

<나는 누구? 여긴 어디?>
소프트웨어 아키텍처 문서

Version <1.1>

<Project Name>	Version: <1.1>
Software Architecture Document	Date: <2025.06.09 >
문서 번호 미정	

수정 내역

날짜	버전	설명	작성자
<2025.06.04>	<1.0>	프로젝트 내용 추가	<김형균>
<2025.06.09>	<1.1>	항목별 세부 내용 보충	<김형균>
<2025.06.14>	<1.2>	최종안 작성 및 누락 및 오기입 정정	<김형균>

<Project Name>	Version: <1.1>
Software Architecture Document	Date: <2025.06.09 >
문서 번호 미정	

Table of Contents

1. 서론
 - 1.1 목적
 - 1.2 범위
 - 1.3 용어 정의
 - 1.4 참고 문서
 - 1.5 개요
2. 아키텍처 표현
3. 아키텍처 목표 및 제약 사항
4. 유스케이스 뷰
 - 4.1 Use-Case Realizations
5. 논리뷰
 - 5.1 개요
 - 5.2 아키텍처적으로 중요한 설계 패키지
6. 프로세스 뷰
7. 배포 뷰
8. 구현 뷰
 - 8.1 개요
 - 8.2 계층
9. 데이터 뷰 (선택 사항 I)
10. 크기 및 성능
11. 품질

오류! 책갈피가 정의되어 있지 않습니다.

<Project Name>	Version: <1.1>
Software Architecture Document	Date: <2025.06.09 >
문서 번호 미정	

소프트웨어 아키텍처 문서

1. 서론

1.1 목적

이 앱은 바쁜 일상 속에서도 짧은 여유 시간을 활용해 소규모 여행을 즐기고자 하는 사용자를 위해 개발되었다. 사용자가 원하는 시간대를 입력하면 해당 시간 내 이동이 가능한 여행지를 무작위로 추천하고, 주변 음식점 정보를 함께 제공하며, 경로 안내는 TMAP API 를 통해 지원한다. 본 문서에서는 해당 애플리케이션의 전반적인 아키텍처 구조와 핵심 기능에 대해 기술하였다.

1.2 범위

본 문서는 Android 기반 WebView 앱을 통해 실행되는 React 프론트엔드 구조를 기반으로 작성되었다. 추천 알고리즘, 사용자 입력 흐름, 외부 API 연동 방식 등 앱의 전반적인 흐름과 기능 설계를 포함한다. 데이터베이스 없이 외부 API 만을 이용하는 구조를 중심으로 설명하였다.

1.3 정의, 약어 및 축약어

- WebView
Android 앱 내에서 HTML, CSS, JavaScript 로 구성된 웹 페이지를 표시할 수 있도록 도와주는 컴포넌트이다. 본 프로젝트에서는 React 웹 앱을 Android 앱에 내장하여 실행하는데 사용되었다.
- React
Facebook 에서 개발한 JavaScript 기반의 UI 프레임워크로, 재사용 가능한 컴포넌트를 통해 효율적인 사용자 인터페이스를 개발할 수 있다.
- TMap API
SK 텔레콤에서 제공하는 지도 및 길찾기 API. 출발지와 도착지를 기반으로 경로 안내, 거리 계산, 소요 시간 정보 등을 제공한다.
- Time Slot
사용자가 여행을 갈 수 있는 시간 범위를 의미한다. 이 앱에서는 시간대를 기반으로 추천 가능한 여행지를 필터링하는 데 활용된다.
- 추천 알고리즘
사용자 입력(시간대, 위치 등)을 바탕으로 여행지를 자동으로 선별하는 내부 로직이다. 현재는 무작위(Random) 선택 방식을 사용하고 있다.
- LocalStorage
웹 브라우저 내부에서 데이터를 클라이언트 측에 임시로 저장할 수 있도록 제공하는 저장소. DB 를 사용하지 않는 대신 간단한 정보 저장에 활용된다.
- 컴포넌트
React 기반 UI 요소를 구성하는 단위 기능 블록이다. 예: TimeSelector, RecommendationPage 등.

<Project Name>	Version: <1.1>
Software Architecture Document	Date: <2025.06.09 >
문서 번호 미정	

- 시퀀스 다이어그램
시스템 내 객체들 간의 상호작용을 시간 순서대로 보여주는 UML 다이어그램 유형이다. 본 문서에서는 사용자 입력부터 결과 출력까지의 흐름을 나타내는 데 사용되었다.
- 배포 다이어그램
시스템 구성 요소가 물리적으로 어떻게 배치되어 있는지를 나타내는 UML 다이어그램이다. 클라이언트, 서버, API 간의 상호 배치 구조를 설명할 때 사용되었다.
- 컴포넌트 다이어그램
시스템 내 기능 또는 서비스 모듈 간의 관계를 시각화한 UML 다이어그램이다. 구현 구조를 설명할 때 사용된다.

1.4 참고 문서

- TMAP API 공식 문서
경로 탐색 및 지도 API 연동을 위해 SK 텔레콤에서 제공하는 TMAP Developers 문서를 참고, API 키 발급, 경로 요청 방식, 응답 형식 등에 대한 상세 정보가 포함
URL : <https://tmapapi.sktelecom.com/>
- React 공식 문서
프론트엔드 UI 구현은 React 를 기반으로 하였으며, 공식 문서를 통해 컴포넌트 작성, 상태 관리, 이벤트 처리 등에 대한 개념을 이해하고 적용하였다.
URL : <https://reactjs.org/>
- Android 개발자 문서(WebView)
React 웹 앱을 Android 앱에서 실행하기 위해 WebView 컴포넌트를 사용하였으며, 관련 구현 방법은 Android 공식 개발자 문서를 참고하였다.
URL : <https://developer.android.com/reference/android/webkit/WebView>

<Project Name>	Version: <1.1>
Software Architecture Document	Date: <2025.06.09 >
문서 번호 미정	

1.5 개요

본 문서는 ‘여긴 어디? 나는 누구?’의 소프트웨어 아키텍처를 체계적으로 설명하기 위해 작성되었다.

초기에는 사용자의 입력 흐름과 추천 기능 중심의 유스케이스를 정의, 이후 시스템을 논리적으로 분해하여 각 구성 요소의 역할을 정리하였다.

문서의 구성은 다음과 같다.

- 2 장에서는 아키텍처의 전체 구조와 각 뷰의 개요를 설명하였다.
- 3 장에서는 시스템 설계 시 고려한 주요 목표와 제약 사항을 정리하였다.
- 4 장에서는 사용자 입장에서 시스템이 어떻게 작동하는지를 유스케이스 중심으로 설명하였다.
- 5 장부터 9 장까지는 시스템 내부 구조, 배포 환경, 구현 계층, 데이터 구조 등을 다양한 관점에서 기술하였다.
- 10 장과 11 장에서는 시스템의 성능 목표와 품질 특성에 대해 서술하였다.

이를 통해 시스템의 구조와 기능을 명확히 이해하고, 유지보수와 확장 시 참고할 수 있는 기반 자료로 사용할 수 있도록 하였다.

<Project Name>	Version: <1.1>
Software Architecture Document	Date: <2025.06.09 >
문서 번호 미정	

2. 아키텍처 표현

본 시스템은 다양한 소프트웨어 아키텍처 뷰로 구성되어 있으며, 이를 통해 전체적인 기능 흐름과 구조를 효과적으로 설명하고자 했다. 각 뷰는 시스템의 다른 측면을 표현하며, 주요 구성은 다음과 같다.

- **유스케이스 뷰**에서는 사용자와 시스템 간의 주요 상호작용을 중심으로 기능 흐름을 정리하였다.
- **논리 뷰**에서는 UI, 로직, API 처리 등 각 모듈의 역할과 상호관계를 기술하였다.
- **프로세스 뷰**에서는 입력부터 출력까지의 흐름을 비동기 처리 중심으로 표현하였다.
- **배포 뷰**에서는 Android 기기, WebView, 외부 API 서버 간의 실제 배포 환경을 설명하였다.
- **구현 뷰**에서는 계층화된 코드 구조와 각 레이어 간의 책임을 명확히 구분하였다.
- 이러한 아키텍처 표현을 통해 시스템의 기능적 요구사항은 물론, 유지보수성과 확장성 측면에서도 구조적인 명확성을 확보하고자 했다.

3. 아키텍처 목표 및 제약 사항

시스템 아키텍처 설계의 주요 목표는 다음과 같았다.

- 첫째, 직관적이고 간편한 사용자 경험을 제공하는 것이었다. 사용자는 시간대만 입력하면 여행지 추천부터 음식점 탐색, 경로 안내까지 모든 과정을 자동으로 받을 수 있어야 하므로, 인터페이스는 React 기반으로 직관적이고 반응성 있게 구성하였다.
- 둘째, 외부 API 활용의 효율성을 확보하는 것이었다. 지도 및 경로 기능은 TMAP API, 음식점 정보는 T Map 지역 API 를 통해 실시간으로 불러오며, 각각은 비동기 처리로 UI 응답성을 저하시키지 않도록 설계하였다.
- 셋째, 유지보수성과 기능 확장성을 고려하였다. 기능별 로직을 컴포넌트화하고, API 호출 모듈을 분리함으로써 기능이 추가되거나 교체되는 경우에도 전체 시스템에 미치는 영향을 최소화할 수 있도록 하였다.

이와 함께 다음과 같은 제약 사항도 고려해야 했다.

- 앱은 외부 API 에 의존하므로, API 호출 제한, 사용량 초과, 정책 변경 등에 따라 일부 기능이 제한될 수 있다.
- DB 없이 동작하도록 설계되어 있어, 사용자 데이터 저장이나 분석 등의 기능은 제공하지 않는다.
- WebView 기반 구조로 인해, 저사양 기기에서는 성능 저하가 발생할 가능성이 있다.

이러한 목표와 제약 사항을 종합적으로 고려하여, 본 시스템은 기능성과 실현 가능성의 균형을 맞춘 구조로 설계되었다.

<Project Name>	Version: <1.1>
Software Architecture Document	Date: <2025.06.09 >
문서 번호 미정	

4. 유스케이스 뷰

- UC-01: 시간대 입력

사용자가 여행 가능한 시간대를 설정하는 기능이다. 해당 입력은 추천 알고리즘의 필터링 기준으로 활용된다.

- UC-02: 여행지 추천

입력된 시간대를 기반으로 무작위로 적절한 여행지를 추천한다. 이 기능은 사용자의 현재 위치와 예상 소요 시간을 고려하여 필터링된 리스트에서 장소를 선정한다.

- UC-03: 음식점 검색

추천된 여행지 주변의 음식점을 검색하고, 외부 API 를 통해 받아온 정보를 기반으로 사용자에게 제공한다.

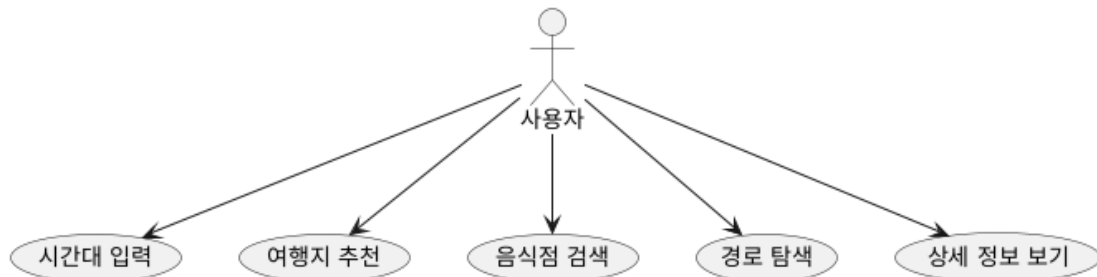
- UC-04: 경로 탐색

사용자의 현재 위치에서 추천된 여행지까지의 이동 경로를 TMAP API 를 통해 안내한다. 자동차 또는 도보 경로 선택이 가능하며, 소요 시간과 거리도 함께 제공된다.

- UC-05: 상세 정보 보기

추천된 장소나 음식점에 대한 설명, 위치, 사진, 후기 등의 상세 정보를 사용자에게 제공한다.

4.1 유스케이스 실현



<Project Name>	Version: <1.1>
Software Architecture Document	Date: <2025.06.09 >
문서 번호 미정	

5. 논리 뷰

본 앱은 React 기반 프론트엔드를 중심으로 구성되어 있으며, 주요 기능은 UI, 추천 로직, API 연동 세 영역으로 나뉘었다. 사용자의 입력은 React 컴포넌트에서 처리되며, 로직 컴포넌트에서는 입력된 시간대와 위치 정보를 바탕으로 추천 기능이 수행된다. 외부 API 호출은 별도의 서비스 모듈에서 관리하며, 이 모듈은 음식점 검색과 경로 탐색 요청을 담당한다.

5.1 개요

본 시스템은 기능별로 역할을 분리하여 설계되었으며, 각 기능은 논리적으로 독립된 구성요소로 구분하였다. 전체 구조는 사용자 인터페이스를 담당하는 프론트엔드(UI), 추천 기능을 처리하는 로직 레이어, 외부 API 와 통신하는 서비스 모듈로 나누어 구성하였다.

사용자는 React 로 구현된 웹 기반 화면을 통해 시간대를 입력하고 결과를 확인하게 되며, 이 과정에서 발생하는 이벤트는 내부 로직 모듈로 전달된다. 로직 모듈에서는 입력된 시간대를 기준으로 추천 대상지를 선별하고, 외부 API 를 호출해 음식점 정보와 경로 정보를 요청한다.

각 영역은 다음과 같이 정의하였다.

- UI 계층
사용자의 입력을 수집하고, 추천 결과와 장소 정보를 시각적으로 출력하는 역할을 수행했다.
- 로직 계층
시간 필터링 및 랜덤 추천 알고리즘을 통해 여행지 후보를 결정하는 기능을 담당했다.
- API 계층
TMAP API 및 장소 검색 API 와의 연동을 통해 외부 정보를 수신하는 기능을 수행했다.

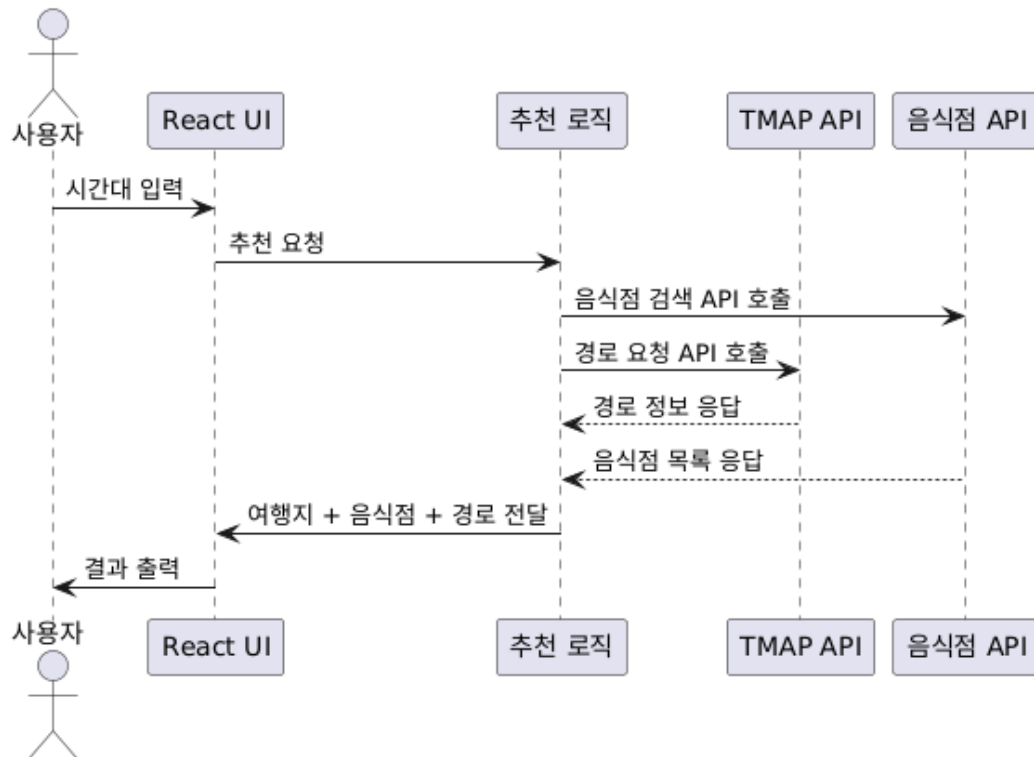
각 레이어는 상태(state) 기반 설계를 통해 명확히 구분되며, 모듈 간 결합도를 낮추어 유지보수와 기능 확장을 고려한 구조로 설계하였다.

5.2 아키텍처적으로 중요한 설계 패키지

- UI 컴포넌트
사용자 입력을 처리하는 TimeSelector, 추천 결과를 보여주는 RecommendationView, 상세 정보를 표시하는 DetailView 등으로 구성되어 있다.
- 추천 로직
입력된 시간대에 따라 소요 시간을 계산하고, 그에 적합한 장소를 랜덤하게 선택하는 TimeFilterService 와 RandomSelector 로 구성된다.
- API 연동 모듈
TMAP 경로 요청을 처리하는 TmapService, 음식점 검색을 처리하는 PlaceService 등이 포함되어 있으며, 비동기 요청을 통해 외부 데이터와 통신한다.

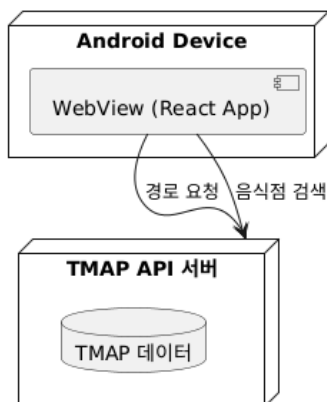
<Project Name>	Version: <1.1>
Software Architecture Document	Date: <2025.06.09 >
문서 번호 미정	

6. 프로세스 뷰



사용자가 앱을 실행하고 시간대를 입력하면, 해당 정보를 기반으로 추천 알고리즘이 작동하여 장소를 선정한다. 이후 장소에 대한 상세 정보와 음식점 정보가 외부 API를 통해 불러와진다. TMAP API를 호출하여 현재 위치에서 장소까지의 경로를 받아오고, 최종적으로 모든 정보가 통합되어 사용자에게 표시된다. 이러한 프로세스는 React 내의 컴포넌트 기반 구조에서 상태 변화와 이벤트 흐름으로 처리된다.

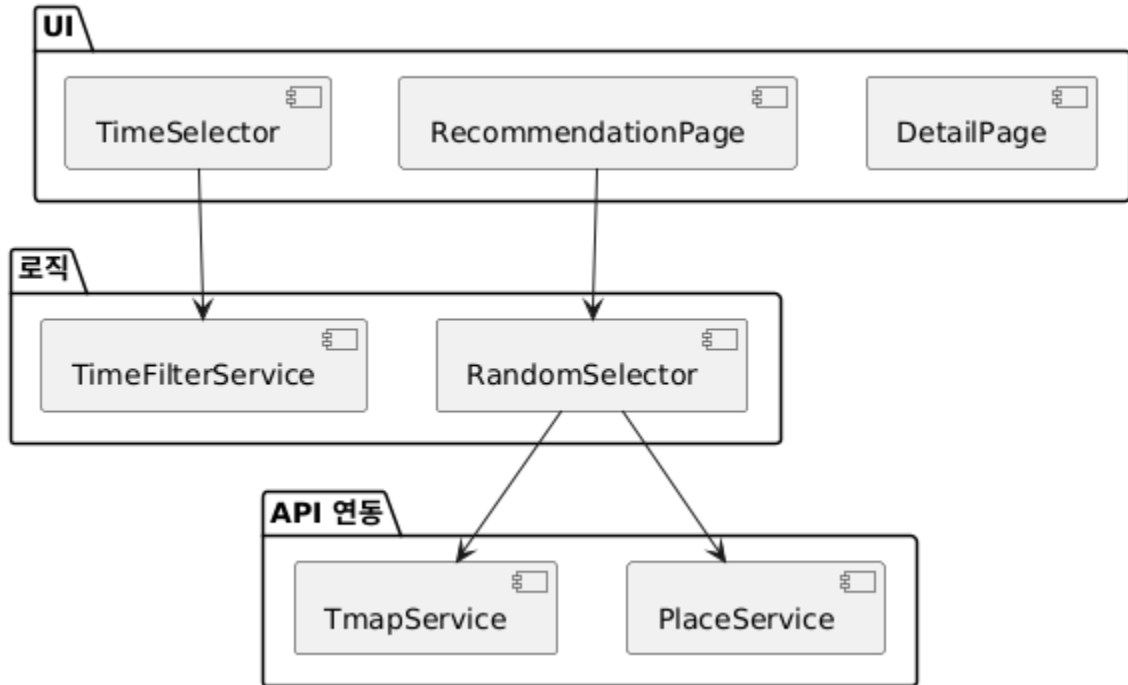
7. 배포 뷰



앱은 Android Studio를 기반으로 개발되었으며, WebView를 통해 React 웹 애플리케이션이 로드되는 구조로 되어 있다. 모든 주요 연산은 클라이언트에서 이루어지며, 외부 API와의 통신은 HTTPS 요청으로 처리된다. 물리적인 배포 환경은 Android 디바이스와 외부 API 서버들(TMAP 등)로 구성된다.

<Project Name>	Version: <1.1>
Software Architecture Document	Date: <2025.06.09 >
문서 번호 미정	

8. 구현 뷰



소프트웨어는 크게 세 개의 레이어로 나뉘어 있다. 프레젠테이션 레이어에서는 사용자의 입력을 받고 결과를 시각적으로 출력한다. 로직 레이어에서는 입력된 데이터에 기반한 처리 및 추천 알고리즘이 수행된다. API 연동 레이어에서는 외부 시스템과의 통신이 이루어진다. 각 레이어는 기능적으로 분리되어 있으며 유지보수에 용이한 구조를 가진다.

9. 크기 및 성능

전체 애플리케이션의 예상 사용자 수는 1,000 명 이하이며, 초기 버전에서는 API 호출 속도와 렌더링 성능이 핵심 이슈가 될 것으로 예상되었다. 장소 추천과 경로 탐색 기능은 평균 1~2 초 내에 응답이 완료되도록 설계되었으며, 지도와 이미지가 포함된 화면에서는 WebView의 성능 저하가 발생할 가능성이 있다.

10. 품질

- 확장성: 장소 DB와 연동하거나 조건 필터를 추가함으로써 추천 기능을 확장할 수 있다.
- 신뢰성: 외부 API 호출이 실패할 경우를 대비하여 사용자에게 안내 메시지를 제공하도록 설계되었다.
- 이식성: React 기반 UI 구조를 통해 iOS 플랫폼으로의 전환이 비교적 수월하다.
- 보안성: API 키는 앱 외부에서 노출되지 않도록 환경 변수 또는 프록시 서버를 통해 관리할 수 있다.