정보는 다수 기능에서 공통적으로 참조되므로 제품 단위로 1:1 매핑되는 테이블에 단일 튜플로 저장하였다. 기본 정보의 범주는 ATC 코드, 제형, 유효기간, 제조사 등으로 구성하였다. 이렇게 함으로써 목록/상세/검색 등 광범위한 조회에서 추가 조인 없이 핵심 식별·설명 정보를 즉시 활용할 수 있게 하였다.

제품의 효능·용법·주의사항은 특정 기능에서 해당 정보만 선택적으로 필요할

---

수 있음을 고려하여, 제품 ID를 외래키로 갖는 1:N 구조로 분리 저장하였다. 다만 이들 세 범주를 지나치게 세분 테이블로 정규화할 경우 상세 화면 조회 시 과도한 JOIN이 발생해 성능 저하가 우려되므로, 하나의 테이블에 유형 구분을 두어 관리하도록 설계하였다.

성분과 성분 함량은 별도로 분리하였다. 동일 성분을 포함하는 제품이 다수 존재하므로 성분을 제품 레코드 내부에 중복 저장하는 것은 비효율적이며, 성분별 함량 계산 시에도 동일 성분 여부 판별 비용이 증가한다. 이에 성분을 독립 테이블로 관리하고, 성분 함량은 제품 ID와 성분 ID를 결합한 복합키(제품 ID, 성분 ID)를 가지는 연결 테이블로 저장하여 특정 제품의 특정 성분 함량을 직접적으로 표현하도록 하였다. 이 구조를 통해 동일 성분 여부는 DB 차원에서 보장되고, 계산 단계에서는 함량 비교만 수행하면 된다.

보관 정보는 UI 상 별도 탭에서 지연 로딩하도록 초기에 기획했었기에, 불필요한 조인을 피하기 위해 독립 테이블로 분리하였다. 이로써 상세 화면에서도 필요한 시점에 한해 보관 정보를 개별 조회할 수 있다.

결과적으로 본 스키마는 제품 기본 정보의 단일 접근, 효능·용법·주의사항의 선택적 고립 조회, 성분/함량의 중복 제거 및 계산 친화적 구조, 보관 정보의 지연 조회를 균형 있게 충족하며, 사용 시나리오별로 최소 조인·최소 전송량을 지향하는 고효율 데이터를 제공한다.

아래는 시스템 전체 ERD 및 세부 ERD이다.





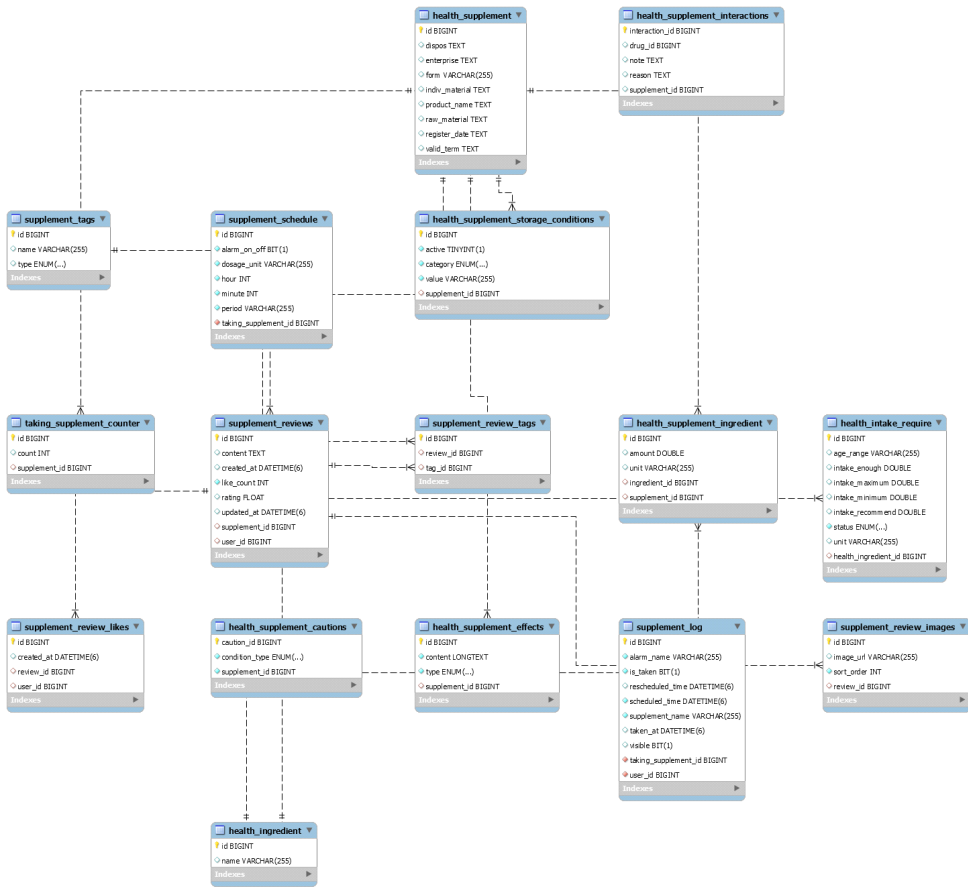


그림 28 <건강기능식품 데이터베이스 ERD>

### 3.3 개인맞춤형 DUR

본 연구에서는 사용자에게 실시간 개인 맞춤형 DUR(Drug Utilization Review) 정보를 제공하기 위해 메모리 기반 데이터베이스인 Redis를 활용하였다.

#### Solution

#### 개인 맞춤형 DUR 시스템 탑재

기존 서비스는 동일한 약품 - 건강기능식품 데이터를 안내만 제공할 뿐 개인화된 기능은 전혀 없었습니다. 그러나, 필립은 자체 개발한 개인맞춤형 DUR 시스템을 통해 사용자의 건강 정보와 복용 이력을 바탕으로, 맞춤형 의약품 안전 가이드의 최적화된 건강기능식품 큐레이팅 서비스를 제공하고자 합니다.

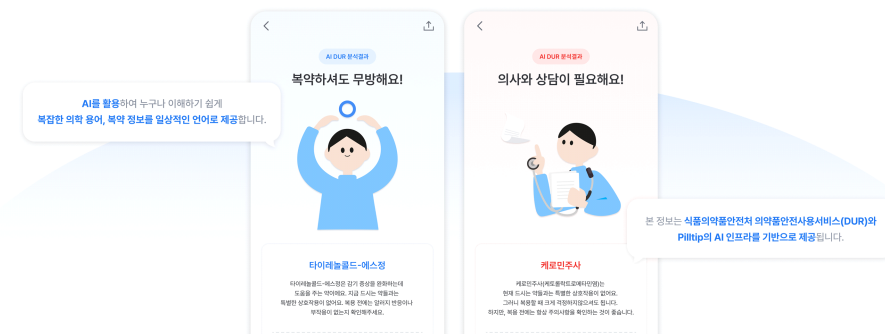


그림 29 <앱 내 개인맞춤형 DUR 시스템 탑재>

---

본 연구가 지향한 DUR 시스템은 검색 결과에 포함된 모든 의약품에 대해 일괄적으로 DUR 검사를 수행하고, 금기 사항이 존재할 경우 주의 표시를 부여하는 기능을 제공한다. 그러나 방대한 DUR 정보를 관계형 데이터베이스(MySQL)에서 SQL 문으로 직접 조회하는 방식은 응답 지연이 커 비효율적이라고 판단하였다. 이에 따라 Redis에 데이터를 캐시하여 사용자 요청 시 지연 없이 결과를 반환하도록 시스템을 설계하였으며, 키 스키마는 DRUG:DUR:AGE:{drugId} 형태로 정의하였다.

```
private void saveDrugCautions() throws JsonProcessingException {
    List<DrugCaution> cautions = drugCautionRepository.findAll();

    for (DrugCaution dc : cautions) {
        String key = "DRUG:DUR:" + dc.getConditionType().name() + ":" + dc.getDrugId();
        Map<String, String> value = Map.of(
            "conditionValue", dc.getConditionValue() == null ? "" :
            dc.getConditionValue(), dc.getNote() == null ? "" : dc.getNote()
        );
        redisTemplate.opsForValue().set(key, objectMapper.writeValueAsString(value));
    }
}
```

이후 DUR 데이터는 Redis에 Map<String, Map<String, String>> 구조로 적재하였다. 사용자의 연령과 성별 등 특성 및 복용 대상 의약품 식별자(drugId)를 입력으로 받으면, 사전에 정의한 키를 통해 1차 Map을 조회하고, Map<String, String> 형태의 세부 DUR 정보를 즉시 반환하도록 설계하였다. 이 세부 Map에는 category(금기 종류), conditionValue(금기 이유), note(비고) 등의 필드를 포함하도록 하였으며, 존재하는 필드를 활용해 사용자 맞춤 금기 정보를 실시간으로 제공할 수 있었다.

```

public List<DurTagDto> checkForWithoutInteraction(Drug drug, UserProfile userProfile, DurUserContext userContext) throws
JsonProcessingException {
    List<DurTagDto> tags = new ArrayList<>();
    Long drugId = drug.getId();

    // 임부금기
    tags.add(buildDurTag("임부금기", readJsonFromRedis("DRUG:DUR:PREGNANCY:" + drugId), userProfile.isPregnant()));

    // 노인금기
    tags.add(buildDurTag("노인금기", readJsonFromRedis("DRUG:DUR:ELDER:" + drugId), userContext.isElderly()));

    // 연령금기
    Map<String, String> ageValue = readJsonFromRedis("DRUG:DUR:AGE:" + drugId);
    boolean showAgeTag = ageValue != null && isUserInRestrictedAge(userProfile.getBirthDate(),
ageValue.get("conditionValue"));
    tags.add(buildDurTag("연령금기", ageValue, showAgeTag));

    // 효능군 중복주의
    Map<String, String> therValue = readJsonFromRedis("DRUG:DUR:THERAPEUTIC_DUP:" + drugId);
    String className = therValue != null ? therValue.get("className") : null;
    boolean isDup = className != null && userContext.classToProductIdsMap().containsKey(className);
    tags.add(buildDurTag("효능군중복주의", therValue, isDup));

    return tags;
}

```

복용 중인 의약품 간 상호작용 여부를 판별하기 위해 DRUG:DUR:INTERACT:{drugId} 형태의 키를 정의하고, 해당 의약품과 병용금기 정보가 존재하는 대상 약품 목록을 사전 매핑하였다. 이를 통해 모든 약품 조합을 전수 탐색하는 비효율을 제거하고, 병용금기 후보만 선별하여 비교하는 효율적 루프를 구성하였다.

```

private void saveDrugInteractions() throws JsonProcessingException {
    List<DrugInteraction> interactions = drugInteractionRepository.findAll();
    Map<String, List<String>> map = new HashMap<>();
    Map<Long, String> drugIdNameMap = drugRepository.findAll().stream()
        .collect(Collectors.toMap(Drug::getId, Drug::getName));

    for (DrugInteraction di : interactions) {
        map.computeIfAbsent(drugIdNameMap.get(di.getDrugId1()), k -> new ArrayList<>
()).add(drugIdNameMap.get(di.getDrugId2()));
        String key1 = drugInterDetailTag + drugIdNameMap.get(di.getDrugId1()) + ":" + drugIdNameMap.get(di.getDrugId2());

        Map<String, String> value = Map.of(
            "reason", di.getReason() == null ? "" : di.getReason(),
            "note", di.getNote() == null || di.getNote().equals("없음") ? "" : di.getNote()
        );

        String json = objectMapper.writeValueAsString(value);
        redisTemplate.opsForValue().set(key1, json);
    }

    pushKey(map, drugInterTag);
}

```

두 의약품(drugId1, drugId2) 간의 구체적 상호작용 정보는 DRUG:DUR:INTERACT\_DETAIL:{drugId1}+{drugId2} 형태의 키에 저장하여, 필요 시 해당 키를 통해 상세 정보를 즉시 조회할 수 있도록 설계하였다.

```

private List<DurTagDto> checkInteractionBetweenTwoDrugs(Drug drug1, Drug drug2) throws JsonProcessingException
{
    List<DurTagDto> tags = new ArrayList<>();
    String drugName1 = drug1.getName();
    String drugName2 = drug2.getName();

    List<String> contraList = redisTemplate.opsForList().range(drugInterTag + drugName1, 0, -1);
    if (contraList != null && contraList.contains(drugName2)) {
        String detailKey = drugInterDetailTag + drugName1 + ":" + drugName2;
        Map<String, String> detail = readJsonFromRedis(detailKey);
        List<DurDto> tagDesc = new ArrayList<>();
        if (detail != null) {
            tagDesc.add(new DurDto(
                drugName1 + " + " + drugName2,
                detail.getOrDefault("reason", ""),
                detail.getOrDefault("note", "")
            ));
        }
        tags.add(new DurTagDto("병용금지", tagDesc, !tagDesc.isEmpty()));
    }
}

```

이와 같은 Redis 기반의 효율적 데이터 구조와 고속 처리 덕분에, 사용자는 복용 예정 의약품의 개인별 금기 및 상호작용 정보를 실시간으로 확인할 수 있다.

### 3.4 암호화 및 비식별화

#### 3.4.1 AES-GCM

AES-GCM은 데이터의 기밀성과 무결성을 동시에 보장하는 인증 암호화 방식으로, 평문을 암호화함과 동시에 위변조 여부를 검증하는 인증 태그를 생성한다. 본 연구에서는 사용자의 민감 개인정보 보호를 위해 AES-GCM을 적용하였으며, Java와 MySQL 간 데이터 저장·호출 과정에서 암복호화가 자동 수행되도록 코드를 구성하여, 데이터베이스가 외부로 유출되더라도 평문 노출을 방지하도록 보안을 강화하였다.

#### 3.4.2 TLS

TLS는 인터넷 구간의 통신을 암호화하여 기밀성·무결성·상호 인증을 보장하는 표준 보안 프로토콜이며, HTTPS가 대표적 적용 사례이다. 모바일 앱/웹과 서버 간 민감 정보가 오가는 개발 환경 특성을 고려하여, 공인 인증기관에서 발급한 TLS/SSL 인증서를 서버에 적용하고 전 구간 통신에 TLS를 의무화하여 안전한 전송 경로를 구축하였다.

#### 3.4.3 Nginx

Nginx는 고성능 웹 서버이자 리버스 프록시로서, 클라이언트 요청을 수신해 백엔드 서버로 안정적으로 분산 전달한다. 본 연구에서는 Nginx에 TLS/SSL 인증서를 적용해 통신을 암호화하고, 리버스 프록시 기능으로 요

청 분산을 수행하여 서비스 안정성을 제고하였다.

### 3.4.4 산업체 멘토 의견 반영

아울러 중간보고서에서 산학협력 멘토 의견을 반영하여, JWT 기반 로그인과 AES-GCM을 활용한 저장 데이터 암호화에 더해, 공인 TLS/SSL 인증서를 적용하고 Nginx로 프론트엔드-백엔드 간 통신 암호화를 추가 구성하여 높은 수준의 개인정보 보안 시스템을 구축하였다.

## 3.5 자동완성 및 검색

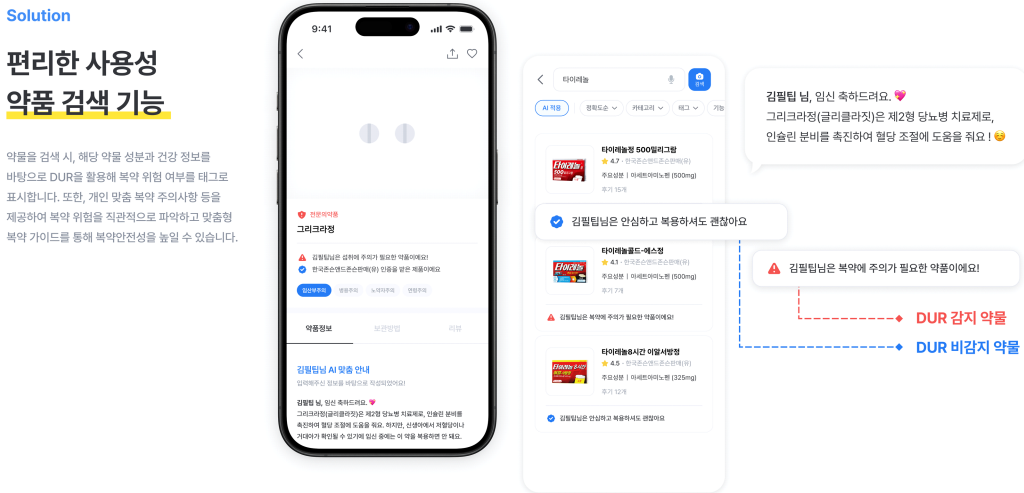


그림 34 <앱 내 자동완성 및 검색 기능>

### 3.5.1 Elasticsearch

Elasticsearch는 대용량 데이터에 대한 실시간 검색·분석을 지원하는 분산형 검색 엔진이며, 본 연구에서는 사용자의 검색 편의 극대화를 위해 이를 채택하였다. 특히 자동완성 기능 구현을 위해 N-gram과 Edge N-gram 토크나이즈를 병행 적용하였다.

N-gram은 문자열을 N개씩 나누어 분할하는 방식으로, 예를 들어 "자동완성"이라는 문자열이 들어왔을 때 N의 제한이 1~3이라면 "자", "동", "완", "성", "자동", "동완", "완성", "자동완", "동완성"과 같이 모든 가능한 문자열 조합을 색인합니다. 이를 통해 검색어의 어느 부분에 해당하더라도 관련된 결과를 찾을 수 있도록 한다.

Edge N-gram은 N-gram과 동일한 방법이지만, 항상 접두사를 포함한다는

---

점에서 다르다. 예를 들어 "자동완성"이라는 단어는 "자", "자동", "자동완", "자동완성"과 같이 접두사만 색인한다.

순위 산정에서는 사용자의 일반적 기대에 부합하도록 접두사 일치(Edge N-gram)에 가중치를 상향 부여하고, 중간 일치(N-gram)는 보조 신호로 유지하되 상대 가중치를 낮추어 후순위에 배치하였다. 이 구성으로 첫 글자 입력만으로도 높은 정확도의 후보를 신속히 제시하는 한편, 앞부분 입력이 부정확한 경우에도 중간 부분 문자열로 결과를 회수하여 검색의 정확성과 편의성을 동시에 확보하였다.

### 3.5.2 DUR 태깅

DUR(Drug Utilization Review) 태깅은 의약품의 오남용을 방지하고 안전한 복용을 지원하기 위해 약물 간 상호작용 정보를 데이터베이스에 구조적으로 부착하는 과정이다.

본 연구에서는 Elasticsearch에 색인된 의약품 정보에 앞서 기술한 Redis를 결합하여 DUR(의약품안전사용서비스) 데이터를 연동하였다. 검색 결과 화면에는 DUR 태그를 노출하여 상호작용 가능 품목과 금기 품목을 한눈에 식별하도록 하였고, 의약품 상세 화면에서는 기본 정보와 함께 약물 간 상호작용 위험과 복용 시 주의사항을 즉시 확인할 수 있도록 구성하였다. 이를 통해 사용자는 개인의 건강 상태와 기존 복용 약물을 종합적으로 고려하여 보다 안전하게 의약품을 선택할 수 있다.

### 3.5.3 이미지 데이터 및 YOLO v8

본 연구에서는 사용자가 알약 이미지를 촬영하여 원하는 의약품을 손쉽게 검색할 수 있도록 AI 기반 이미지 인식 기술을 도입하였다. 이를 위해 대량의 알약 이미지 데이터를 수집·정제하여 YOLO(You Only Look Once) 모델을 학습하였다.

YOLO는 실시간 객체 탐지 모델로, 이미지를 분석하여 알약의 모양, 색상, 식별 문자 등을 인식하도록 훈련하였다. 그 결과 사용자가 알약 사진을 업로드하면 모델이 해당 알약을 식별하여 의약품 검색을 가능하게 함으로써, 기존의 문자 기반 검색의 한계를 보완하였다.

### 3.6 약품 정보 및 맞춤형 안내

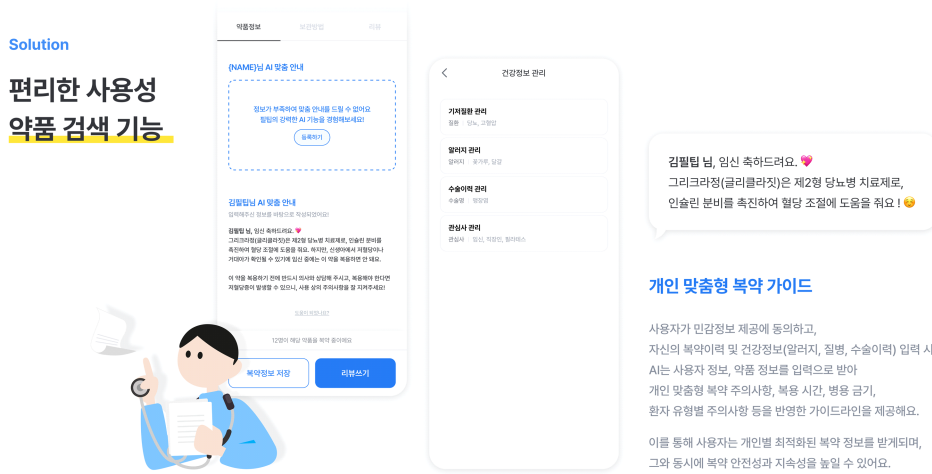


그림 35 <앱 내 개인 맞춤 복약 종합 안내 기능>

#### 3.6.1 구어체 변환(GPT API)

본 연구에서는 사용자가 난해한 의약품 정보를 쉽게 이해하도록 GPT API를 활용해 전문 서술을 구어체로 재작성하였다. 단순한 문장 변환에 그치지 않고 프롬프트 엔지니어링 기법을 적용하여 모델 응답의 품질과 일관성을 극대화하였다.

#### 3.6.2 Prompt Engineering

구체적으로 프롬프트에 "문장은 따뜻하고 공감 가는 말투(해요체)로 바꿔주세요.", "전문적인 의학용어는 고치지 말고, 환자가 이해할 수 있도록 괄호로 쉬운 설명을 붙이세요."와 같은 지시를 명시하여, 약품 정보가 친절하고 이해하기 쉬운 형태로 출력되도록 유도하였다. 이와 같은 절차를 통해 사용자 관점에서 의약품 정보 접근성이 향상되었다.

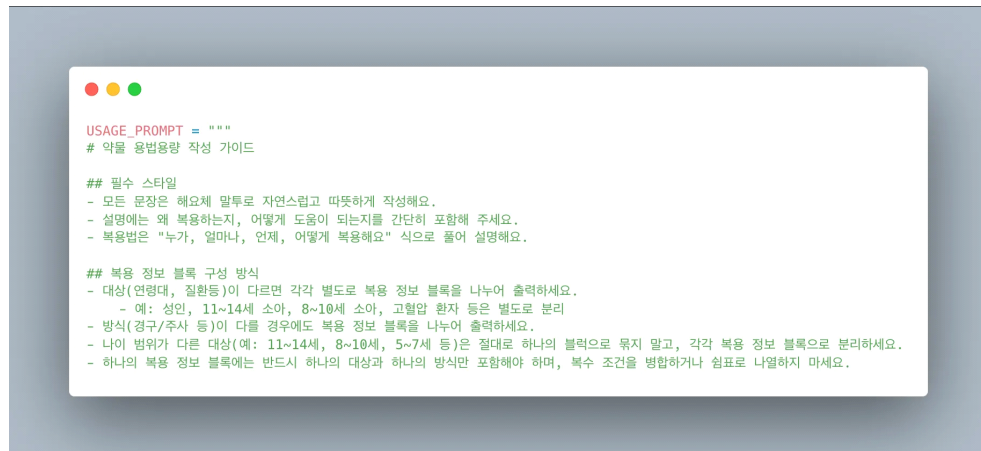


그림 36 <구어체 프롬프팅 예시>

### 3.6.3 개인 맞춤 안내

또한 사용자가 제공한 개인 정보(성별, 연령, 건강 상태 등)와 약품 정보를 결합하여 맞춤형 안내와 리뷰를 제공하였다. GPT API에는 프롬프트 엔지니어링을 적용해 사용자 특성을 반영하도록 하였으며, "이 약은 어린이에게는 권장되지 않아요", "고혈압 환자는 복용 전 의사와 상담하세요"와 같이 개인의 상황에 맞춘 조언을 구어체로 생성하도록 구성하였다.

```
// DUR 정보
sb.append("DUR 정보: ");
sb.append(dto.durInfo() == null || dto.durInfo().isEmpty() ? "\"\" :
dto.appendf{});

// 사용자 정보
sb.append("사용자 정보: { ")
.append("닉네임 : ").append(dto.nickname()).append("\n")
.append("나이 : ").append(dto.age()).append("\n")
.append("성별 : ").append(dto.gender()).append("\n")
.append("임신 여부 : ").append(dto.isPregnant()).append("\n")
.append("알러지 : ").append(dto.allergy()).append("\n")
.append("기저질환 : ").append(dto.underlyingDisease()).append("\n")
.append("복용중인 약 : {");

if (dto.currentDrugs() != null && !dto.currentDrugs().isEmpty()) {
String drugList = dto.currentDrugs().stream()
.map(d -> "\" + d + "\"")
.collect(Collectors.joining(", "));
sb.append(drugList);
}
sb.append("}\n");
sb.append("복용중인 건강기능식품 : {");

if (dto.currentSupplements() != null && !dto.currentSupplements().isEmpty()) {
String supplementList = dto.currentSupplements().stream()
.map(d -> "\" + d + "\"")
.collect(Collectors.joining(", "));
sb.append(supplementList);
}
sb.append("} }\n");
```

그림 37 <개인맞춤형 서비스 제공 코드 중 일부>

이러한 맞춤형 안내는 단순한 정보 전달을 넘어, 개인의 건강 상태에 부합하는 의사결정을 촉진함으로써 올바른 복용 습관 형성과 의약품의 안전한 활용에 실질적으로 기여할 수 있다.

### 3.7 복약 관리

#### Solution

#### 놓치지 않을 거예요 스마트 복약 알림

미리 설정한 알람 시작마다  
푸시 알림이 전송됩니다.  
알림 내 '먹었어요' 버튼을 통해  
간편하게 복약을 기록하고 복약 순응도를 높일 수 있어요.

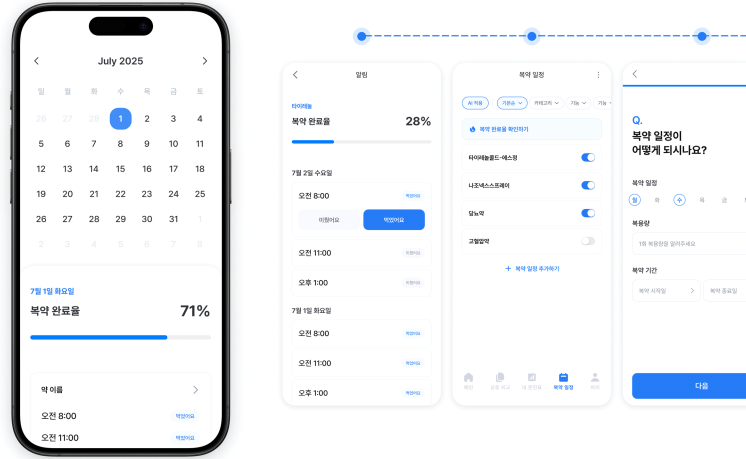


그림 38 <앱 내 복약 알림, 등록 및 관리 기능>

#### 3.7.1 복약 완료율 시각화

본 연구에서는 사용자가 복용해야 할 약품과 시간을 등록하면 해당 정보를 체계적으로 관리하는 시스템을 구축하였다. 사용자는 약품을 등록하고 복용 시간을 설정할 수 있으며, 시스템은 이 데이터를 기반으로 정해진 시간에 약을 복용했는지 여부를 추적한다.

복약 이행률을 한눈에 파악할 수 있도록 시각화 기능을 제공하였다. 사용자의 복약 기록을 바탕으로 막대그래프를 생성하여, 정해진 시간 대비 실제 복용 비율을 퍼센트(%)로 제시함으로써 사용자가 자신의 복약 습관을 점검하고 개선할 수 있도록 지원한다.

#### 3.7.2 FCM을 활용한 복약 알림

복약 시간을 놓치지 않도록 Firebase Cloud Messaging(FCM)을 활용한 실시간 알림 기능을 구현하였다. 등록된 복약 시간에 맞춰 사용자 기기에 푸시 알림을 전송하여 복용 시점을 안내함으로써 규칙적인 복약 습관 형성을 돕는다.

이와 같은 복약 등록·관리, 복약 알림, 복약 시각화 기능을 통해 사용자는 복약 습관을 편리하게 관리할 수 있으며, 시각적 지표를 통해 복약 이행률을 확인하여 복약 순응도를 효과적으로 높일 수 있다.

### 3.8 문진표 및 QR 촬영

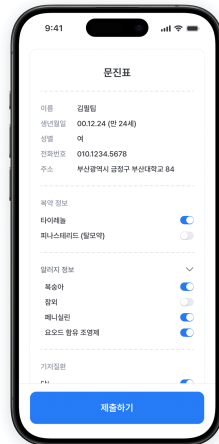
#### Solution

#### 철저한 보안과 신뢰 스마트 문진표

병원에 비치된 필립 QR코드를 촬영하면 내 스마트 문진표가 병원 측으로 안전하게 제출돼요. 제출된 문진표는 일정 시간이 지나면 자동 파기되어 내 소중한 개인정보가 안전하게 보호됩니다. 병원 측에선 간편하게 원자를 관리할 수 있습니다.



\* GPT 생성 이미지



[사용자를 앱 화면]

문진표 관리 (병원측)

문진표 관리

환자명 검색 입력해주세요

2025년 7월 10일

제출 시각	남은 조회 시간	주소	성별	나이	환자명	환자 연락처	복약
2025/07/01 11:54	54분	부산시 금정구 부산대대로 3	남	35세	이수근	010-1234-5678	무
2025/07/01 11:52	52분	부산시 금정구 언천항로 139	남	61세	김호동	010-2345-4567	유
2025/07/01 11:50	50분	부산시 금정구 석동원로 64	여	38세	이은지	010-3456-5678	유
2025/07/01 11:49	49분	부산시 동래구 구정로 1	남	49세	은지원	010-5634-2634	무
2025/07/01 11:46	46분	부산시 금정구 석동원로 64	남	52세	나영석	010-1632-8456	유
2025/07/01 11:39	39분	부산 동래구 금강공원로 27	남	40세	조규환	010-3628-4567	유
2025/07/01 11:02	2분	부산광역시 해운대구 센텀로	여	22세	이영지	010-2346-6123	무
2025/06/30 16:28	-	-	-	-	산지니	-	-
2025/06/29 14:49	-	-	-	-	김기문	-	-
2025/06/29 10:19	-	-	-	-	이주혜	-	-
2025/06/28 09:12	-	-	-	-	강태민	-	-
2025/06/27 09:09	-	-	-	-	김계형	-	-

[병원용 웹 화면]

그림 39 <앱 내 스마트 문진표 제출 기능 및 병원 측 문진표 조회 웹>

#### 3.8.1 React

React는 재사용 가능한 UI 컴포넌트로, 사용자 인터페이스를 구성하는 자바스크립트 라이브러리이다. 본 연구에서는 환자가 QR 코드로 제출한 문진표를 병원 측 웹에서 열람할 수 있도록 하는 페이지 개발에 활용하였다. 문진표는 제출 시점으로부터 1시간의 열람 제한이 존재하므로, React의 상태 관리로 남은 시간을 실시간 표시하고, 제한 시간이 경과하면 자동으로 접근을 차단하도록 구현하여 시간 기반 접근 제어의 효율성을 확보하였다.

#### 3.8.2 Next.js

Next.js는 React 기반의 서버 사이드 렌더링(SSR) 프레임워크로, 초기 렌더링을 서버에서 수행해 HTML을 전달함으로써 페이지 로딩 지연을 줄이고 반응성을 향상시키는 데 사용하였다. 이를 통해 병원 측이 문진표를 신속하게 확인할 수 있도록 페이지 로딩 속도를 최적화하였다.

또한 JavaScript의 동적 타입 특성으로 인한 잠재 오류를 예방하고 코드 안정성을 높이기 위해 TypeScript를 도입하였다. 정적 타입을 통해 컴파일 단계에서 타입 불일치 등을 조기 탐지하고, 변수와 함수의 역할을 명확히 기술함으로써 가독성과 유지보수성을 높였으며, 협업 과정의 의사소통 비용을 감소시켰다. 종합하면, React·Next.js·TypeScript의 결합을 통해 문진

표와 같은 중요한 데이터를 다루는 시스템의 데이터 무결성을 강화하고, 런타임 버그를 최소화하여 최종 애플리케이션의 신뢰도를 제고하였다.

### 3.9 피어링 및 계정 전환

#### Solution

#### 내가 직접 챙기는 우리 가족 건강 관리

사용자는 개인정보보호법에 따라, 만 14세 미만의 자녀 또는 가족의 정보를 필드에 등록하고 대신 관리할 수 있어요. 터치 한 번으로 계정을 전환하고 필드의 모든 기능을 사용해보세요!

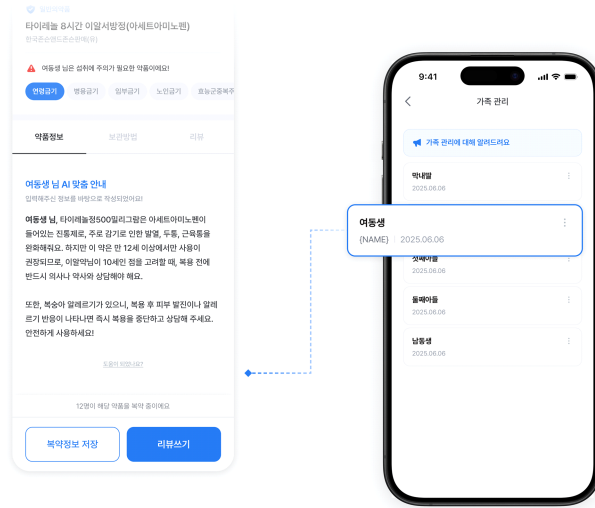


그림 40 <앱 내 가족관리(계정 전환) 기능 및 친구 맺기 기능>

#### 3.9.1 친구 맺기(피어링)

본 연구의 피어링(친구 맺기) 및 가족 관리 기능은 사용자 간 협업형 복약 관리를 가능하게 하여 복약 순응도를 높이고 돌봄 부담을 분산하도록 설계하였다. 두 기능은 최소한의 진입 장벽과 명시적 동의 절차를 전제로 하며, 계정·프로필 단위의 데이터 분리 원칙을 준수한다.

피어링 기능은 초대자가 앱에서 생성한 초대 링크를 메시저로 전송하고, 수신자가 해당 링크를 통해 앱에 접속해 친구 요청을 수락하는 흐름으로 동작한다.

초대 링크가 카카오톡 등 메시저에서 정상적으로 작동하도록 도메인 주소를 사용하며, 해당 도메인에 연결된 HTML 랜딩 페이지를 구성한다. 이 랜딩 페이지에서 링크 정보를 해석하여 앱의 딥링크로 리디렉션함으로써, 사용자가 한 번의 클릭으로 초대 수락 화면에 도달하도록 처리한다.

수신자가 앱에서 수락을 완료하면 관계 상태가 활성화되며, 상호 복약 기록 확인과 복약 알림 전송 등 상호 복약 관리가 가능해진다. 이를 통해 사

용자는 지인이나 보호자의 도움으로 복약 시간을 놓치지 않도록 독려받고, 실사용 맥락에서 소셜 지원 체계를 형성할 수 있다.

### 3.9.2 가족 관리(계정 전환)

가족 관리 기능은 넷플릭스의 멀티 프로필 개념을 적용하여 하나의 로그인 계정으로 여러 사용자의 개인정보와 복약 정보를 독립적으로 저장·관리한다. 안드로이드 클라이언트는 활성 프로필을 선택한 뒤 모든 API 호출에 대해 Request Header로 X-Profile-Id를 포함해 전송하며, 서버는 이 식별자를 기준으로 프로필 범위 내에서만 데이터에 접근하도록 강제한다. 이로써 동일 계정 내에서도 프로필 간 데이터가 엄격히 분리되고, 어떤 기능을 사용하더라도 선택된 프로필의 정보만 활용되는 일관성이 확보된다.

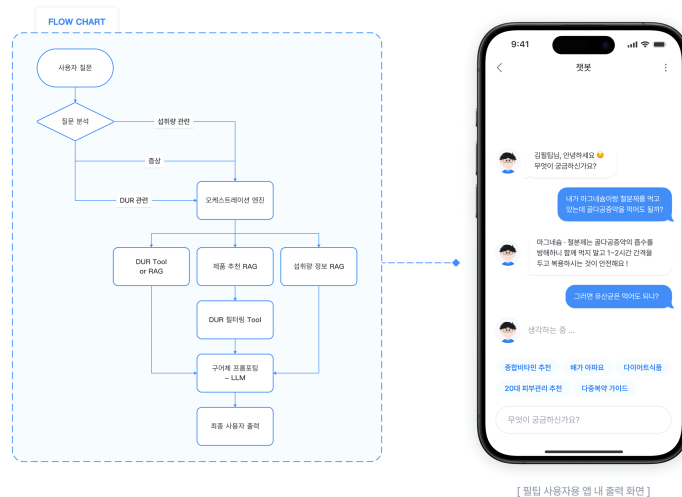
본 설계는 특히 20~40대 보호자가 70~80대 고령자나 아동의 복약을 동시에 관리하는 상황을 지원하는 데 초점을 맞춘다. 사용자는 앱 내에서 손쉽게 프로필을 전환하며 각 가족 구성원의 복약 이력 확인, 알림 설정, 일정 조정 등을 수행할 수 있어, 가정 단위의 건강 관리를 효율적으로 운영할 수 있다. 결과적으로 피어링을 통한 상호 감시·격려와 멀티 프로필 기반의 보호자 중심 관리가 결합되어, 사용자 경험을 해치지 않으면서도 안전성과 편의성을 동시에 달성한다.

## 3.10 AI 챗봇

### Solution

#### 나만의 약사님 필팁 AI 챗봇

DUR(의약품 안전성 검토)을 확인하지 않는 타사 챗봇들과 달리, 필팁은 사용자의 건강 정보와 복약 이력을 기반으로 개인 맞춤형 답변을 제공합니다. 또한, 자동으로 DUR 진행하여 안전한 제품만 추천하여 안전성과 편리함을 동시에 제공합니다.



[ 필팁 사용자용 앱 내 출력 화면 ]

그림 41 <앱 내 탑재된 필팁 AI 챗봇>

---

### 3.10.1 Orchestration

본 시스템의 핵심은 LLM 단독 사용으로 인한 할루시네이션을 구조적으로 억제하는 오케스트레이션에 있다. 사용자의 자연어 입력이 수신되면 Orchestration Engine이 의도 분류 결과를 바탕으로 사용할 톨과 실행 순서를 결정한다. 톨 호출은 코드로 검증 가능한 경로를 통해 수행되며, 최종 응답은 구어체 프롬포팅을 통해 사용자에게 전달된다. 이를 통해 정보 출처와 처리 과정을 추적 가능하게 만들고, 생성형 오류가 응답으로 전파되는 것을 차단한다.

### 3.10.2 자연어 처리(NLP)

대화형 질의에 대한 정확한 톨 선택과 파라미터 결정을 위해 의도 분류, 개체 인식, 슬롯 필링 기반의 NLP 전처리를 수행한다. 멀티턴 시나리오를 지원하기 위해 세션 컨텍스트를 유지하고(사용자 프로필, 직전 발화, 미해결 슬롯 등), 프롬프트 엔지니어링으로 필요한 정보만을 JSON 형태로 엄격히 추출하도록 설계하였으며, 해당 과정이 안정적으로 동작해야 톨 체인의 호출이 정확해지고, 최종 응답의 일관성과 재현성이 확보된다.

### 3.10.3 RAG

의료 도메인에서 할루시네이션은 치명적이므로, 검색증강생성(RAG)으로 지식 회수를 모델 밖에서 선행한 뒤 생성한다. 질문을 쿼리로 변환하여 벡터 임베딩과 전통 검색을 결합한 하이브리드 검색을 수행하고, 상위 Top-k 근거 문서를 바탕으로 응답을 구성한다. 벡터 DB는 Java 라이브러리를 제공하고 하이브리드 검색을 지원하는 Weaviate를 채택했으며, 제품 정보, DUR 규칙, 섭취량 근거를 인덱싱하였다. RAG 계층은 생성 단계에서 근거 문장을 함께 반영하도록 하여 응답의 검증 가능성을 높였다.

### 3.10.4 DUR Tool

제품 추천 결과는 반드시 개인 맞춤 DUR 필터를 통과해야 한다. DUR Tool은 사용자 데이터(나이, 임신 여부, 복용 중인 약·건강기능식품, 알러지, 기저 질환)를 입력받아 성분·제품 단위의 상충, 중복, 금기 여부를 판정하고, 필요 시 대체 가능 성분·제품을 제시한다. 필터 결과는 즉시 반영되며, 금기 항목은 차단하며, 추천 단계에서부터 안전성을 제도적으로 보장한다.

---

## 4. 연구 결과 분석 및 평가

### 4.1 개발 일정

본 프로젝트는 2025년 3월부터 9월까지 약 7개월 동안 진행되었으며, 개인 맞춤형 AI 복약 관리 애플리케이션의 핵심 기능을 설계·구현하고 검증하는 것을 목표로 했다. 프로젝트는 크게 기획 및 설계 단계, 기본 기능 개발, AI 프롬프팅 및 데이터 파이프라인 구축, 통합 및 고도화, 테스트 및 대회 출전, 최종 보고 준비로 구분된다.

3월에는 팀 결성과 함께 서비스 목표를 확정하고, 전체 시스템 아키텍처를 설계하였다. 이 과정에서 멀티 프로필 기반 사용자 데이터베이스 구조와 안전한 민감정보 분리 저장 방식을 마련하였다. 또한, 식품의약품안전처와 건강보험심사평가원의 공공데이터를 분석하여 복약 관리에 필요한 핵심 데이터셋을 확보하고 파싱을 통한 표준화 전략을 세웠다.

4월부터는 UI 디자인을 기반으로 하여 안드로이드(Android Jetpack Compose) 애플리케이션 프로토타입 개발을 시작했다. 회원가입 및 로그인, 약품 검색 및 복약 등록, DUR 등과 같은 핵심 기능이 탑재된 UI를 구현하였으며, Spring Boot 기반의 백엔드 서버 환경을 구축하고 MySQL, Redis, Elasticsearch를 연동하여 데이터 흐름을 실험하며 프론트엔드/백엔드 동시 작업이 진행됐다.

5월에는 AI 프롬프팅의 기초 작업이 본격화되었다. 공공 데이터셋을 정규화하고, 사용자 건강정보 및 복약이력 DB와 연동하여 개인맞춤형 DUR(의약품안전사용서비스) 기능을 구현하였고, GPT API를 활용하여 복약 설명과 주의사항을 읽기 쉬운 구어체로 변환하는 모듈을 개발했다. 동시에 FCM 기반 푸시 알림 시스템을 연동하여 복약 알림 기능을 구현했다.

6월에는 전체 기능을 통합하고 멀티 프로필 관리 기능, 복약 로그 시각화, 문진표 제출을 위한 QR 촬영 및 웹 송수신 기능을 완성했다. 또한, 이미지 인식 기반 약품 식별(딥러닝 YOLO 모델 활용) 학습 파이프라인 코드를 구축하였다. 단, 알약 이미지 원천 데이터의 용량이 4TB에 육박하여 실제 기능 개발은 오랜 시간이 소요됐다.

7월은 기능 고도화 단계로, 기존 DUR 시스템의 문제였던 단일 약품명에 국한된 상충작용 검증을 넘어, 성분 단위 다약제 검증 및 개인화 기능을 추가하고 Redis 캐싱을 활용해 실시간 응답 속도를 최적화하였다. 또한, 친구 피어링 기능을 추가하여 사용자가 서로의 복약 이력을 공유 및 관리할 수 있도록 했다. 서버에서는 민감정보 업데이트 API, 복약 기록 통계 API, AES-GCM 암호화, 도메인 연결 및 SSL TLS 암호화 등을 개발하여 실사용성 및 안전성을 높였다.

8월에는 대회 및 공모전 출전을 비롯하여, 통합 테스트와 버그 수정에 집중했다. 사용자 시나리오를 기반으로 한 E2E(End-to-End) 테스트를 진행하며, 안정적인 사용자 경험을 확보하기 위해 UI/UX를 개선하였다. 또한, Elasticsearch 기반 자동완성 검색

기능을 강화하여 사용자 편의성을 높였다. 부산대학교 앞 문창약국 약사님과의 컨택을 통해 사용자 피드백을 받을 발판을 마련하였다.

마지막으로 9월에는 최종 보고서 및 시연 준비가 진행되었다. 서비스 전반의 안정성을 점검하고, 발표를 위한 시연 영상을 제작하였다. 개발 전 과정은 Git 기반 버전 관리와 Notion의 주 단위 스프린트를 통해 체계적으로 관리되었으며, 본 프로젝트는 목표했던 개인 맞춤형 AI 복약 관리 시스템의 핵심 기능을 성공적으로 구현하고, 각종 대회 및 공모전에서 수상하며 실제 출시에 가까워지는 우수한 성과를 거두었다.

기간	주요 내용	세부 성과
3월	기획 및 설계	<ul style="list-style-type: none"> <li>- 팀 결성 및 서비스 목표 확정</li> <li>- 전체 시스템 아키텍처 설계</li> <li>- 멀티 프로파일 기반 DB 구조 및 민감정보 분리 저장 방식 마련</li> <li>- 식약처·심평원 공공데이터 확보 및 파싱/표준화 전략 수립</li> </ul>
4월	기본 기능 개발 착수	<ul style="list-style-type: none"> <li>- UI 디자인 기반 Android 프로토타입 개발 시작</li> <li>- 회원가입·로그인, 약품 검색, 복약 등록, DUR UI 구현</li> <li>- Spring Boot 백엔드 서버 구축</li> <li>- MySQL·Redis·Elasticsearch 연동 및 플로우 테스트</li> </ul>
5월	AI 프롬프팅 및 알림 기능	<ul style="list-style-type: none"> <li>- 공공데이터 정규화 및 DUR 기능 구현</li> <li>- 사용자 건강정보·복약이력 DB와 연동</li> <li>- GPT API 활용 복약 설명/주의사항 구어체 변환 모듈 개발</li> <li>- FCM 기반 푸시 알림 기능 구현</li> </ul>
6월	기능 통합 및 확장	<ul style="list-style-type: none"> <li>- 멀티 프로파일 관리 기능 완성</li> <li>- 복약 로그 시각화 및 QR 문진표 관련 기능 구축</li> <li>- YOLO 기반 약품 이미지 인식 학습 코드 개발</li> <li>- 4TB 규모 원천 데이터 처리 진행</li> </ul>
7월	고도화 및 보안 강화	<ul style="list-style-type: none"> <li>- 성분 단위 다약제 검증 및 개인화 DUR 기능 추가</li> <li>- Redis 캐싱으로 실시간 응답 최적화</li> <li>- 친구 추가(복약 공유) 기능 구현</li> <li>- 민감정보 업데이트 API, 복약 통계 API 개발</li> <li>- AES-GCM 암호화, SSL/TLS 적용으로 보안 강화</li> </ul>
8월	통합 테스트 및 대회 출전	<ul style="list-style-type: none"> <li>- E2E(End-to-End) 테스트 및 버그 수정</li> <li>- UI/UX 개선 및 ElasticSearch 자동완성 강화</li> <li>- 공모전 및 대회 출전, 약사 피드백 접촉(문창약국)</li> </ul>
9월	최종 보고 및 시연	<ul style="list-style-type: none"> <li>- 서비스 안정성 점검</li> <li>- 발표 및 시연 영상 제작</li> <li>- 핵심 기능 구현 완료 및 대회/공모전 수상 성과</li> </ul>

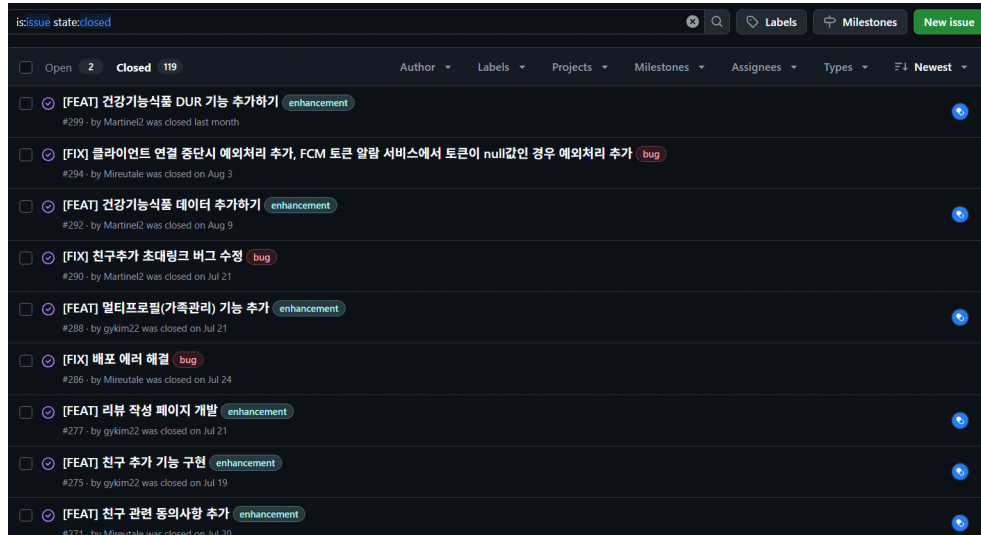
## 4.2 역할 분담

성명	역할	수행
김기윤	기획 / 안드로이드 프론트엔드	Kotlin 기반 Android 앱(Jetpack Compose) 화면 구성 앱 전반 UI/UX 및 기능 설계 및 작성 음성 검색 및 QR 인식 기능 작성 앱-서버 간 통신 로직 중 애플리케이션 담당 OkHttp/Flow 기반 SSE 챗봇 스트리밍 처리
강태진	백엔드	사용자 정보/복약 이력 저장 및 API 설계 데이터 암호화(AES-GCM), JWT 인증 처리 QR문진표 제출 및 웹 로직 구성 비식별화 처리 및 자동 파기 로직 구현 자연어 처리 및 오케스트레이션, RAG 구성
김재형	백엔드	OpenAI GPT-4o mini 연동 및 프롬프트 설계 Redis를 통한 약물 상호작용 분석 로직 구축 Elasticsearch API를 활용한 자동완성 FCM을 통한 알림 기능 구현 자연어 처리 및 오케스트레이션, RAG 구성

## 4.3 워크스페이스

### 4.3.1 Github

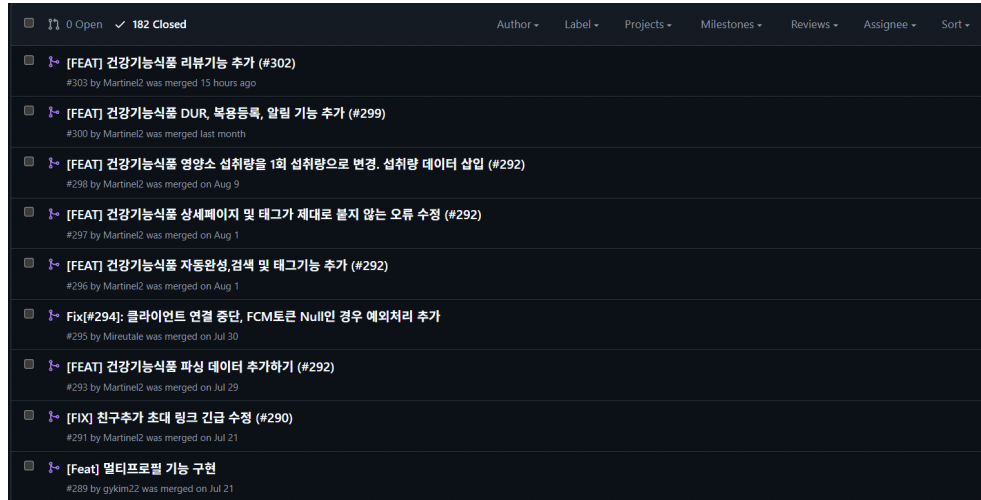
연구에서는 분산 버전 관리 시스템인 Git과 이를 기반으로 한 협업 플랫폼 GitHub를 결합하여, 버전 관리와 팀 협업을 표준화된 절차로 수행하였다. 저장소는 main과 dev의 이중 축을 중심으로 운영하고, 기능 개발은 이슈 단위로 정의한 뒤 브랜치를 분기하여 진행하였다.



<그림 2. 이슈 기록>

모든 개발 착수는 GitHub Issues에서 시작되며, 이슈 번호를 포함한 브랜치 네이밍(feat/ISSUE-123, fix/ISSUE-124 등)을 일관 적용해 변경 이력과 요구사항 간의 추적성을 확보했다. 브랜치 목적은 feature, chore, fix, hotfix로 구분하고, dev에서의 통합 검증을 통과한 변경만을 main에 병합함으로써 안정적인 릴리스

품질을 유지했다.

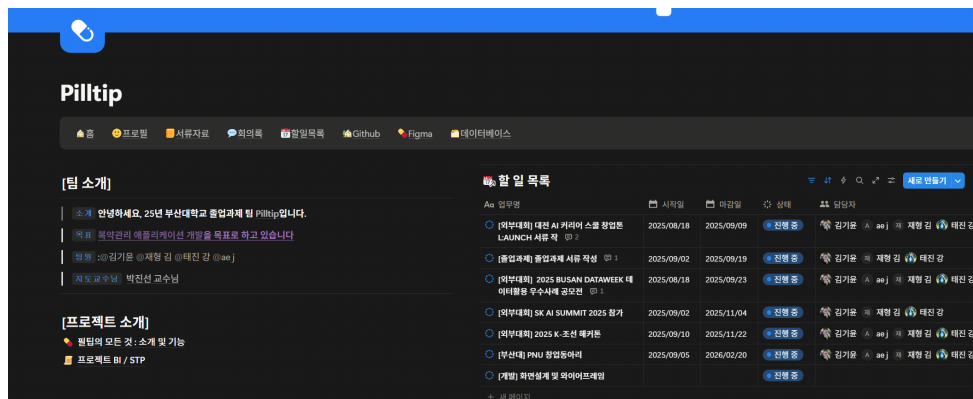


<그림 2. PR 기록>

코드 변경은 Pull Request(PR) 중심으로 관리했다. PR에는 개발 범위, 이유, 기간, 간단한 설명을 명시하고, 피어리뷰를 통해 구현 상의 토의를 기록으로 남겼다.

#### 4.3.2 Notion

본 연구는 노션(Notion)을 팀의 주력 워크스페이스로 설정하여, 기획·개발·운영 전 과정을 일관된 데이터베이스로 관리했다. 메인 대시보드 상단에는 홈, 프로필, 서류자료, 회의록, 할 일 목록, GitHub, Figma, 데이터베이스로 구성된 네비게이션을 배치해, 문서와 리포지토리, 디자인 산출물, 개발 데이터 간 이동 비용을 최소화했다. 각 섹션은 상호 링크되어 있어서 요구사항 변경이 코드와 일정, 디자인에 미치는 영향을 추적 가능한 형태로 연결된다.



<노션 워크스페이스>

의사결정 및 기록 관리 측면에서, 모든 의사결정은 회의를 거쳐 회의록으로 남겼다. 의제-결정-할 일을 구조화하여 남겼고, 논의 과정에서 생성된 근거 및 토론 자료는 해당 회의 페이지에 첨부했다. 회의 종료와 동시에 각 팀원에게 할당된 일을 할 일 목록 데이터베이스에 자동 등록되도록 연동하여, 회의→실행으로 이어지는 흐름을 문서 수준에서 보장했다. 이때 상태, 담당자, 마감일, 태그 속성을 표준화해, 필터·정렬·칸반·캘린더 뷰를 상황에 따라 전환해 사용했다.

일정 및 실행 관리에서는 “할 일 목록” 데이터베이스를 주력으로 활용했다. 각 업무 레코드는 시작일-마감일, 담당자, 진행 상태를 포함하며, 주간 스탠드업에서는 보드 뷰로 진행 상황을 점검했다. 마감 임박 항목은 캘린더-타임라인 뷰에서 한눈에 식별할 수 있다.

회의-의사결정-문서-일정-개발 산출물을 하나로 묶어, 보고서·개발·디자인·일정 스프린트 간을 유기적으로 연결하고, 신속한 의사결정을 가능하게 하여, 3-8월의 집중 개발 기간 동안 생산성을 확보하는 데 핵심 역할을 했다.

## 4.4 API 명세서

### 계정 관련 API

엔드포인트	메서드	설명
/api/auth/signup	POST	회원가입
/api/auth/login	POST	ID/PW를 활용한 로그인
/api/auth/social-login	POST	소셜 로그인
/api/auth/check-duplicate	POST	ID,닉네임,전화번호 중복 여부 조회
/api/auth/refresh	POST	리프레시 토큰을 활용한 JWT토큰 재발급
/api/auth/me	GET	사용자 프로필 정보 조회
/api/auth/terms	POST	앱 사용 동의
/api/auth/nickname?nickname={nickname}	PUT	닉네임 변경
/api/auth/personal-info	PUT	실명,주소 추가 및 수정
/api/auth/logout	PUT	로그아웃

### 사용자 프로필 관련 API

엔드포인트	메서드	설명
/api/user-profile/create	POST	프로필 생성
/api/user-profile/pregnant	PUT	임신 여부 갱신
/api/user-profile	DELETE	프로필 삭제

### 복약 정보 관련 API

엔드포인트	메서드	설명
/api/taking-pill	POST	복약 등록
/api/taking-pill	GET	복약 리스트 조회
/api/taking-pill/{medicationId}	GET	복약 세부사항 조회
/api/taking-pill	PUT	등록된 복약 정보 수정
/api/taking-pill/{medicationId}	DELETE	등록된 복약 이력 삭제

### 사용자 동의사항 관련 API

엔드포인트	메서드	설명
/api/questionnaire/permissions	GET	사용자 동의 목록 조회
/api/questionnaire/permissions/{permissionType}?grante={true or false}	PUT	단일 필드 동의
/api/questionnaire/permissions/multi	PUT	다중 필드 동의
/api/questionnaire/available	GET	문진표 사용 가능 여부 조회

### 사용자 민감정보 관련 API

엔드포인트	메서드	설명
/api/sensitive-info	GET	민감정보 전체 호출
/api/sensitive-info	PUT	민감정보 전체 수정
/api/sensitive-info/{category}	PUT	카테고리별 민감정보 수정
/api/sensitive-info	DELETE	카테고리별 민감정보 삭제
/api/sensitive-info/all	DELETE	민감정보 전체 삭제
/api/sensitive-info/profile	PUT	민감정보,이름,주소,전화번호 모두 변경

#### 문진표 관련 API

엔드포인트	메서드	설명
/api/questionnaire	GET	문진표 조회
/api/questionnaire	PUT	문진표 수정
/api/questionnaire	DELETE	문진표 삭제
/api/questionnaire/qr-url/{hostpital-code}	POST	JWT토큰으로 병원 코드 조회
/api/questionnaire/qr-url/all?accessToken={}	POST	QR 문진표 생성 및 제출

#### 약 정보 검색 관련 API

엔드포인트	메서드	설명
/api/autocomplete/drugs?input={drugName}&page=(defaultValue=0)	GET	약 이름으로 자동완성
/api/autocomplete/manufacturers?input={manufacturer}&page=(defaultValue=0)	GET	제조사로 자동완성
/api/autocomplete/ingredients?input={ingredientName}&page=(defaultValue=0)	GET	성분으로 자동완성
/api/search/drugs?input={drugName}&page=(defaultValue=0)	GET	약 이름으로 검색
/api/search/manufacturers?input={manufacturer}&page=(defaultValue=0)	GET	제조사로 검색
/api/search/ingredients?input={ingredientName}&page=(defaultValue=0)	GET	성분으로 검색
/api/detailPage?id={drugId}	GET	약 상세페이지 출력
/api/detailPage/gpt	POST	AI 맞춤 안내
/api/dur/gpt?drugId1={drugId1}&drugId2={drugId2}	GET	DUR 상충비교

#### 의약품 복용 알림 및 이력 관련 API

엔드포인트	메서드	설명
/api/alarm/token?token={String}	POST	FCM 토큰 발급
/api/drug/alarm/{logId}/pending	POST	5분뒤 알림 재전송
/api/drug/alarm/{logId}/taken	POST	복용 상태 토글 - 알림용
/api/dosageLog/{logId}/taken	POST	복용 상태 토글 - 앱용
/api/dosageLog/date?date={LocalDate}	GET	복약 정보 호출 - 1일치
/api/dosageLog/week?date={LocalDate}	GET	복약 정보 호출 - 1주치

#### 의약품 리뷰 관련 API

엔드포인트	메서드	설명
/api/review/create	POST	의약품 리뷰 추가
/api/review/delete/{reviewId}	DELETE	의약품 리뷰 삭제
/api/review/drug?drugId={drugId}&page=i	GET	의약품 리뷰 리스트 조회 - 정렬포함

nt}&size={int}&sortKey={String}&direction={String}		
/api/review/user	GET	사용자가 작성한 의약품 리뷰 조회
/api/review/{reviewId}/like	POST	리뷰 좋아요 토글
/api/review/drug/{drugId}/stats	GET	리뷰 평점 평균 및 태그 통계

### 병원 관련 API

엔드포인트	메서드	설명
/api/hospital/search?name=아산	GET	이름으로 병원 검색
/api/hospital	POST	병원 등록
/api/hospital/{id}	PUT	병원 수정
/api/hospital/{id}	DELETE	병원 삭제
/api/hospital/{HospitalCode}/access-token	GET	접근 토큰 조회 성공

### 친구 관련 API

엔드포인트	메서드	설명
/api/friend/invite	POST	친구 초대 링크
/api/friend/accept	POST	초대 수락
/api/friend/list	GET	친구 리스트 조회
/api/dosageLog/{friendId}/date?date={date}	GET	친구 복용정보 관리
/api/alarm/{friendId}/{logId}	GET	친구 복용 기록 알림 전송

### 건강기능식품 정보 관련 API

엔드포인트	메서드	설명
/api/autocomplete/supplements?input={supplementName}&page=(defaultValue=0)	GET	약 이름으로 자동완성
/api/autocomplete/enterprises?input={enterprise}&page=(defaultValue=0)	GET	제조사로 자동완성
/api/autocomplete/supplementIngredients?input={ingredientName}&page=(defaultValue=0)	GET	성분으로 자동완성
/api/search/supplements?input={supplementName}&page=(defaultValue=0)	GET	약 이름으로 검색
/api/search/enterprises?input={enterprise}&page=(defaultValue=0)	GET	제조사로 검색
/api/search/supplementIngredients?input={ingredientName}&page=(defaultValue=0)	GET	성분으로 검색
/api/supplement/detailPage?id={supplementId}	GET	약 상세페이지 출력
/api/supplement/detailPage/gpt	POST	AI 맞춤 안내

### 건강기능식품 복용 관리 API

엔드포인트	메서드	설명
/api/taking-supplement	POST	복약 등록
/api/taking-supplement	GET	복약 리스트 조회
/api/taking-supplement/{supplementId}	GET	복약 세부사항 조회
/api/taking-supplement	PUT	등록된 복약 정보 수정
/api/taking-supplement/{supplementId}	DELETE	등록된 복약 이력 삭제

### 건강기능식품 복용 알림 및 이력 관련 API

엔드포인트	메서드	설명
/api/supplement/alarm/{logId}/pending	POST	5분뒤 알림 재전송
/api/supplement/alarm/{logId}/taken	POST	복용 상태 토글 - 알림용
/api/supplementLog/{logId}/taken	POST	복용 상태 토글 - 앱용
/api/supplementLog/date?date={LocalDate}	GET	복약 정보 호출 - 1일치
/api/supplementLog/week?date={LocalDate}	GET	복약 정보 호출 - 1주치

### 건강기능식품 리뷰 관련 API

엔드포인트	메서드	설명
/api/supplement/review/create	POST	건강기능식품 리뷰 추가
/api/supplement/review/delete/{reviewId}	DELETE	건강기능식품 리뷰 삭제
/api/supplement/review/supplement?supplementId={supplementId}&page={int}&size={int}&sortBy={String}&direction={String}	GET	건강기능식품 리뷰 리스트 조회 - 정렬포함
/api//supplement/review/user	GET	사용자가 작성한 건강기능식품 리뷰 조회
/api/supplement/review/{reviewId}/like	POST	리뷰 좋아요 토글
/api/review//supplement/{supplementId}/stats	GET	리뷰 평점 평균 및 태그 통계

## 4.5 서비스 아키텍처

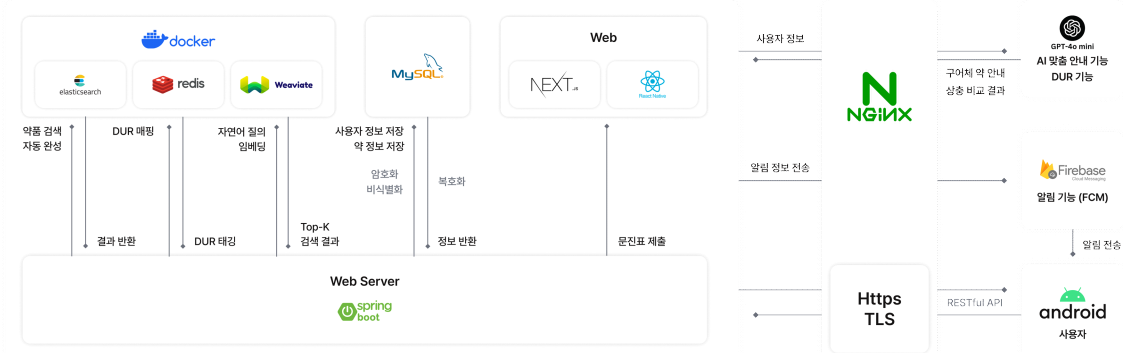


그림 45 Pilltip 서비스 아키텍처

## 4.6 수상 및 전시 이력

- 2025 SW중심대학 디지털 경진대회 : SW부문
  - 부산대학교 대표팀 선발 및 본선 진출
  - 서울 중구 대한상공회의소 발표 및 딥노이드 상 수상
- 2025 K-ICT in Busan
  - Pilltip 출품 및 BEXCO 부스 전시
- SK AI SUMMIT 2025
  - 우수사례 선정 및 전시부스 운영
  - 코엑스 (그랜드볼룸, 아셈볼룸, 오디토리움) 전시
- Busan Dataweek 2025 데이터활용 우수사례 공모전
  - 우수사례 선정
  - 벅스코 제2전시장 4A홀 전시
- Google.org AI 커리어스쿨 창업톤 LAUNCH
  - 현재 본선 진출

## 5. 결론 및 향후 연구 방향

### 5.1 결론

본 연구에서 수행한 Pilltip 프로젝트는 고령화 사회에서 심화되는 복약 문제와 기존 DUR 제도의 한계, 개인화의 부재 그리고 개인정보 보호 미비를 해결하기 위한 개인 맞춤형 AI 복약 관리 서비스를 구현하는 것을 목표로 하였다. 프로젝트의 결과, 사용자는 자신의 복약 이력을 안전하게 통합 관리하고, 의약품·건강기능식품을 아우르는 개인 맞춤형 복약 안내를 제공받을 수 있는 환경을 구축하였다. 특히, 성분을 추가한 데이터 모델과 실시간 상호작용 점검 로직은 기존 약품 단위 중심 체계가 놓치던 위험을 포착하도록 설계되어 안전 탐지의 범위와 정밀도를 동시에 끌어올렸다. 성과는 하기한 바와 같다.

---

첫째, 성분 단위의 데이터 표준화와 DUR 확장 적용으로 기존 DUR 시스템의 한계를 극복하고 실제 생활에 밀접한 복약 안전성을 확보하였다.

둘째, AI 기반 구어체 변환 및 맞춤형 복약 안내를 통해 디지털 헬스케어 취약계층 및 일반인에게도 이해하기 쉬운 정보를 제공할 수 있었다.

셋째, 프로필 단위의 데이터 분리 및 통신·저장 시 민감정보 암호/복호화 구조를 적용하여 개인정보 보안성과 활용성을 동시에 확보하였다.

넷째, 부산대 앞 문창약국에서의 키오스크형 베타테스트 준비를 통해, 실제 의료 현장과 사용자의 생활 환경에 적용 가능한 수준의 기술적 완성도를 검증할 기반을 마련했다.

이로써 Pilltip은 단순한 복약 관리 앱을 넘어, 디지털 전환 기반의 사회적 안전망으로서의 가능성을 입증하였다.

## 5.2 향후 연구 방향

본 연구는 성공적으로 마무리되었으나, 더욱 확장된 가치 창출을 위해 다음과 같은 연구 과제가 필요하다.

- 베타테스트 검증 및 고도화  
실제 약국 현장에서의 사용자 피드백을 반영하여 UI/UX를 개선해야한다.
- 데이터 확보를 통한 DUR 확장 및 정밀화  
기존 의약품 중심의 DUR을 넘어, 건강기능식품, 한약재, 복합제제까지 아우르는 전방위 상호작용 데이터베이스를 구축할 필요가 있다.
- 의료기관·플랫폼 연계  
병원 전자의무기록(EMR), 약국 조제 이력, 건강보험공단 데이터와의 연동을 통해 국가적 복약 안전 플랫폼으로 확장하는 것이 필요하다.
- 법적 이슈 해결  
사용자의 개인정보 및 의료 데이터를 방대하게 다루기에 철저한 관계 법령 분석과 검토가 필요하며, 구어체 안내나 DUR, 챗봇 등의 핵심 기능이 약사법, 의료법을 위반하지 않도록 약관과 주의사항 고지를 명확히 해야한다.

## 5.3 맺음말

필립 프로젝트는 단순히 하나의 애플리케이션을 넘어, "안전하고 능동적인

---

복약 관리”라는 새로운 패러다임을 제시하였다. 부산대 앞 문창약국에서 진행될 베타테스트는 그 첫걸음으로, 본 연구가 추구한 사회적 가치와 실질적 효과를 현장에서 증명하는 과정이 될 것이다. 향후 연구는 이러한 성과를 확산하고 제도화하여, 고령화 사회의 복약 안전 문제 해결에 기여할 수 있는 실질적인 디지털 헬스케어 인프라로 발전해야 한다.

## 6. 참고 문헌

[1] Google. Jetpack Compose [Online]. Available: <https://developer.android.com/jetpack/compose>

(accessed Sep. 18, 2025).

[2] Google. LiveData overview [Online]. Available: <https://developer.android.com/topic/libraries/architecture/livedata>

(accessed Sep. 18, 2025).

[3] Google. ViewModel overview [Online]. Available: <https://developer.android.com/topic/libraries/architecture/viewmodel>

(accessed Sep. 18, 2025).

[4] M. Fowler and D. Rice. Patterns of Enterprise Application Architecture: MVVM. Addison-Wesley, 2009.

[5] Kotlin Foundation. Coroutines Guide [Online]. Available: <https://kotlinlang.org/docs/coroutines-overview.html>

(accessed Sep. 18, 2025).

[6] Google. Hilt dependency injection [Online]. Available: <https://dagger.dev/hilt>

(accessed Sep. 18, 2025).

[7] Square. OkHttp [Online]. Available: <https://square.github.io/okhttp/>

(accessed Sep. 18, 2025).

[8] Elastic. Elasticsearch documentation [Online]. Available: <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>

(accessed Sep. 18, 2025).

---

[9] Redis Ltd. Redis documentation [Online]. Available: <https://redis.io/docs/>

(accessed Sep. 18, 2025).

[10] Weaviate. Weaviate vector database [Online]. Available: <https://weaviate.io/developers/weaviate>

(accessed Sep. 18, 2025).

[11] Docker Inc. Docker overview [Online]. Available: <https://docs.docker.com/get-started/overview/>

(accessed Sep. 18, 2025).

[12] OpenAI. GPT API reference [Online]. Available: <https://platform.openai.com/docs/>

(accessed Sep. 18, 2025).

[13] Google. Firebase Cloud Messaging (FCM) [Online]. Available: <https://firebase.google.com/docs/cloud-messaging>

(accessed Sep. 18, 2025).

[14] Pivotal Software. Spring Boot reference documentation [Online]. Available: <https://docs.spring.io/spring-boot/docs/current/reference/html/>

(accessed Sep. 18, 2025).

[15] Oracle. MySQL 8.0 reference manual [Online]. Available: <https://dev.mysql.com/doc/refman/8.0/en/>

(accessed Sep. 18, 2025).

[16] National Institute of Standards and Technology (NIST). Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D, 2001 [Online]. Available: <https://csrc.nist.gov/publications/detail/sp/800-38d/final>

(accessed Sep. 18, 2025).

[17] Internet Engineering Task Force (IETF). The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, 2008 [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc5246>

(accessed Sep. 18, 2025).

---

[18] Internet Engineering Task Force (IETF). The OAuth 2.0 Authorization Framework. RFC 6749, 2012 [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc6749>

(accessed Sep. 18, 2025).

[19] NGINX Inc. NGINX documentation [Online]. Available: <https://nginx.org/en/docs/>

(accessed Sep. 18, 2025).

[20] Meta. React documentation [Online]. Available: <https://react.dev/>

(accessed Sep. 18, 2025).

[21] Vercel. Next.js documentation [Online]. Available: <https://nextjs.org/docs>

(accessed Sep. 18, 2025).

[22] Ministry of Health and Welfare. Drug Utilization Review (DUR) system information and inspection statistics. Policy page, 2023–2025 [Online]. Available: <https://www.mohw.go.kr>

(in Korean).

[23] H. S. Jeon. Legislative direction for activation of DUR. Health Insurance Review and Assessment Service (HIRA) OAK Repository, Policy Research Report, 2019 [Online]. Available: <https://repository.hira.or.kr>

(in Korean).

[24] Health Insurance Review and Assessment Service (HIRA). DUR operation guidelines, including portal conditions and prescription/dispensing support software. Ministry of Health and Welfare Notice No. 2010-84, 2010 [Online]. Available: <https://www.hira.or.kr>

(in Korean).

[25] Ministry of Health and Welfare. Overview, history, and performance of the DUR system (2008 pilot onwards). 2023–2025 [Online]. Available: <https://www.mohw.go.kr>

(in Korean).

[26] KPANews. “Drug expenses increase 28% in 5 years, management of polypharmacy patients needed.” Seoul: KPANews; 2024 [Online]. Available: <https://www.kpanews.co.kr/article/show.asp?idx=253715&category=C>

(in Korean).

---

[27] Y. H. Moon. "47% get drug information from pharmacists vs 42% from the Internet." NewsTheVoice Healthcare; Apr. 13, 2023 [Online]. Available: <https://www.newsthevoice.com/news/articleView.html?idxno=32184>

(in Korean).

[28] H. R. Bang. "1.36 million patients take 10+ drugs for over 60 days." HitNews; Nov. 28, 2024 [Online]. Available: <https://www.hitnews.co.kr/news/articleView.html?idxno=59503>

(in Korean).

[29] Yonhap News. "53% of patient safety accidents are drug-related: mandatory DUR needed." Apr. 19, 2024 [Online]. Available: <https://www.yna.co.kr/view/AKR20240419071800530>

(in Korean).

[30] D. Y. Kim. "Prescription papers discarded as trash, mass personal data leakage." DealSite Economy TV; Apr. 19, 2022 [Online]. Available: <https://news.dealsitetv.com/articles/78278>

(in Korean).

[31] J. H. Hong. "[Reader complaints] Prescriptions abandoned in pharmacies, personal data exposed." Jemin Ilbo; Mar. 22, 2023 [Online]. Available: <https://www.jemin.com/news/articleView.html?idxno=752278>

(in Korean).

[32] J. H. Lee. "Ignored DUR for narcotics, 21.9 million duplicate prescriptions in 5 years." DailyPharm; Oct. 18, 2023 [Online]. Available: <https://www.dailypharm.com/Users/News/NewsView.html?ID=304930>

(in Korean).