



AI 기반 맞춤형 여행 추천 플랫폼

“TravelGenie”

팀명: 일조(1조)

팀원: 김정현 김정민 공수민 박상아

트립어드바이저 스타일의 AI 여행 추천 및 일정 생성 플랫폼



2025년 11월 25일

목차

- 1 프로젝트 개요
- 2 요구사항 분석
- 3 시스템 설계
- 데이터베이스 설계
- 5 UI/UX 설계
- 6 사용자 흐름
- 7 프로젝트 진행 현황
- 8 핵심 기능 시연 (1)
- 9 핵심 기능 시연 (2)
- 10 기술 구현
- 성과 및 기대효과
- 12 향후 계획
- 13 결론 & Q&A

팀원 역할

기획/PM 공수민 20251248

- 프로젝트 범위 정의 및 로드맵 수립
- KPI 설정 및 측정 계획
- 외부 파트너십 관리
- 리스크 및 데이터 거버넌스 관리

데이터 공수민,김정현

- 여행 데이터 수집 및 정제
- 피처 스토어 구축 및 품질 관리
- 데이터 카탈로그 및 메타데이터 관리
- 데이터 파이프라인 구축 및 모니터링

모델링 김정민 20251233

- 추천 알고리즘 선택 및 학습
- 모델 평가 및 재현성 확보
- 온라인 서빙 시스템 구축
- 모델 성능 모니터링 및 개선

발표/운영 박상아 20252347

- 프로젝트 데모 및 발표 자료 작성
- 기술 문서화 및 API 가이드 작성
- CS 대응 및 분석 대시보드 구축
- 사용자 교육 및 온보딩 지원

UI/UX 김정현 20252658

- 대화형 여행 플래너 인터페이스 설계
- 지도/동선 시각화 UX 개발
- 접근성 및 다국어 지원
- 사용자 테스트 및 피드백 반영

1.프로젝트 개요



배경 및 필요성

- 여행 정보의 과잉으로 인한 **정보 피로** 현상 발생
- 기존 플랫폼의 **개인화된 추천** 부족
- LLM 기술의 성숙**으로 대화형 서비스 구현 가능성 증가





2025년 현재 국내 여행 시장은 연간 7천만 명 이상이 국내외 여행 이용 ”



목표

- 사용자 취향, 예산, 일정 **기반 맞춤형 AI 추천 시스템** 구축
- 자연어 대화**를 통한 직관적인 여행 일정 수립
- 리뷰 감성 분석**을 통한 정확한 장소 평가 제공
- 여행 계획 수립 시간 **70% 단축** 목표



-  **ChatGPT API (GPT-4)** - 자연어 처리 및 여행 상담
-  **프롬프트 엔지니어링** - Role-based 설계, Few-shot Learning
-  **Google Maps API** - 위치 정보, 경로 탐색
-  **감성 분석** - 리뷰 평가 및 시각화

2.요구사항 분석







기능 요구사항

-  **AI 챗봇**
자연어 기반 여행 추천 및 상담
-  **여행 일정 생성**
AI 자동 및 수동 일정 생성/편집
-  **리뷰 및 평점**
감성 분석 및 시각화
-  **검색 및 필터링**
다양한 조건별 장소 검색
-  **커뮤니티**
여행 후기 공유 및 Q&A

비기능 요구사항

-  **성능**
API 응답 3초 이내, 페이지 로딩 2초 이내
-  **안정성**
시스템 가용성 99.5%, 자동 백업
-  **보안성**
HTTPS, 개인정보 암호화(AES-256)
-  **확장성**
마이크로서비스, API 버전 관리
-  **접근성**
WCAG 2.1 표준, 반응형 디자인

타겟 사용자

-  **계획형 여행자**
정민아 (29세)
철저한 계획 선호, 맛집·카페 관심
-  **즉흥형 여행자**
김태양 (25세)
유연한 일정, SNS 핫플레이스 추구
-  **가족 여행자**
박지현 (38세)
아이 동반, 안전성과 편의시설 중시
-  **시니어 여행자**
이영수 (62세)
여유로운 일정, 접근성과 문화 중시

3.시스템 설계 - 아키텍처 & 스택

클라이언트 레이어



웹 브라우저



모바일 브라우저

프레젠테이션 레이어



프론트엔드 서버 (Nginx)



CDN (정적 파일)

애플리케이션 레이어



API Gateway



인증 서비스



추천 서비스



검색 서비스



리뷰 서비스



일정 서비스

데이터 레이어



MySQL
사용자, 장소, 리뷰



Redis
세션, 캐시



MongoDB
대화, 로그

외부 연동 레이어



OpenAI API



Google Maps API



공공데이터 API

기술 스택

프론트엔드



React.js 18.2
+ TypeScript



Tailwind CSS
스타일링

백엔드



Node.js 20.x
Express.js



Python 3.11
AI 추천 서비스

데이터베이스



MySQL 8.0
사용자/장소 데이터



Redis 7.0
캐시, 세션 관리

배포 환경

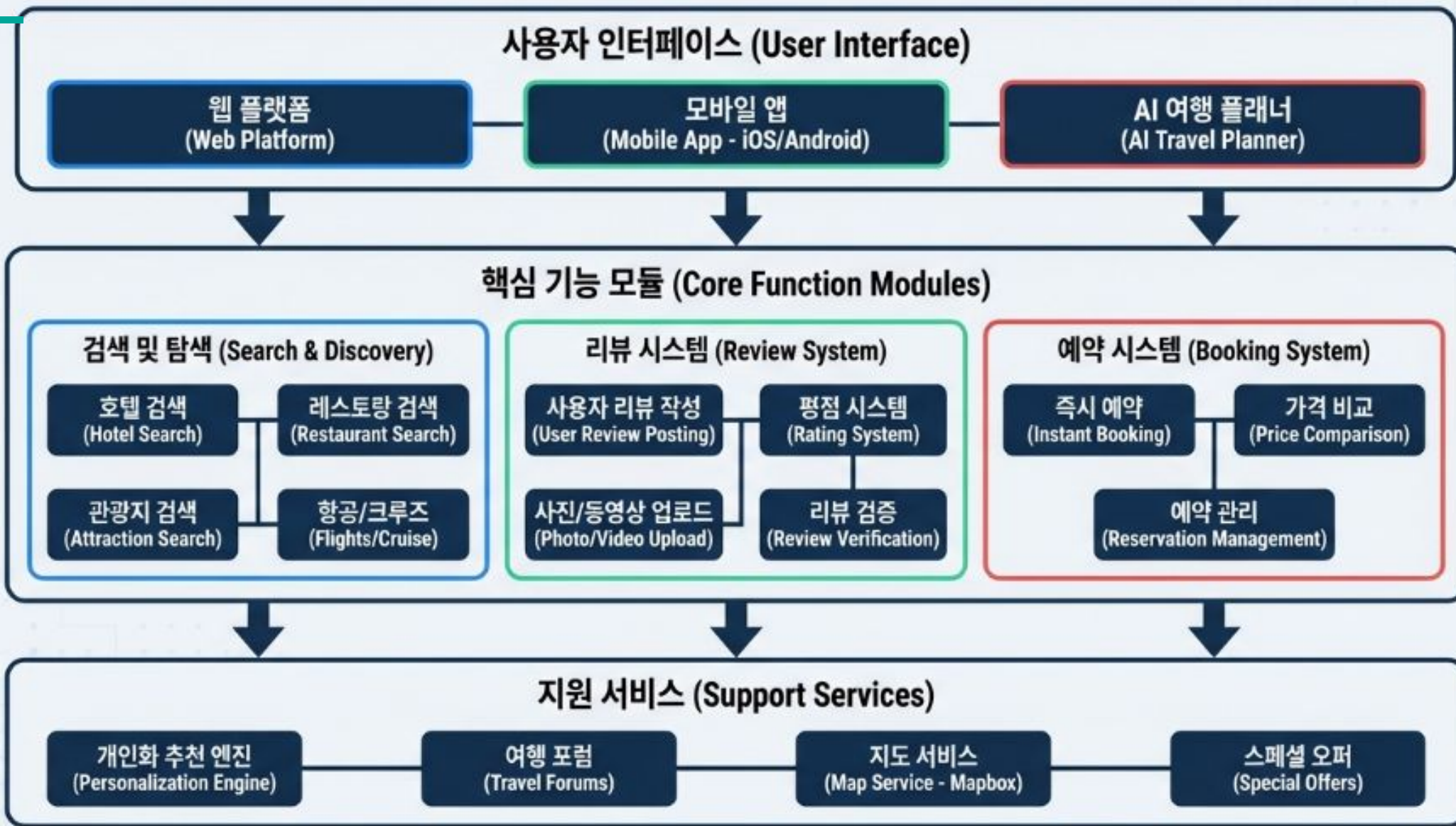


AWS
EC2, RDS, S3

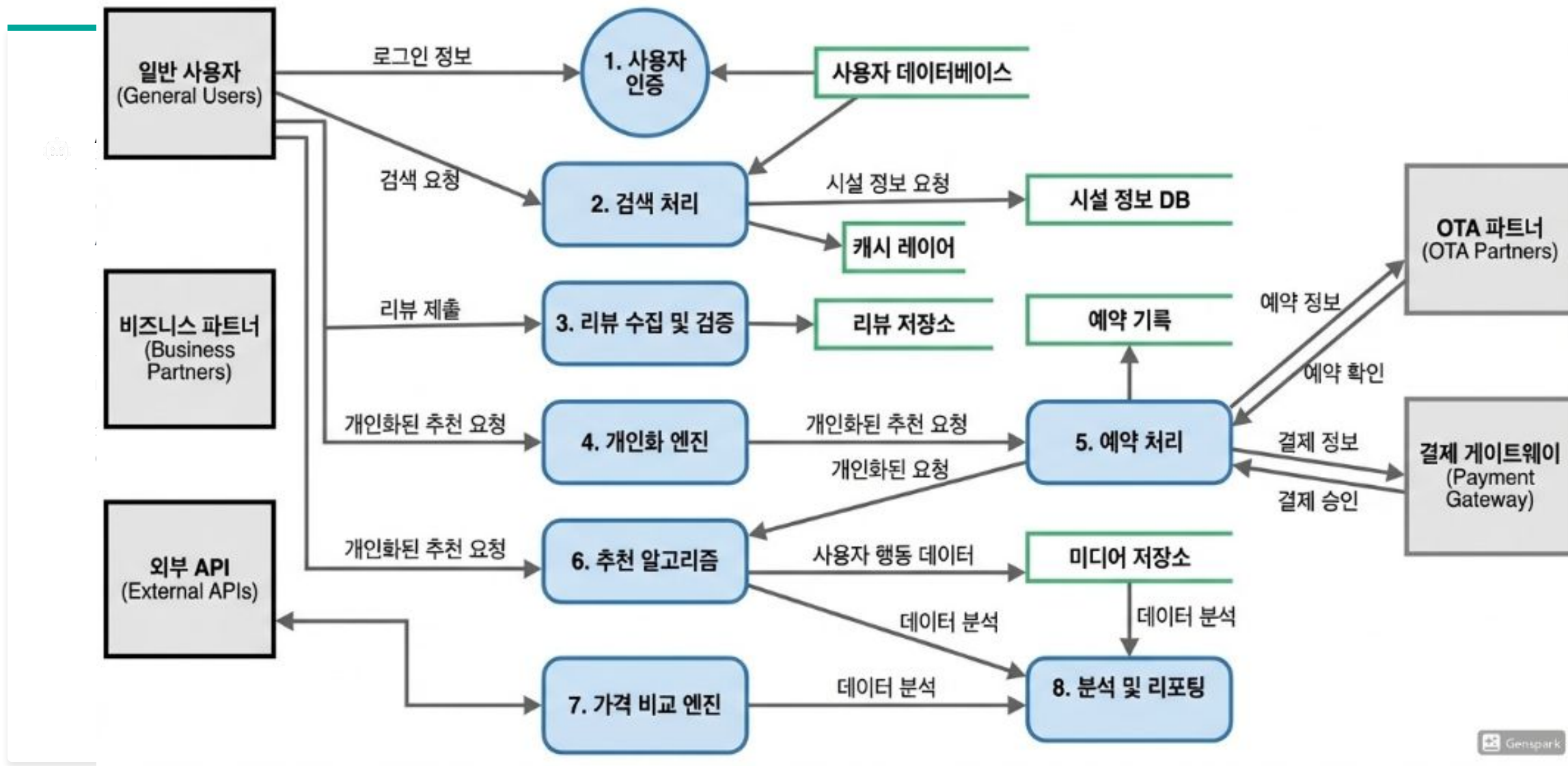


Docker
컨테이너화

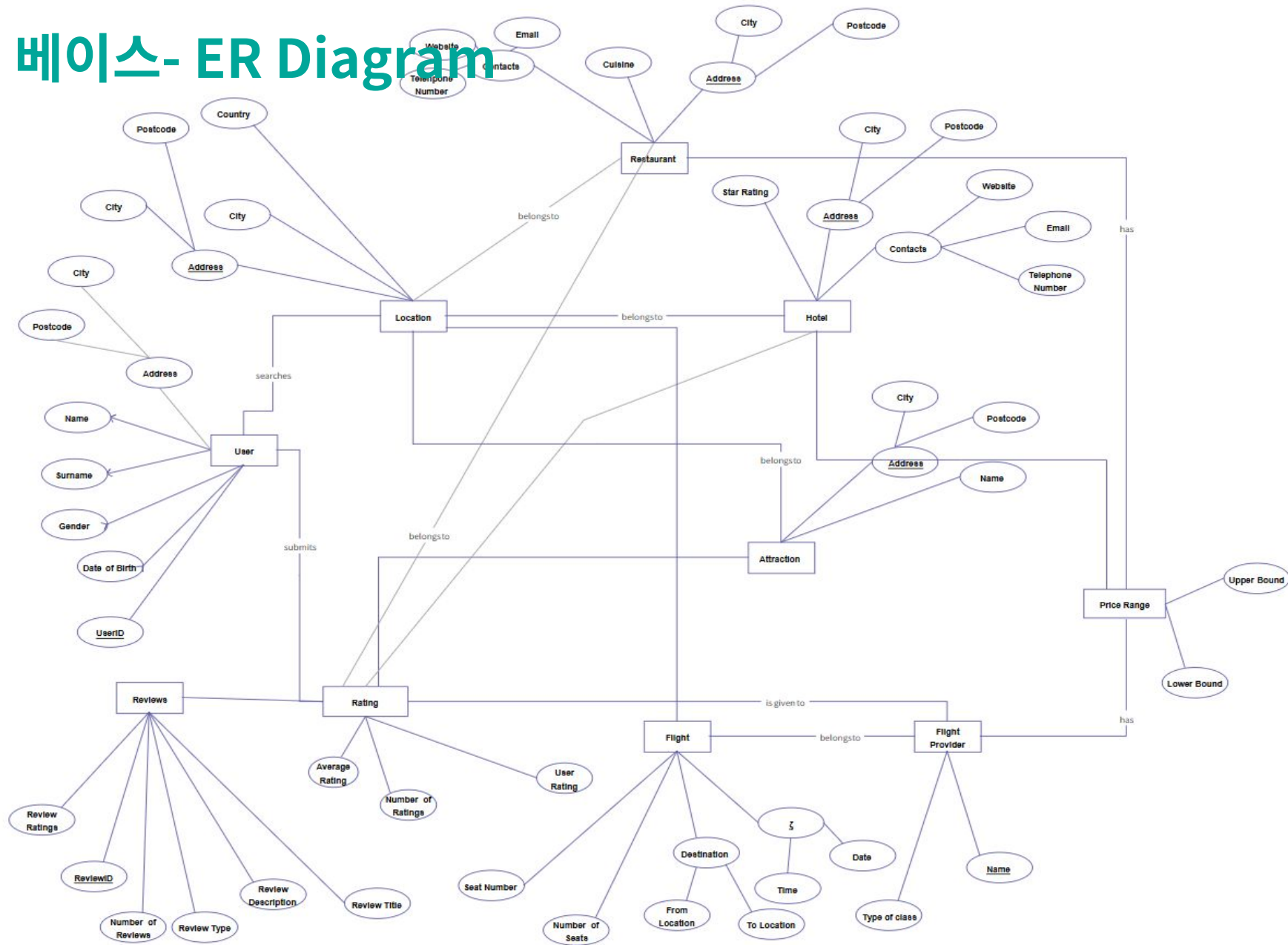
기능 구조 설계- 기능 블록도



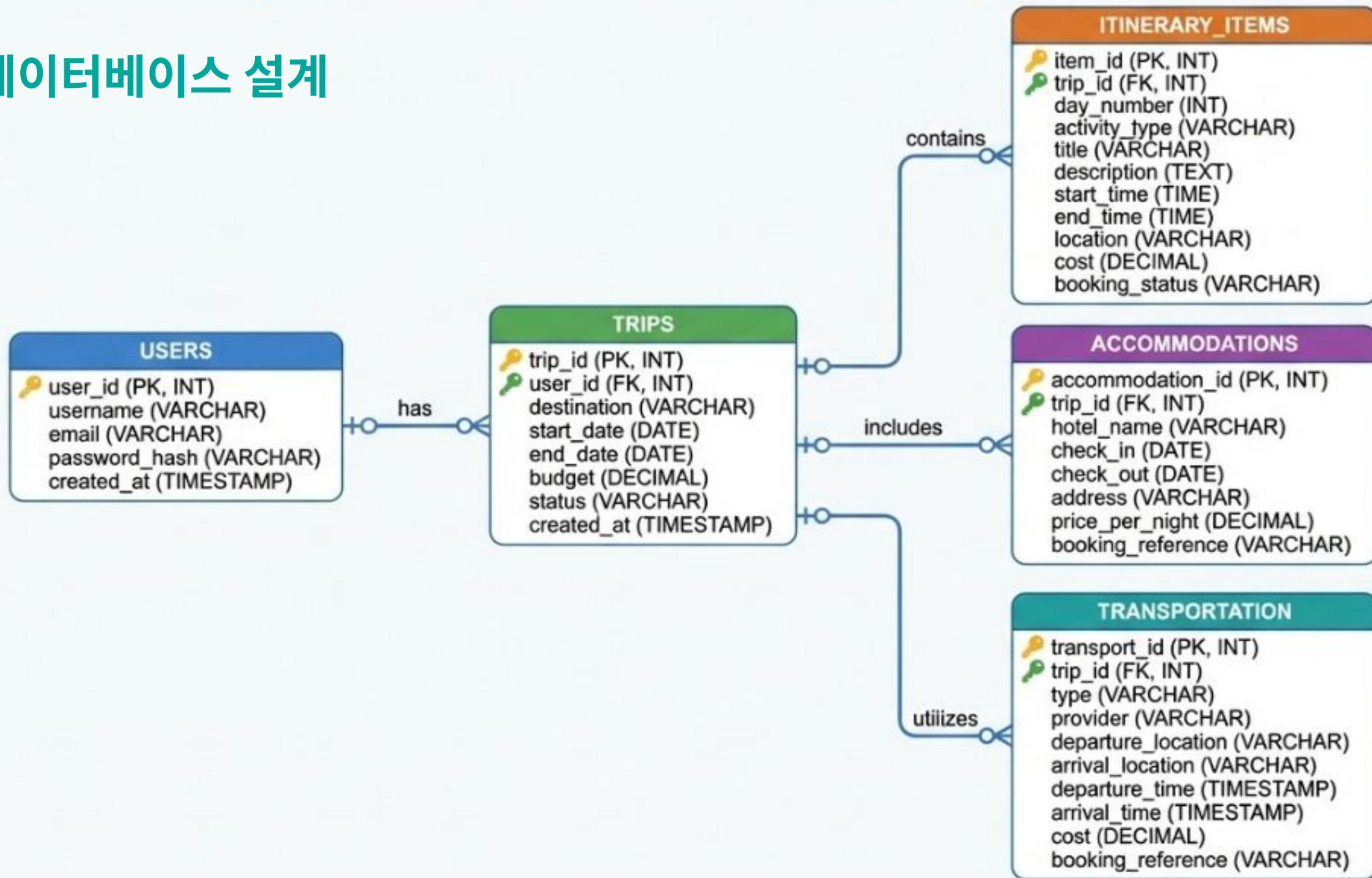
데이터 흐름도



데이터 베이스- ER Diagram



데이터베이스 설계



API 및 외부 연동 구조



4.UI/UX 설계

💡 설계 철학

직관성 (Intuition)

처음 사용하는 사용자도 3분 내 핵심 기능 사용 가능한 직관적 인터페이스

효율성 (Efficiency)

최소 클릭으로 원하는 정보에 접근하는 효율적 동선 설계

일관성 (Consistency)

트립어드바이저와 유사한 레이아웃으로 학습 곡선 최소화

접근성 (Accessibility)

모든 연령대가 사용 가능한 명확한 인터페이스 및 WCAG 표준 준수

🏗️ 디자인 원칙



모바일 퍼스트

70%+ 모바일 사용자
고려한 반응형 설계



AI 중심

챗봇을 주요 인터랙션
포인트로 설정



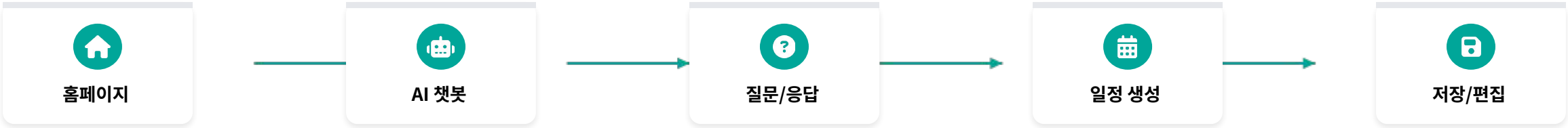
시각적 계층

중요도에 따른 정보
우선순위 표현



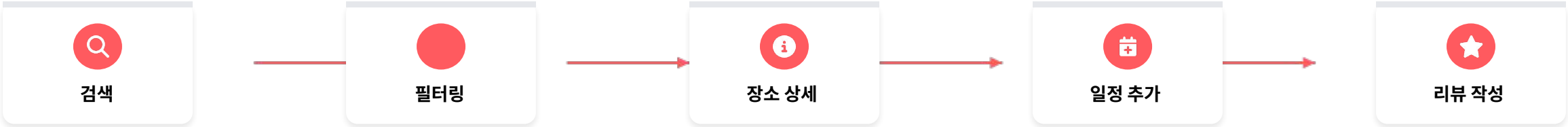
사용자 흐름 (주요 시나리오)

🏠 시나리오 1: AI 추천 기반 여행 계획



사용자가 홈페이지에서 챗봇을 통해 여행 요구사항을 입력하면 AI가 맞춤형 질문을 하고, 응답을 기반으로 일정을 자동 생성하여 저장하거나 편집할 수 있습니다.

🔍 시나리오 2: 검색 탐색 및 리뷰 작성



사용자가 특정 지역이나 카테고리 검색 후, 필터링을 통해 원하는 장소를 찾아 상세 정보를 확인하고, 일정에 추가하거나 방문 후 리뷰를 작성할 수 있습니다.

✍️ 시나리오 3: 수동 일정 생성



사용자가 직접 새 여행 일정을 만들고, 원하는 장소를 추가한 후, 최적의 경로를 확인하고 친구나 가족과 일정을 공유할 수 있습니다.

타킷 사용자 페르소나: 계획형 여행자



김민아

35세, 회사원

여행 스타일 태그

#계획형 #효율성 #검증된장소

#인스타감성 #맛집탐방



배경

연차를 모아 1년에 한 번 장기 여행(10일 내외)을 떠납니다. 여행을 떠나기 몇 달 전부터 모든 것을 미리 예약하고 계획하는 것을 선호합니다.



여행 목적

효율적이고 만족도 높은 여행. 숨겨진 보석 같은 장소보다 검증된 인기 명소와 맛집을 놓치지 않는 것이 중요합니다.



트립어드바이저 사용 목표

최다 리뷰와 평점 4.5 이상의 호텔 및 액티비티를 찾고, 가장 최근의 상세 리뷰를 읽어 현지 상황을 파악합니다. 여행 일정을 짜기 위해 저장 기능을 적극적으로 활용합니다.



주요 활용 기능



필터링 기능 (평점, 가격, 거리)



'트립 저장' 기능



Q&A 섹션



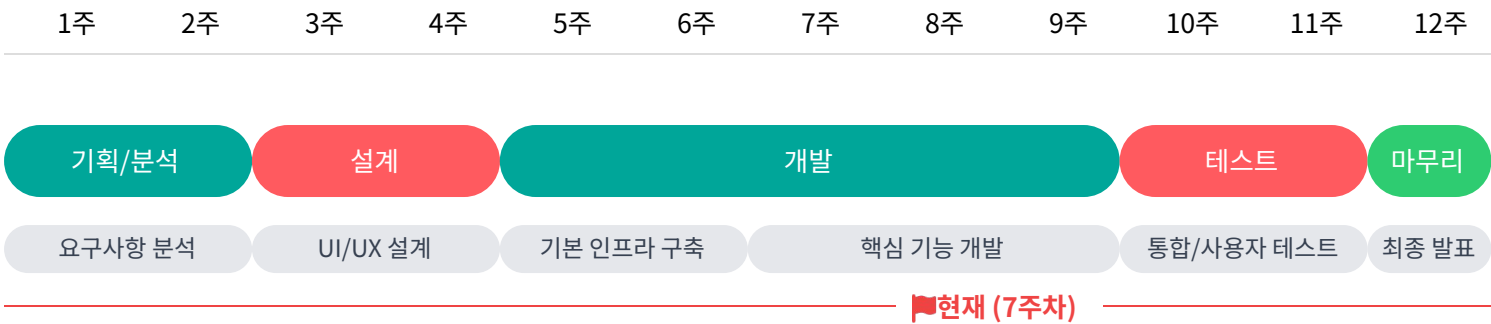
전문가 리뷰



핵심 니즈

모든 정보를 한 곳에서 비교하고 확신을 얻은 후 예약하고 싶음.

프로젝트 진행 현황 (12주)



● 기획/개발 ● 설계/테스트 ● 마무리

주요 마일스톤

● 기획/요구사항 분석 완료 ● 시스템/UI 설계 완료 ● 핵심 기능 개발 진행중 ● 테스트 및 발표 준비

예정된 기능

- 커뮤니티 게시판
- 팔로우 시스템
- 알림 기능

완료된 기능 (100%)

- 사용자 인증
회원가입, 로그인, JWT 인증
- 장소 데이터베이스
500개 주요 관광지 데이터
- 검색 기능
키워드, 카테고리별 검색
- 홈페이지 UI
반응형 메인 화면 구현

진행 중인 기능

- AI 챗봇
85% - 컨텍스트 관리 개발 중
- 리뷰 시스템
90% - 감성 분석 API 연동 중
- 여행 일정 생성
75% - AI 자동 생성 개발 중
- 지도 연동
70% - 경로 탐색 기능 개발 중

핵심 기능 시연 (1) – 장소 상세 페이지

핵심 UI 요소

이미지 갤러리

슬라이더 형식의 이미지 갤러리, 확대 기능 및 썸네일 탐색 제공

평점 및 카테고리

방문자 평가 기반 5점 척도 평점, 시설 유형 카테고리 표시

기본 정보

주소, 전화번호, 운영시간, 입장료 등 필수 방문 정보 제공

AI 추천 포인트

방문객 유형별 맞춤형 AI 추천 및 방문 팁 제공

리뷰 감성 분석

방문자 리뷰 텍스트 기반 긍정/부정 비율 시각화


자주 언급되는 키워드

리뷰에서 추출한 주요 키워드 태그 클라우드 제공

차별화 포인트


트립어드바이저 스타일의 UI에 AI 기반 추천과 감성 분석을 더해 개인화된 여행 정보 제공

장소 상세 페이지 (예시)







Tours Destination Virtual tour Blog About Contact English Wishlist

Venice, Rome & Milan

★ 4.8 (256 reviews) Queenstown, Otago, New Zealand



Show all photos



Overview

The secret journey of Tonkin is designed for travelers who want to touch, taste and feel the souls of Northern Vietnam through their unique highlights and experiences. More than simple whirlwind tours, the secret journey of Tonkin of Tonkin marry iconic destinations and must-see spots with the hidden corners and below-the-skin experiences to discover some of the famous and unique places around the Halong region.

Show more

Included/Excluded

✓ Specialized bilingual guide

✗ Departure Taxes

\$119 \$102 /person

★ 4.8 (256 reviews)

Date

May 15, 2024

Guests

2 guests

CHECK IN

May 15, 2024

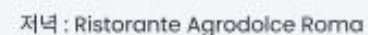
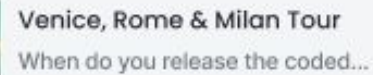
CHECK OUT

May 18, 2024

ACTIVITY



 Search message



```
# 설치 (Colab / Jupyter 환경에서 실행)
!pip uninstall -y numpy
!pip install numpy==1.26.4
!pip install scikit-surprise

# --- 코드 시작 ---
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel
from surprise import Reader, Dataset, SVD
from surprise.model_selection import train_test_split
from collections import defaultdict

# 예시 데이터
business_df = pd.DataFrame({
    'item_id': [1, 2, 3, 4, 5],
    'name': ['맛집 A', '호텔 B', '카페 C', '맛집 D', '명소 E'],
    'category': ['음식점', '숙소', '카페', '음식점', '관광'],
    'description': ['파스타 맛있는 곳', '전망 좋은 호텔', '조용하고 아늑한 카페', '가성비 좋은 고기집', '가족 나들이 명소']
})

rating_df = pd.DataFrame({
    'user_id': [101, 101, 102, 103, 103, 101],
    'item_id': [1, 3, 1, 2, 4, 5],
    'rating': [5.0, 3.0, 4.0, 5.0, 4.0, 2.0]
})

# 1. 콘텐츠 기반 추천 함수
def get_content_recommendations(target_item_id, item_df=business_df, top_n=3):
    tfidf = TfidfVectorizer(stop_words=None) # 한국어는 형태소 분석이 좋지만 예제라 간단히
    item_df['description'] = item_df['description'].fillna('')
    tfidf_matrix = tfidf.fit_transform(item_df['description'])

    cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)
    indices = pd.Series(item_df.index, index=item_df['item_id']).drop_duplicates()

    idx = indices[target_item_id]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:top_n+1] # 자기 자신 제외
    item_indices = [i[0] for i in sim_scores]

    result = item_df.iloc[item_indices].copy()
    result['similarity'] = [i[1] for i in sim_scores]
    return result[['item_id', 'name', 'category', 'similarity']]
```

```
# 2. 협업 필터링 - SVD 모델 학습 및 예측 함수
def train_svd_and_predict(rating_df, n_factors=50, n_epochs=20):
    reader = Reader(rating_scale=(1, 5))
    data = Dataset.load_from_df(rating_df[['user_id', 'item_id', 'rating']], reader)
    trainset = data.build_full_trainset() # 전체 데이터를 학습에 사용

    algo = SVD(n_factors=n_factors, n_epochs=n_epochs, random_state=42)
    algo.fit(trainset)

    return algo, trainset

# 특정 사용자에게 아직 평가하지 않은 모든 아이템에 대한 예측 평점 반환
def get_collaborative_recommendations(target_user_id, algo, trainset, item_df=business_df, top_n=3):
    # 사용자가 아직 평가하지 않은 아이템 목록
    already_rated = rating_df[rating_df['user_id'] == target_user_id]['item_id'].tolist()
    candidates = item_df[~item_df['item_id'].isin(already_rated)]['item_id'].tolist()

    predictions = []
    for item_id in candidates:
        pred = algo.predict(uid=str(target_user_id), iid=str(item_id))
        predictions.append((item_id, pred.est))

    predictions = sorted(predictions, key=lambda x: x[1], reverse=True)[:top_n]

    result = item_df[item_df['item_id'].isin([p[0] for p in predictions])].copy()
    result = result.merge(pd.DataFrame(predictions, columns=['item_id', 'predicted_rating']), on='item_id')
    return result[['item_id', 'name', 'category', 'predicted_rating']].sort_values('predicted_rating', ascending=False)

# 3. 하이브리드 추천 (간단한 가중 평균 방식)
def get_hybrid_recommendations(target_user_id, target_item_id_for_content=None, top_n=3):
    # 1) 콘텐츠 기반 (특정 아이템과 유사한 아이템)
    if target_item_id_for_content:
        content_res = get_content_recommendations(target_item_id_for_content, top_n=10)
    else:
        content_res = pd.DataFrame()

    # 2) 협업 필터링
    algo, trainset = train_svd_and_predict(rating_df)
    cf_res = get_collaborative_recommendations(target_user_id, algo, trainset, top_n=10)

    # 아이템 ID 기준으로 병합 후 가중 평균
    if not content_res.empty:
        merged = cf_res.merge(content_res[['item_id', 'similarity']], on='item_id', how='left')
        merged['similarity'] = merged['similarity'].fillna(0)
        # 가중치 조정 가능 (예: 0.6 * CF + 0.4 * Content)
        merged['hybrid_score'] = 0.6 * merged['predicted_rating'] + 0.4 * merged['similarity'] * 5 # similarity는 0~10이므로 5를 곱해 스케일 맞춤
        merged = merged.sort_values('hybrid_score', ascending=False)
        return merged[['item_id', 'name', 'category', 'predicted_rating', 'similarity', 'hybrid_score']].head(top_n)
    else:
        return cf_res.head(top_n)
```

기술 구현

```
# =====  
# 실제 실행 예시  
# =====
```

```
print("=== 1. 콘텐츠 기반 추천 (item_id=1 '맛집 A'와 유사한 곳) ===")  
print(get_content_recommendations(1))
```

```
print("\n=== 2. 협업 필터링 추천 (user_id=101에게 추천) ===")  
algo, trainset = train_svd_and_predict(rating_df)  
print(get_collaborative_recommendations(101, algo, trainset))
```

```
print("\n=== 3. 하이브리드 추천 (user_id=101 + 콘텐츠 기준 item_id=1) ===")  
print(get_hybrid_recommendations(target_user_id=101, target_item_id_for_content=1))
```

Requirement already satisfied: numpy==1.16.0 in /usr/local/lib/python3.7/site-packages (from scikits-learn==0.19.0)

=== 1. 콘텐츠 기반 추천 (item_id=1 '맛집 A'와 유사한 곳) ===

	item_id	name	category	similarity
1	2	호텔 B	숙소	0.0
2	3	카페 C	카페	0.0
3	4	맛집 D	음식점	0.0

=== 2. 협업 필터링 추천 (user_id=101에게 추천) ===

	item_id	name	category	predicted_rating
0	2	호텔 B	숙소	3.833333
1	4	맛집 D	음식점	3.833333

=== 3. 하이브리드 추천 (user_id=101 + 콘텐츠 기준 item_id=1) ===

	item_id	name	category	predicted_rating	similarity	hybrid_score
0	2	호텔 B	숙소	3.833333	0.0	2.3
1	4	맛집 D	음식점	3.833333	0.0	2.3

성과 및 기대효과

🏆 현재 성과



500

장소 데이터

서울, 부산, 제주 주요 관광지, 맛집, 카페 정보



1,200

리뷰 데이터

사용자 리뷰 및 평점, AI 감성 분석 완료



50

베타 테스터

서비스 테스트 및 피드백 수집 중

🚀 기대 효과



여행 계획 시간

기존 대비 시간 절약

70% 단축



추천 정확도

사용자 만족도 기반

85% 이상



월간 활성 사용자

서비스 안정화 후

1만 명 목표

🎯 차별화 포인트



향후 계획 (로드맵)



단기 계획 (3~6개월)

프로젝트 안정화 및 고도화



알고리즘 고도화

- 추천 알고리즘 정확도 향상
- 자연어 처리 능력 개선
- 맞춤형 여행 계획 최적화

1~3개월 내



장기 계획 (1년+)

서비스 확장 및 기능 고도화



캐싱/성능 최적화

- Redis 캐싱 전략 강화
- API 응답 시간 단축
- 이미지 최적화 및 CDN 활용

2~4개월 내



사용자 피드백 반영

- 베타 테스터 피드백 분석
- UI/UX 개선사항 적용
- 버그 수정 및 안정화

3~6개월 내



AR 관광 안내

- 카메라를 통한 관광지 정보 제공
- 역사적 장소 AR 복원
- 길 찾기 AR 내비게이션

12~18개월 내



음성 인식 AI 어시스턴트

- 음성 명령으로 여행 일정 관리
- 실시간 여행 정보 질의응답
- 모바일 환경 최적화

6~12개월 내



모바일 앱 & 다국어 지원

- 네이티브 iOS/Android 앱 출시
- 영어, 일본어, 중국어 지원
- 결제 및 예약 시스템 구축

18~24개월 내

결론 & Q&A



프로젝트 의의

AI 기술을 활용한 개인화된 여행 경험 제공

고객 데이터 및 선호도를 학습하여 맞춤형 여행 계획을 제안하고, 리뷰 감성 분석과 실시간 대화형 인터페이스로 사용자 만족도 극대화

학습 성과

</> LLM API 활용

OpenAI의 GPT API 통합 경험과 효율적인 컨텍스트 관리 방법론 습득

📋 프롬프트 엔지니어링

최적의 결과를 위한 시스템 역할 설계 및 구조화된 출력 생성 기법 개발

🔗 풀스택 개발 경험

프론트엔드부터 백엔드, 데이터베이스, 클라우드 배포까지 실무 개발 역량 강화