

# 소프트웨어 설계 명세서(SDS)



이름/학번: 202212343 강현호, 202213523 손천익  
지도교수: 유준범 교수님

# 목차 (Table of Contents)

## 1. 소개 (Introduction)

1.1 SDS 의 목적 (Purpose)

1.2 문서 개요 (Overview)

1.3 참조 문서 (References)

## 2. 시스템 구조 설계 (System Architecture)

2.1 전체 아키텍처 개요 (System Overview)

2.2 계층 구조 (Layered Architecture)

## 3. 모듈 설계 (Module Design)

3.1 사용자 인증 모듈 (User Authentication Module)

3.2 스케줄 관리 모듈 (Schedule Management Module)

3.3 항공/호텔 검색 모듈 (Amadeus Integration Module)

3.4 관광지/레스토랑 검색 모듈 (Google Places Module)

3.5 거리 계산 모듈 (Distance Calculation Module)

3.6 AI 설계 (AI Design – Genetic Algorithm)

## 4. 데이터 설계 (Data Design)

4.1 MySQL 테이블 구조 (Database Schema)

4.2 Redis 데이터 구조 (Data Structure in Redis)

## 5. 인터페이스 설계 (Interface Design)

5.1 REST API 명세 (API Endpoints and Methods)

## 6. 보안 및 오류 처리 (Security & Error Handling)

## 7. 유지보수 및 확장성 (Maintenance & Scalability)

# 1. 소개 (Introduction)

## 1.1 목적 (Purpose)

본 문서는 "여행 일정 생성 AI" 시스템의 상세 설계(Detailed Design)를 기술한다. SRS(Software Requirements Specification)에서 정의된 요구사항을 충족시키기 위해, 시스템 아키텍처, 데이터 흐름, 모듈별 인터페이스, AI 설계 및 보안 설계를 구체적으로 작성한다. 최종적으로 개발자가 시스템을 명확히 이해하고 일관된 표준에 따라 개발을 수행할 수 있도록 하는 설계 문서로 활용된다.

## 1.2 문서 개요 (Overview)

본 SDS 는 다음의 핵심 영역으로 구성된다:

- 전체 아키텍처 (Architecture)
- 모듈 설계 (Module Design)
- 데이터 설계 (Data Design)
- AI 설계 (AI Design)
- 인터페이스 설계 (Interface Design)
- 보안 및 오류 처리 (Security & Error Handling)
- 배포 설계 (Deployment Design)

## 1.3 참조 문서 (References)

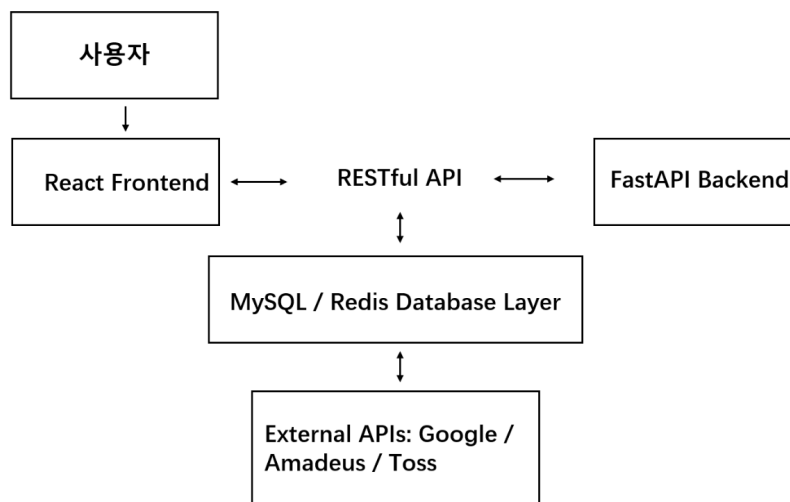
- SRS: "여행 일정 생성 AI 시스템 요구사항 명세서"
- IEEE Std 1016-1998 (Software Design Description Standard)
- Google Maps / Places / Distance Matrix API 문서
- Amadeus API 문서 (Flights, Hotels)
- Toss Payments SDK 및 PortOne 결제 API 문서
- Redis 및 MySQL 공식 문서
- FastAPI 공식 개발 문서
- React, Docker 공식 문서

## 2. 시스템 구조 설계 (System Architecture)

### 2.1 전체 아키텍처 개요 (System Overview)

본 시스템은 웹 기반 AI 시스템(Web-based AI System)으로, 다음과 같은 계층으로 구성된다.

- **프론트엔드 (Frontend):** React
- **백엔드 (Backend):** FastAPI
- **데이터베이스 계층 (Database Layer):** MySQL 및 Redis
- **외부 API 계층 (External API Layer):** Google API 및 Amadeus API



### 2.2 계층 구조 (Layered Architecture)

- **프론트엔드(Frontend):** 사용자 인터페이스 구현, Google Maps 를 활용한 지도 시각화, 사용자 일정 관리 뷰 제공
- **백엔드(Backend):** 시스템의 핵심 로직을 담당하는 REST API 제공, JWT 기반 사용자 인증 처리, 스케줄 관리, AI 모듈 호출 및 통합
- **데이터 계층(Data Layer):** MySQL 은 사용자 및 경로 정보의 영구 저장을 담당, Redis 는 사용자 세션 및 임시 데이터 저장에 사용되며, Arq 는 비동기 작업 큐 처리 시스템으로 활용
- **AI 모듈(AI Module):** 여행 노드 순서 최적화를 위한 유전자 알고리즘(GA) 기반의 경로 최적화 연산 처리
- **통합 계층 (Integration Layer):** Google API (Places, Distance Matrix), Amadeus (항공/호텔), Toss (결제) 등 외부 서비스와의 안정적인 API 연동을 수행

## 3. 모듈 설계 (Module Design)

### 3.1 사용자 인증 모듈 (User Authentication Module)

- 주요 기능: 회원가입(Sign-up), 로그인(Login), JWT 기반 인증
- 입력: id, password
- 출력: JWT token, user\_id
- 오류 처리: 중복 계정, 비밀번호 불일치, DB 연결 오류

### 3.2 스케줄 관리 모듈 (Schedule Management Module)

- 주요 기능: 사용자 일정(Schedule)을 Redis 에 임시 저장 및 조회
- 입력: ScheduleList(JSON), JWT Token
- 출력: 저장 결과 (success/failure)
- 비고: Redis 의 Hash 구조를 사용하여 일정 순번별로 관리

### 3.3 항공/호텔 검색 모듈 (Amadeus Integration Module)

- 연동 API: Amadeus Flight Offers, Hotel Search
- 입력: 출발/도착 공항, 도시명, 날짜, 인원
- 출력: 항공권/호텔 JSON 데이터
- 비고: Access Token 캐싱, 15 개 단위 청크 요청 (Chunk Request)

### 3.4 관광지/레스토랑 검색 모듈 (Google Places Module)

- 연동 API: Google Places, Geocoding
- 입력: 도시명, 위도/경도 28, 카테고리 (type), 키워드 (query)
- 출력: 장소 목록 (JSON: 이름, 평점, 위치 등)
- 비고: API Key 보호, type 검증 수행

### 3.5 거리 계산 모듈 (Distance Calculation Module)

- **연동 API:** Google Distance Matrix API
- **출력:** 장소 간 거리 행렬 (Matrix)
- **비고:** Arq Worker 기반 Queue 시스템을 통해 비동기 처리 실행

### 3.6 AI 설계 (AI Design – Genetic Algorithm)

- **방법:** 유전자 알고리즘(Genetic Algorithm, GA)을 활용한 AI 기반 경로 탐색 시스템
- **기능:** 여행 노드(Node) 순서 최적화
- **염색체:** 유전자 (Chromosome)여행 노드 ID 들의 순열 (permutation)
- **피트니스 함수:** 활동성 (Activity), 효율성(Efficiency), 균형성(Balance)의 가중합
- **반영 요소:** 거리, 체류 시간, 이동 효율, 방문 선호도 반영
- **Selection 방법:** Roulette Wheel 또는 Tournament 방식 선택
- **Crossover 방법:** Partially Mapped Crossover (PMX) 기법 사용
- **결과:** 부모 경로의 일부분을 교환하여 자식 염색체 생성
- **종료:** 조건결과 처리, 최종 경로를 Redis 에 저장하여 지도에 시각화

## 4. 데이터 설계 (Data Design)

### 4.1 MySQL 테이블 구조

	주요 columns	설명
User	id, email, password_hash	사용자 정보
경로	id, user_id, route_json, score	최적화 결과 저장
예약 정보	id, user_id, hotel_id, payment_status	예약 정보 및 결제 내역

### 4.2 Redis 데이터 구조

키	유형	값	비고
schedule:{user_id}	Hash	일정 리스트	
Node:{user_id}	Hash	노드들의 이름 및 경도/위도	node 들의 일반적 형태
distances:{user_id}	hash	거리 행렬	
ga_result:{user_id}	Json	최적화 결과	

## 5. 인터페이스 설계 (Interface Design)

### 5.1 주요 API 목록 (REST API)

기능	함수	방법	설명
회원가입	/auth/register	POST	사용자 계정 생성
로그인	/auth/login	POST	JWT 토큰 발급
일정 저장	/schedule/save	POST	일정 Redis 저장
경로 생성	/ai/generate	POST	유전자 알고리즘 실행
경로 조회	/ai/result	GET	GA 결과 조회
호텔 검색	/amadeus/hotel	GET	호텔 목록 조회
명소 검색	/google/place	GET	관광지 검색

## 6. 보안 및 오류 처리 (Security & Error Handling)

- **JWT 인증:** 모든 API 요청에 Authorization Header 필요
- **API Key 보호:** 파일에 저장 및 Docker 환경 변수로 주입, env 파일 이용
- **오류 처리:** 400(잘못된 요청), 401(인증 실패), 500 (내부 서버 오류), 504 (타임아웃)

## 7. 유지보수 및 확장성 (Maintenance & Scalability)

- **AI 모듈 확장:** AI의 발전에 따라 업데이트된 모델을 사용
- **API 연동:** 외부 숙박/항공/배달 API 연동 확장
- **DB 확장:** Redis → Redis Cluster, MySQL → Read Replica 구성 가능