

**SK네트웍스 Family AI과정 15기  
 모델링 및 평가 시스템 아키텍처**



**□ 개요**

* 산출물 단계 : 모델링 및 평가
* 평가 산출물 : 시스템 아키텍처
* 제출 일자 : 2025. 10. 02.
* 깃허브 경로 : https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN15-final-3Team
* 작성 팀원 : 권주연

| **컴포넌트 다이어그램** | * 구성 요소 * 설명 |
| --- | --- |
| **시퀀스 다이어그램** | * 참여자 (Actors) * 주요 흐름 |
| **액티비티 다이어그램** | * 구성 * 주요 액션 노드 |

**1. 컴포넌트 다이어그램**

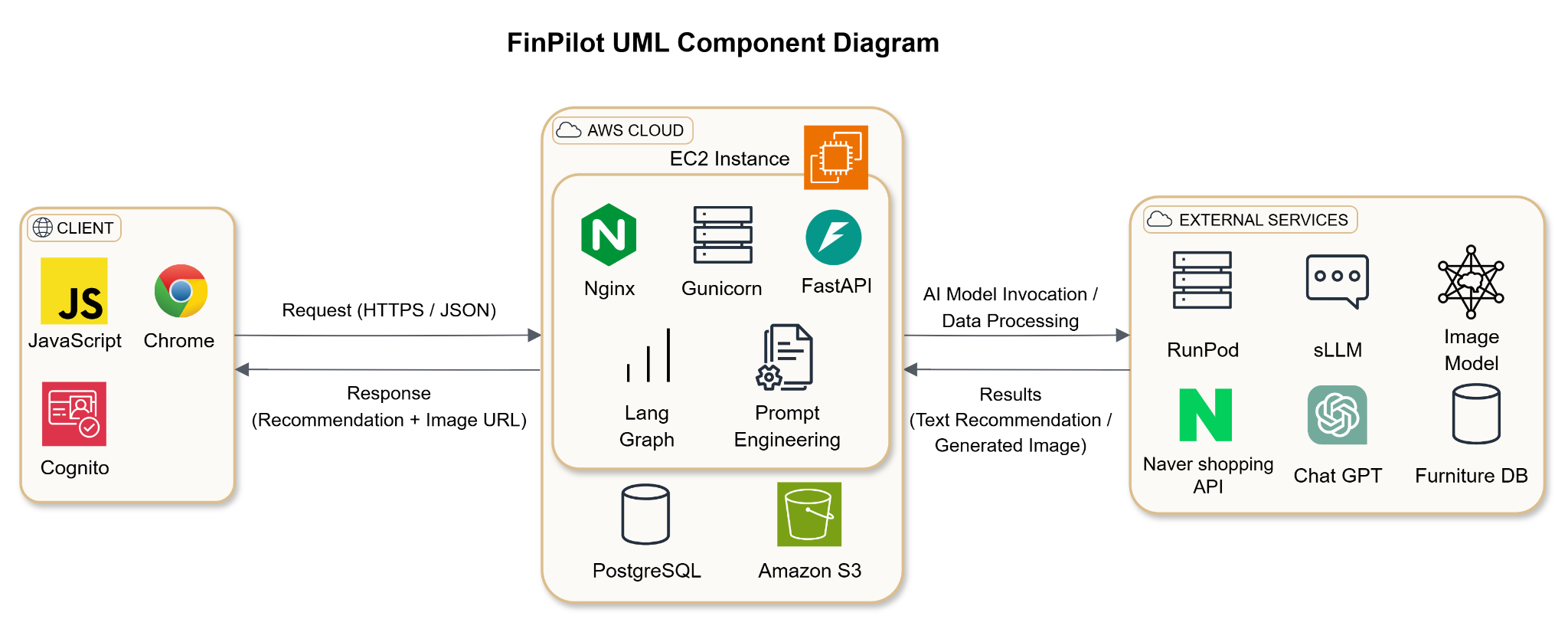
****

Figure 1. 컴포넌트 다이어그램

**구성 요소**

1. 클라이언트(Client)

- 컴포넌트

- JavaScript 기반 웹 애플리케이션 (메인 UI)

- Chrome 브라우저

- 인터페이스

- Cognito 기반 관리자 계정 로그인 (JWT 인증)

- REST API 호출 (HTTPS, JSON 포맷)

1. AWS Cloud

- EC2 인스턴스

- Nginx: 리버스 프록시 및 정적 파일 제공

- Gunicorn: WSGI 서버

- FastAPI: 백엔드 애플리케이션, 요청 처리 및 오케스트레이션

- Orchestration

- LangGraph: 요청 흐름 관리, AI 모듈 호출

- AI 처리 모듈

- sLLM (RunPod 실행): 사용자 입력 기반 인테리어 추천 방안 생성

- Prompt Engineering 계층

- sLLM의 출력 텍스트 → 이미지 생성 모델 prompt(JSON)로 최적화 변환

- 프롬프트 구조화 요소: 스타일, 오브젝트, 조명, 카메라 시점 등

- Image-to-Image 모델 (RunPod 실행)

- sLLM이 생성한 프롬프트 기반으로 인테리어 이미지 생성

1. 데이터베이스

- Amazon RDS (PostgreSQL)

- 사용자 계정 및 인증 데이터 저장  
 - 가구/자재 메타데이터 관리  
 - 생성 요청 및 결과 로그 저장

1. 스토리지

- Amazon S3

- 생성된 인테리어 이미지를 저장

- 이미지 URL을 프론트엔드로 반환

1. 외부 서비스

- RunPod

- sLLM 실행 환경

- 이미지 생성 모델 실행 환경

- ChatGPT

- 추가 프롬프트 최적화나 보조 질의 처리

1. 결과 반환

- FastAPI

- 추천 인테리어 방안(텍스트) + 생성된 이미지 URL을 JSON으로 묶어 프론트엔드에 반환

- 사용자:

- 웹 UI에서 결과 확인

- 불만족 시 “More Input”을 재전달 → sLLM →Prompt Engineering → Image Model 흐름 반복

**설명**

- 클라이언트는 Cognito 기반 관리자 계정 로그인을 통해 인증을 수행하고, 입력된 데이터를 JSON 포맷으로 HTTPS를 통해 AWS Cloud로 전송한다.

- AWS Cloud의 EC2 인스턴스는 요청을 수신하며, Nginx와 Gunicorn을 거쳐 FastAPI 애플리케이션이 실행된다.

- FastAPI와 LangGraph 오케스트레이터는 요청을 처리하면서 sLLM을 호출하여 인테리어 추천 방안을 생성하고, Prompt Engineering 계층을 통해 이미지 생성 모델이 이해할 수 있는 프롬프트(JSON 구조)를 생성한다.

- 이미지 생성 모델은 RunPod 환경에서 실행되며, 생성된 이미지는 Amazon S3에 저장되고 URL 형태로 반환된다.

- 동시에 LangGraph는 ChatGPT를 통해 프롬프트 기반 가구 키워드를 추출하고, 이를 Naver Shopping API로 검색하여 관련 상품을 얻는다.

- 검색된 상품명은 PostgreSQL의 가구/자재 메타데이터와 매칭되어 실제 가구 이미지와 가격 정보를 함께 제공한다.

- 최종적으로 클라이언트는 추천 인테리어 방안(텍스트), 생성된 이미지 URL, 가구 추천 정보(상품명, 이미지, 가격)를 응답으로 전달받아 웹 UI에서 확인할 수 있다.

**2. 시퀀스 다이어그램**

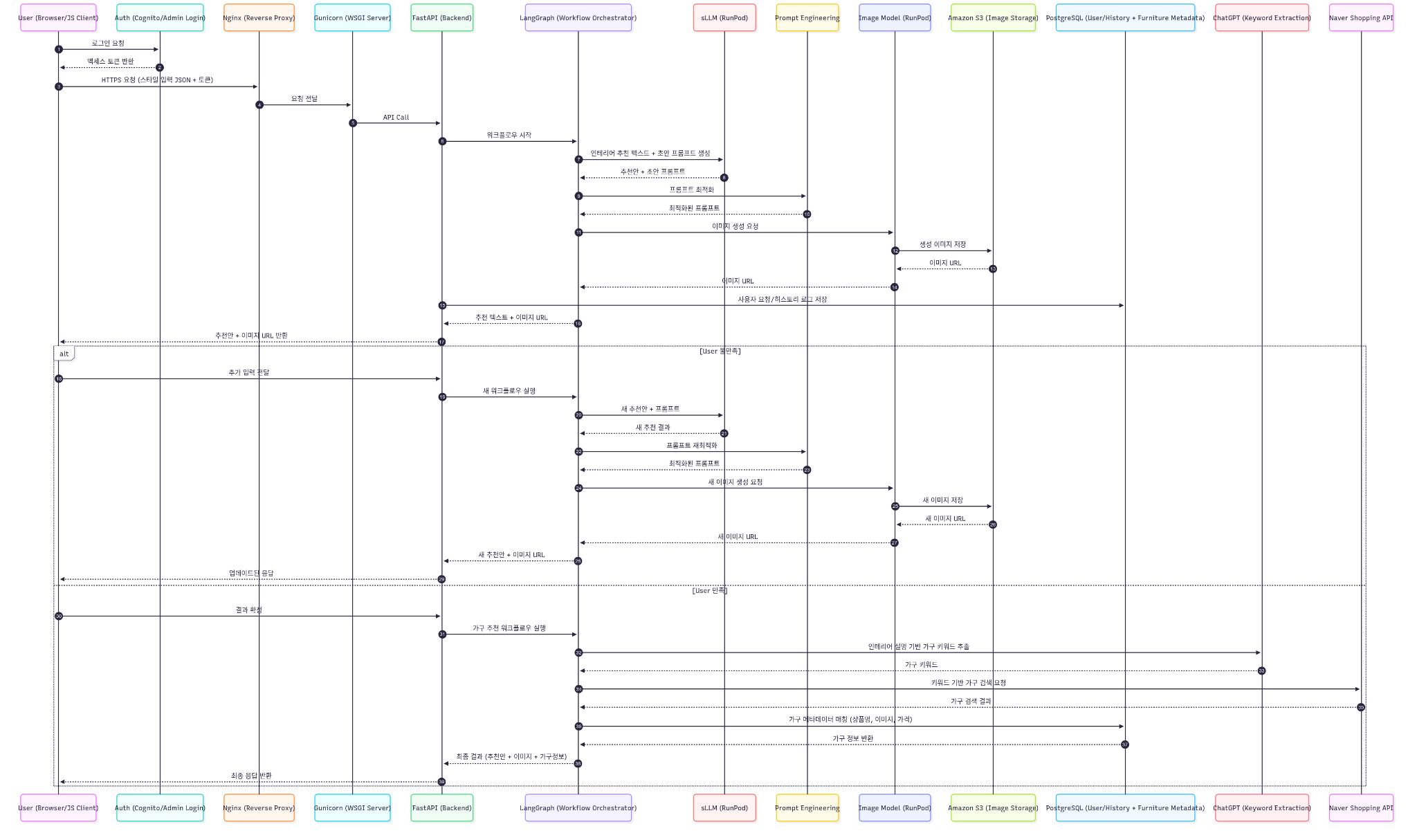
****

Figure 2. 시퀀스 다이어그램 | 별첨1 사진 참조

**참여자 (Actors)**

1. User: JS 프론트엔드 (Chrome 브라우저)
2. Auth: Cognito/Admin-Created Login(JWT)
3. AWS Cloud:

* Nginx: 요청을 수신, 리버스 프록시
* Gunicorn: FastAPI 실행 환경 (WSGI)
* FastAPI: 요청 처리, LnagGraph 호출 및 응답 집계
* LangGraph: 워크플로우 오케스트레이터 (sLLM 호출, Prompt Engineering 수행, ChatGPT/Naver Shopping API 호출)
* PostgreSQL: 사용자 로그/히스토리 및 가구 상품 검색
* Amazon S3: 생성된 인테리어 이미지 저장 및 URL 반환

1. External Services

* RunPod(sLLM, Image Model): 인테리어 추천 텍스트 생성 및 프롬프트 기반 이미지 생성 실행 환경
* ChatGPT: sLLM 결과 기반 가구 키워드 추출
* Naver Shopping API: 추출된 키워드로 가구 상품 검색 및 추천

**주요 흐름**

1. User: JavaScript 기반 프론트엔드(브라우저)에서 스타일 입력 후 요청 시작
2. Auth 모듈(Cognito/관리자 계정 기반 로그인): 사용자 인증 → 액세스 토큰 반환
3. Nginx: 클라이언트 요청 수신 → Gunicorn → FastAPI로 전달
4. FastAPI: LangGraph 호출 → 워크플로우 실행
5. LangGraph → sLLM(RunPod): 인테리어 추천 텍스트 및 초안 프롬프트 생성
6. LangGraph → Prompt Engineering: 생성된 텍스트를 이미지 모델용 JSON 프롬프트로 최적화
7. LangGraph → Image Model(RunPod): 최적화된 프롬프트로 이미지 생성 요청
8. Image Model → Amazon S3: 생성된 이미지를 저장, URL 반환
9. FastAPI ↔ PostgreSQL: 사용자 요청/히스토리 로그 저장
10. FastAPI → User: 응답 반환(추천 방안+이미지 URL)
11. User: 결과 확인
    1. 불만족 시: User → FastAPI로 추가 입력 전달 → 5)~10) 반복
    2. 만족 시: 결과 확정
12. LangGraph → ChatGPT: 최종 확정된 인테리어 설명 기반으로 가구 키워드 추출
13. ChatGPT → Naver Shopping API: 추출된 키워드로 가구 상품 검색
14. Naver shopping API ↔ PostgreSQL: 검색된 상품명과 DB 내 가구 메타데이터 매칭 → 가구 이미지/이름/가격 반환
15. FastAPI → User: 최종 확정 응답 반환(추천 인테리어 방안+확정된 이미지 URL+가구 추천 정보)

**3. 액티비티 다이어그램**

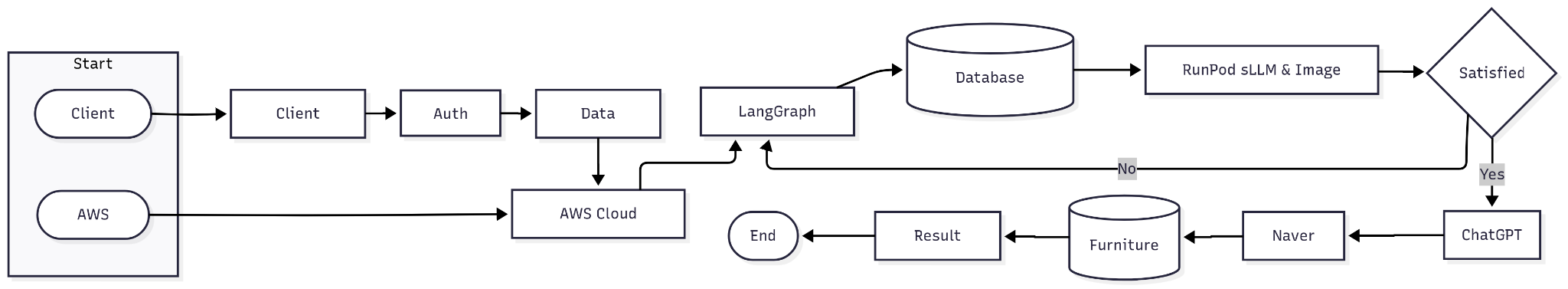
****

Figure 3. 액티비티 다이어그램 | 별첨2 사진 참조

**구성**

1. 시작: 사용자가 JavaScript 웹 클라이언트에서 스타일·옵션 입력 후 작업 시작

2. 클라이언트 활동: 로그인 요청 → Cognito 관리자 계정 인증 완료

3. 데이터 전송: 사용자 입력을 JSON 형식으로 만들어 HTTPS로 AWS에 전송

4. AWS 클라우드 내부 활동: Nginx 수신 → Gunicorn → FastAPI 오케스트레이션

5. LangGraph 오케스트레이션:

- sLLM 호출: 인테리어 추천 텍스트와 초안 프롬프트 생성

- Prompt 최적화: 이미지 모델 친화적 JSON으로 정제

- Image model 실행 RunPod: 이미지 생성

- S3 저장: 생성 이미지를 S3에 저장하고 URL 획득

6. 데이터베이스 작업: PostgreSQL을 통해 사용자 요청/히스토리/메타데이터 저장

7. 결과 반환 1차: FastAPI가 추천 텍스트 + 이미지 URL을 클라이언트로 응답

8. 만족 여부 분기

- 불만족: 사용자가 추가 입력 제공 → 5단계부터 반복

- 만족: 이미지 확정

9. 가구 추천 파이프라인 (이미지 확정 후 단 한 번 실행)

- ChatGPT: 최종 인테리어 설명 기반 가구 키워드 추출

- Naver Shopping API: 키워드로 상품 검색

- PostgreSQL 가구 메타데이터: 상품명 매칭 후 이미지·가격 확정

- 최종 집계: 생성 이미지 URL + 인테리어 설명 + 가구 정보 합쳐서 응답

10. 종료: 작업 완료

**주요 액션 노드**

- Client activity: 입력·인증·요청 전송

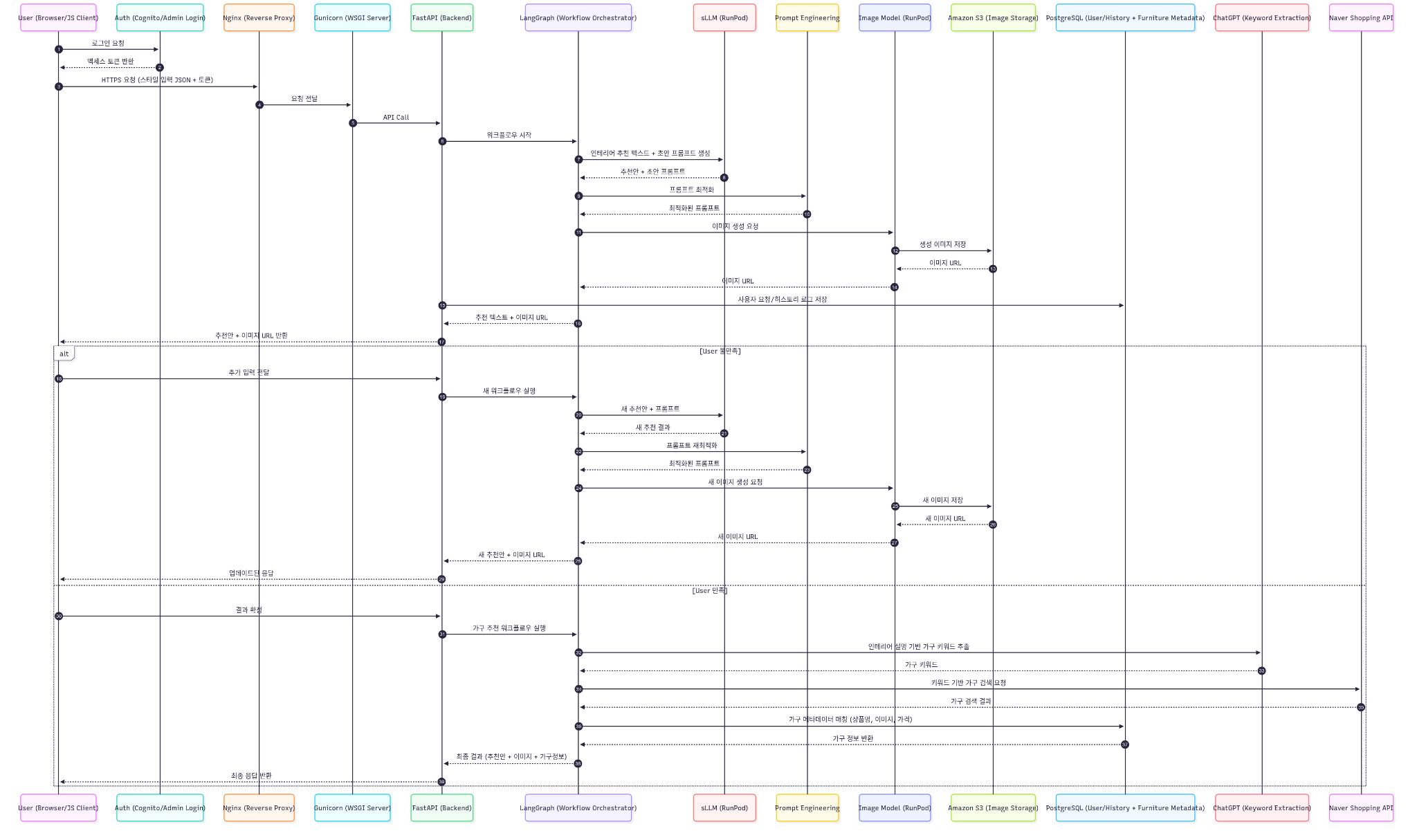
- Data transfer: HTTPS JSON 전송

- AWS cloud activity: Nginx → Gunicorn → FastAPI

- LangGraph orchestration: sLLM → 프롬프트 최적화 → 이미지 생성 → S3 저장

- Database ops: PostgreSQL 로그·히스토리·가구 메타데이터 저장  
- Result return: 1차 응답은 추천 텍스트 + 이미지 URL, 만족 후 최종 응답에 가구 정보 포함  
- Decision: 만족 여부 판단, 불만족 시 루프, 만족 시 가구 추천 실행

별첨 1



별첨 2

