

System Test Plan(STP)



201811223 조찬형

202213524 오택호

목차

1. 서문 (Introduction)

- 1.1. 목적 (Purpose)
- 1.2. 범위 (Scope)
- 1.3. 참조 문서 (References)
- 1.4. 정의, 약어 및 용어 (Definitions, Acronyms, and Abbreviations)

2. 시험 아이템 (Test Items)

3. 시험 환경 (Test Environment)

- 3.1. 하드웨어 환경
- 3.2. 소프트웨어 환경
- 3.3. 네트워크 및 외부 연동 환경

4. 시험 설계 (Test Design)

- 4.1. 시험 접근 방법
- 4.2. 요구사항과 시험 케이스 추적성
- 4.3. 시험 유형 및 범위 (단위, 통합, 시스템, 인수 시험)

5. 시험 기준 (Pass/Fail Criteria)

6. 시험 케이스 및 절차 (Test Cases and Procedures)

7. 기록 및 보고 (Test Records and Reporting)

1. 서문 (Introduction)

1.1. 목적 (Purpose)

본 시험 계획서의 목적은 리그오브레전드 전적 분석 및 뱅픽 추천 시스템의 기능 및 성능이 요구사항대로 구현되었는지 검증하기 위한 시험 활동을 정의하는 것이다.

시험 계획서는 시험 범위, 시험 항목, 시험 환경, 시험 방법, 시험 기준 등을 명확히 하여, 개발된 시스템의 품질과 안정성을 평가하는 기준으로 활용된다.

1.2. 범위 (Scope)

시험 대상 시스템은 **Android** 기반 클라이언트와 **FastAPI** 기반 서버로 구성되며, 주요 기능은 다음과 같다.

- API 키 및 유저 idTag 입력 및 관리
- 챔피언 숙련도 데이터 수집
- 뱅픽 추천 및 UI 표시
- 플레이어 전적 수집 및 서버 전송
- 승률 예측 및 피드백 생성
- 챔피언 역할 기반 아이템 추천

1.3. 참조 문서 (References)

- SRS (Software Requirements Specification) 문서
- SDS (Software Design Specification) 문서
- Riot Games API 문서

1.4. 정의, 약어 및 용어 (Definitions, Acronyms, and Abbreviations)

를 리그 오브 레전드의 약어

DB 데이터베이스

API 라이엇 API

뱅크 챔피언 뱅과 챔피언 픽의 약어

오브젝트 리그 오브 레전드의 에픽 몬스터(바론, 드래곤,...)

챔피언 리그 오브 레전드 게임 내에서 내가 플레이 중인 캐릭터

티어 리그 오브 레전드 챔피언들의 성능을 등수로 나타낸것

속련도 얼마나	리그 오브 레전드에서 제공하는 점수 시스템으로 해당 챔피언을 플레이 했는지 알려줌
피드백	게임 라인업 분석에서 얻은 구조화 된 결과.
라인/포지션 미드	팀에서 영웅의 고정 위치 또는 역할, 예 : 탑 (상단), 정글 (정글), (중간), 원딜 (ADC), 서폿 (지원)와 같은 역할을 나타냅니다.
익명모드 가려	라이엇에서 제공하는 모드로 실시간 게임에서 자신의 닉네임을 정보를 알 수 없게함

2. 시험 아이템 (Test Items)

시험 대상 시스템은 **Android** 기반 클라이언트와 **FastAPI** 기반 서버로 구성되며, 각 모듈 및 기능별로 시험 항목을 정의한다. 각 항목은 **SRS**에서 정의된 요구사항을 기반으로 검증한다.

시험 아이템	관련 요구사항	설명
API 키 관리	Req-1.1,Req-1.2.1, Req-1.2.2	사용자가 발급받은 API 키를 입력, 저장, 유효성 검증하는 기능
유저 ID 관리	Req-1.3.1,Req-1.3.2	사용자가 입력한 게임 ID 및 태그를 저장, 유효성 검증
로그인 및 속련도 데이터 수집	Req-1.4	입력된 API 키와 유저 ID를 기반으로 챔피언 속련도 데이터를 수집하고 로컬에 저장하고 뱅크 화면으로 이동
뱅크 데이터 전송	Req-2.1	유저의 속련도, 라인, 티어 정보를 서버로 전송
뱅크 추천	Req-2.1	서버에서 전송된 정보를 바탕으로 추천 뱅크 3종류 계산
뱅크 추천	Req-2.1	유저로부터 티어, 라인을 입력받음
뱅크 UI 출력	Req-2.1	유저로부터 입력받은 티어, 라인에 맞춰 뱅크 추천

라인 입력 및 저장	Req-3.1	매치에 참여하는 10명의 플레이어 라인을 입력 및 저장
플레이어 전적 분석	Req-3.1	각 플레이어의 최근 전적 데이터를 분석하여 서버로 전송할 JSON 데이터 생성
플레이어 전적 데이터 전송	Req-3.1, Req-4.1	JSON 데이터를 서버로 전송
승률 예측 및 피드백 생성	Req-4.1, Req-4.2	서버에서 전적 분석 데이터를 기반으로 팀 승률을 계산하고 피드백 생성
피드백 UI 출력	Req-4.1, Req-4.2	서버로부터 전송받은 피드백 데이터를 클라이언트 UI로 출력
선택된 플레이어 피드백 UI	Req-4.3	선택한 다른 플레이어의 피드백 정보를 화면에 표시
아이템 추천	Req-4.4, Req-4.5	챔피언 역할을 기반으로 아이템 추천 및 예측 승률에 따라 아이템 피드백 제공

시험 아이템	관련 비기능 요구사항	설명
모델 정확도	Req-NFR-1	승률 예측 모델의 정확도 평균 80% 이상인지 확인
뱅크 추천 속도	Req-NFR-2	라인, 티어 입력시 뱅크 추천 5초 이내인지 측정
유저 분석 속도	Req-NFR-3	내 라인과 정글 라인의 분석 결과 4분 이내에 제공되는지 측정
전체 분석 시간	Req-NFR-4	전체 라인 분석 결과 10분 이내에 제공되는지 측정
픽밴 사용자 만족도	Req-NFR-5	설문 기반 픽밴 관련 만족도 조사 (CSAT) 70% 이상인지 측정
피드백 사용자 만족도	Req-NFR-6	설문 기반 피드백 관련 만족도 조사 (CSAT) 70% 이상인지 측정
아이템 추천 정확도	Req-NFR-7	모델이 추천한 상위 5개 아이템 중 실제 사용된 아이템이 포함될 확률 70% 이상
모델 일관성 검증	Req-NFR-8	시뮬레이션 테스트를 통해 측정한 모델의 일관성(정확도) 70% 이상
동시 접속 검증	Req-NFR-9, Req-NFR-11	10명이 동시접속시 모든 기능 정상 작동 및 성능 저하 없음

오류 안내	Req-NFR-10	오류 발생시 구체적인 메시지가 제공되는지 확인
UI 사용자 만족도	Req-NFR-12	설문 기반 UI 관련 만족도 조사 (CSAT) 70% 이상인지 측정
학습 용이성	Req-NFR-13	설문 기반 학습 용이성 관련 만족도 조사(CSAT) 70% 이상인지 측정
오류 허용도	Req-NFR-14	오류 후 정상 상태 복구 가능 비율 80% 이상인지 측정

3. 시험 환경 (Test Environment)

3.1. 하드웨어 환경

항목	내용
클라이언트 디바이스	Android Studio Emulator (Medium Phone (2) AVD)
CPU 코어	4
RAM	2GB
디스크	6GB
GPU	Enabled, 자동 모드
화면 해상도	1080 × 2400, LCD 밀도 420
서버	AWS EC2 t3.medium 이상
서버 CPU	2코어
서버 RAM	4GB 이상
서버 저장공간	최소 50GB

3.2. 소프트웨어 환경

항목	내용
클라이언트 OS	Android 13 (API 34, Google Play 이미지)
서버 OS	Ubuntu 22.04 LTS
서버 애플리케이션	Python 3.10, FastAPI, Uvicorn
DBMS	MySQL 8.0 (AWS RDS 사용 가능)
라이브러리/프레임워크	pandas, numpy, requests 등
클라이언트 앱	Android Studio Emulator에 APK 설치 및 초기 설정 완료
서버 실행	FastAPI 서버 배포 및 정상 구동 확인

3.3. 네트워크 및 외부 연동 환경

항목	내용
클라이언트 네트워크	PC 인터넷 연결 공유
서버 네트워크	AWS EC2 인스턴스 인터넷 연결, 포트 8000~9000 개방
외부 API 연동	Riot Games API 접근 가능
인증/보안	Riot 계정 및 발급된 API 키 사용
데이터 연동	서버-클라이언트 간 JSON 기반 데이터 통신, DB 연동

4. 시험 설계 (Test Design)

4.1. 시험 접근 방법

본 시험은 리그오브레전드 전적 분석 및 뱅픽 추천 시스템의 기능과 성능이 요구사항대로 구현되었는지 확인하기 위해 **Black-box** 기반의 기능 시험과 비기능 요구사항 검증을 병행하여 수행한다.

- 기능 시험: 사용자가 입력한 API 키, ID/태그, 라인 정보 등을 기반으로 챔피언 숙련도 수집, 뱅픽 추천, 전적 분석 및 피드백 생성 등 각 기능이

정상적으로 동작하는지 확인

- 비기능 시험: 추천 모델 정확도, 분석 속도, 전체 처리 시간, 사용자 만족도 등을 측정하여 성능 및 신뢰성 검증
- 시험 수행 방법: 클라이언트(Android 스튜디오 에뮬레이터)와 서버(AWS EC2) 환경에서 실제 데이터 및 시뮬레이션 데이터를 사용하여 테스트 수행

4.2. 요구사항과 시험 케이스 추적성

모든 시험 케이스는 SRS에 정의된 요구사항과 연계되며, 시험 아이템별로 추적성을 확보한다.

- 예시
 - TC-1 -> Req-1.1, Req-1.2.1, Req-1.2.2
 - TC-2 -> Req-2.1

이 추적성을 통해 테스트 결과가 특정 요구사항 충족 여부와 연결될 수 있도록 한다.

4.3. 시험 유형 및 범위

본 시험 계획에서는 주로 시스템 시험(System Testing)을 수행한다.

- 목적: 개발된 리그오브레전드 전적 분석 및 뱅크 추천 시스템이 SRS에 정의된 요구사항을 충족하는지 검증한다.
- 범위 :
 - Android 클라이언트와 FastAPI 서버 간 데이터 연동 및 기능 정상 동작 검증
 - API 키 입력 및 검증, 사용자 idTag 관리
 - 챔피언 속련도 데이터 수집 및 전송
 - 뱅크 추천 계산 및 UI 출력
 - 플레이어 전적 분석 및 서버 피드백 생성
 - 아이템 추천 및 방향성 제공
- 시험 제외 범위 :
 - 내부 모듈 단위 테스트(Unit Testing)는 개발 단계에서 수행된 것으로 가정
 - 외부 Riot API 자체의 정확성이나 가용성은 시험 대상이 아님
- 시험 방식 :

- 기능 요구사항과 비기능 요구사항을 중심으로 수행
- 클라이언트와 서버 간 통신, 데이터 흐름, UI 출력까지 포함한 종합적인 검증

5. 시험 기준 (Pass/Fail Criteria)

- 기능 요구사항
 - Req-1.1
 - Pass : api 발급 버튼 클릭시 Riot Developer Site로 이동
 - Fail : Riot Developer Site로 이동되지 않음
 - Req-1.2.1
 - Pass : api키가 정상적으로 저장됨
 - Fail : api키가 정상적으로 저장되지 않음
 - Req-1.2.2
 - Pass : 저장된 api키를 자동으로 입력됨
 - Fail : api키 저장을 선택했지만 자동으로 입력되지 않음
 - Req-1.3.1
 - Pass : idTag가 정상적으로 저장됨
 - Fail : idTag가 정상적으로 저장되지 않음
 - Req-1.3.2
 - Pass : 저장된 idTag가 자동으로 입력됨
 - Fail : idTag 저장을 선택했지만 자동으로 입력되지 않음
 - Req-1.4
 - Pass : idTag의 숙련도 데이터 수집하고 뱀픽 화면으로 넘어감
 - Fail : idTag의 숙련도 데이터 수집 실패
숙련도 데이터를 수집했지만 화면이 넘어가지 않음
 - Req-2.1
 - Pass : 입력한 라인, 티어, 숙련도 데이터 파일에 맞춰 뱀픽 추천
 - Fail : 숙련도 파일이 존재하지만 뱀픽 추천이 안됨
 - Req-2.2
 - Pass : 랭크 게임을 플레이중이라면 라인 선택 화면으로 이동
랭크 게임을 플레이중이 아니거나 익명모드가 활성화 된
상태라면 오류 출력
 - Fail : 랭크 게임을 플레이중이 아니어도 라인 선택 화면으로 이동
랭크 게임을 플레이중인데 오류 출력
 - Req-3.1
 - Pass : 라인 배치 후 서버로 데이터 전송 성공
 - Fail : 라인 배치 후 서버로 데이터 전송 실패
 - Req-4.1
 - Pass : 사용자 라인을 우선적으로 분석해 피드백 제공
서버로부터 피드백을 받고 즉각적으로 UI 업데이트
 - Fail : 서버로부터 피드백을 받지 못함
사용자 라인이 우선적으로 분석되지 않음
피드백을 받았는데 즉각적으로 UI가 업데이트 되지 않음
 - Req-4.2
 - Pass : 사용자 라인 분석 후 자동을 다른 라인 분석 및 피드백 제공
서버로부터 피드백을 받고 즉각적으로 UI 업데이트
 - Fail : 분석 및 피드백 생성이 중지됨
피드백을 받았는데 즉각적으로 UI가 업데이트 되지 않음

- Req-4.3
 - Pass : 선택한 라이너 피드백 출력
 - Fail : 선택한 라이너 피드백이 출력되지 않음
- Req-4.4
 - Pass : 챔피언 역할 데이터를 서버로 전송하고 아이템 추천 데이터를 받음
 - Fail : 아이템 추천 정보를 받지 못함
챔피언 역할 데이터 전송 실패
- 비기능 요구사항
 - Req-NFR-1
 - Pass : 승률 예측 모델의 정확도 80% 이상
 - Fail : 승률 예측 모델의 정확도 80% 미만
 - Req-NFR-2
 - Pass : 뱅크 화면에서 라인및 티어 입력 시 추천 결과 제공 5초 이내
 - Fail : 뱅크 화면에서 라인및 티어 입력 시 추천 결과 제공 5초 초과
 - Req-NFR-3
 - Pass : 내 라인 + 정글 분석 결과 제공 4분 이내
 - Fail : 내 라인 + 정글 분석 결과 제공 4분 초과
 - Req-NFR-4
 - Pass : 전체 라인 분석 및 피드백 제공 10분 이내
 - Fail : 전체 라인 분석 및 피드백 제공 10분 초과
 - ~~○ Req-NFR-5~~
 - ~~■ Pass : 설문 기반 픽밴 관련 만족도 조사 (CSAT) 70% 이상~~
 - ~~■ Fail : 설문 기반 픽밴 관련 만족도 조사 (CSAT) 70% 미만~~
 - ~~○ Req-NFR-6~~
 - ~~■ Pass : 설문 기반 피드백 관련 만족도 조사 (CSAT) 70% 이상~~
 - ~~■ Fail : 설문 기반 피드백 관련 만족도 조사 (CSAT) 70% 미만~~
 - Req-NFR-7
 - Pass : 테스트 데이터에서 모델이 추천한 상위 5개 아이템 중 실제 사용된 아이템이 포함될 확률 70% 이상
 - Fail : 테스트 데이터에서 모델이 추천한 상위 5개 아이템 중 실제 사용된 아이템이 포함될 확률 70% 미만
 - Req-NFR-8
 - Pass : 시뮬레이션 테스트를 통해 측정한 모델의 일관성(정확도) 70% 이상
 - Fail : 시뮬레이션 테스트를 통해 측정한 모델의 일관성(정확도) 70% 미만
 - Req-NFR-9
 - Pass : 10명이 동시접속시 모든 기능 정상 작동
 - Fail : 10명이 동시접속시 오류 발생
 - Req-NFR-10
 - Pass : 오류 발생 시 사용자에게 구체적인 메시지 제공
 - Fail : 오류 발생 시 사용자에게 구체적인 메시지 제공 실패
 - Req-NFR-11
 - Pass : 10명 동시접속시 분석기능 성능저하 없음
 - Fail : 10명 동시접속시 분석기능 성능저하 발생
 - Req-NFR-12

- Pass : 설문 기반 UI 관련 만족도 조사 (CSAT) 70% 이상
- Fail : 설문 기반 UI 관련 만족도 조사 (CSAT) 70% 미만
- Req-NFR-13
 - Pass : 설문 기반 학습 용이성 관련 만족도 조사(CSAT) 70% 이상
 - Fail : 설문 기반 학습 용이성 관련 만족도 조사(CSAT) 70% 미만
- Req-NFR-14
 - Pass : 오류 후 정상 상태 복구 가능 비율 80% 이상
 - Fail : 오류 후 정상 상태 복구 가능 비율 80% 미만

6. 시험 케이스 및 절차 (Test Cases and Procedures)

- 기능 요구사항 테스트 케이스

구분	내용
테스트 케이스ID	TC-1
관련 요구사항	Req-1.1
목적	개발자 사이트 이동이 제대로 이뤄지는지
입력물	API 키 발급받기 버튼 클릭
절차	1. 유저가 앱을 실행함 2. api키 발급받기 버튼 클릭
기대 결과	Riot Developer Site로 이동

구분	내용
테스트 케이스ID	TC-2
관련 요구사항	Req-1.2.1
목적	API 키 유효성 확인
입력물	발급받은 API키, 유저 idTag
절차	1. 유저가 앱을 실행함 2. api키 발급받기 버튼 클릭 3. 발급받은 API키 입력 후 로그인

기대 결과	API키가 유효한지 확인
-------	---------------

구분	내용
테스트 케이스ID	TC-3
관련 요구사항	Req-1.2.2
목적	API 키 불러오기 확인
입력물	로그인 상태 유지 체크박스 체크
절차	<ol style="list-style-type: none"> 1. 로그인 상태 유지 체크박스를 체크함 2. api키 입력 3. 앱을 종료 후 다시 접속해도 입력한 api키 유지
기대 결과	자동으로 입력된 API 키

테스트 케이스ID	TC-4
관련 요구사항	Req-1.2.2
목적	API 키 불러오기 확인
입력물	로그인 상태 유지 체크박스 체크 해제
절차	<ol style="list-style-type: none"> 1. 로그인 상태 유지 체크박스를 체크 해제함 2. api키 입력 3. 앱을 종료 후 다시 접속하면 api 입력란 초기화
기대 결과	비어있는 API 키

테스트 케이스ID	TC-5
관련 요구사항	Req-1.3.1
목적	유저 idTag 유효성 확인
입력물	올바른 유저 idTag, 유효한 API 키, 로그인 버튼

절차	<ol style="list-style-type: none"> 1. 사용자가 앱을 실행함 2. API키를 입력함 3. idTag를 입력 후 로그인
기대 결과	뱅크 추천 화면

테스트 케이스ID	TC-6
관련 요구사항	Req-1.3.2
목적	유저 idTag 저장
입력물	로그인 상태 유지 체크박스 체크
절차	<ol style="list-style-type: none"> 1. 사용자가 로그인 상태 유지 체크박스 체크 2. idTag와 api키 입력 3. 앱을 종료하고 다시 실행해도 입력한 idTag와 api키 유지
기대 결과	입력된 idTag와 api키

테스트 케이스ID	TC-7
관련 요구사항	Req-1.3.2
목적	idTag 불러오기 확인
입력물	로그인 상태 유지 체크박스 체크 해제
절차	<ol style="list-style-type: none"> 1. 사용자가 로그인 상태 유지 체크박스 체크 해제 2. idTag와 api키 입력 3. 앱을 종료하고 다시 실행하면 idTag 및 api키 입력칸 초기화
기대 결과	비어있는 idTag와 api 키

테스트 케이스ID	TC-8
관련 요구사항	Req-1.4

목적	로그인 확인 및 숙련도 데이터 수집
입력물	유저 idTag, 유효한 API 키, 로그인 버튼
절차	<ol style="list-style-type: none"> 1. 유저가 api키와 idTag를 입력함 2. 로그인 버튼 선택 3. idTag와 api키가 유효한것을 확인하고 숙련도 데이터 수집
기대 결과	idTag 유저의 숙련도 데이터 수집

테스트 케이스ID	TC-9
관련 요구사항	Req-2.1
목적	뱅크 추천 확인
입력물	유저의 라인, 티어
절차	<ol style="list-style-type: none"> 1. 유저가 라인과 티어를 입력함 2. 뱅크 추천 UI를 출력함
기대 결과	뱅크 추천 데이터 출력

테스트 케이스ID	TC-10
관련 요구사항	Req-2.1
목적	뒤로가기 기능 확인
입력물	뒤로가기 버튼
절차	<ol style="list-style-type: none"> 1. 유저가 잘못된 라인과 티어를 입력함 2. 뱅크 추천 확인 3. 게임 시작 버튼 클릭 4. 뒤로가기 버튼 클릭 5. 뱅크 화면에서 다시 라인과 티어를 선택
기대 결과	뱅크 화면에서 다시 라인과 티어를 선택할 수 있음

테스트 케이스ID	TC-11
관련 요구사항	Req-2.2
목적	게임중 확인
입력물	유저의 라인, 티어, 게임시작 버튼
절차	<ol style="list-style-type: none"> 1. 유저가 라인과 티어를 입력함 2. 뱅크 UI 확인 3. 게임시작 버튼 클릭 4. 해당 유저가 랭크게임을 진행중인지 확인 5. 라인배치 화면으로 이동
기대 결과	라인 배치 화면

테스트 케이스ID	TC-12
관련 요구사항	Req-2.2
목적	게임중이 아니거나 잘못된 게임 모드일때 확인
입력물	유저의 라인, 티어, 게임시작 버튼
절차	<ol style="list-style-type: none"> 1. 유저가 라인과 티어를 입력함 2. 뱅크 UI 확인 3. 게임 시작 버튼 클릭 4. 해당 유저가 랭크게임을 진행중인지 확인 5. 랭크게임이 아니거나 게임중이 아닐경우 게임중이 아니라는 알림 제공
기대 결과	뱅크 화면

테스트 케이스ID	TC-13
관련 요구사항	Req-2.2
목적	유저가 익명 모드일때 확인
입력물	유저의 라인, 티어, 게임시작 버튼

절 차	<ol style="list-style-type: none"> 1. 유저가 라인과 티어를 입력함 2. 뱅크 UI 확인 3. 게임 시작 버튼 클릭 4. 유저가 익명모드 사용중일때 게임중이 아니라는 알림 제공
기대 결과	뱅크 화면

테스트 케이스ID	TC-14
관련 요구사항	Req-3.1
목적	플레이어 라인 확인
입력물	각 플레이어의 라인
절 차	<ol style="list-style-type: none"> 1. 유저가 플레이어를 확인함 2. 각 플레이어 라인에 맞게 재배치 3. 저장 버튼 클릭
기대 결과	피드백 생성 시작, 피드백 화면

테스트 케이스ID	TC-15
관련 요구사항	Req-3.1
목적	잘못된 라인 배치 시 뒤로가기
입력물	뒤로가기 버튼
절 차	<ol style="list-style-type: none"> 1. 유저가 플레이어를 확인함 2. 잘못된 플레이어 라인 배치 3. 저장 버튼 클릭 4. 플레이어 분석중 뒤로가기 버튼 클릭 5. 분석이 중지되고 라인배치 화면으로 돌아감
기대 결과	피드백 중지, 라인 배치 화면

테스트	TC-16
-----	-------

케이스ID	
관련 요구사항	Req-4.1
목적	사용자 라인 피드백 확인
입력물	사용자 분석 데이터
절차	<ol style="list-style-type: none"> 1. 라인 배치를 완료하고 저장버튼 선택 2. 피드백 UI 출력 3. 분석 완료후 서버로 데이터 전송 후 피드백 내용을 받아 UI에 출력
기대 결과	화면에 출력된 사용자 라인 피드백

테스트 케이스ID	TC-17
관련 요구사항	Req-4.2
목적	순차적 피드백 확인
입력물	타 라인 분석 데이터
절차	<ol style="list-style-type: none"> 1. 사용자 라인 분석 완료 2. 자동으로 다음 순서 유저 분석 3. 분석 완료후 서버로 데이터 전송 후 피드백 내용을 받아 UI에 출력
기대 결과	화면에 출력된 타 라인 피드백

테스트 케이스ID	TC-18
관련 요구사항	Req-4.2
목적	피드백 로딩중 확인
입력물	분석 데이터
절차	<ol style="list-style-type: none"> 1. 피드백 화면 출력 2. 피드백 정보가 존재하면 피드백을 출력하고 존재하지 않으면 로딩 표시 출력

기대 결과	피드백 도착하지 않은 라인들 로딩 표시
-------	-----------------------

테스트 케이스ID	TC-19
관련 요구사항	Req-4.3
목적	선택한 라인 피드백 확인
입력물	라인 선택
절차	<ol style="list-style-type: none"> 1. 피드백 화면 출력 2. 사용자가 로딩중이 아닌 라인 선택 3. UI를 해당 라인 피드백 내용으로 변경
기대 결과	사용자가 선택한 라인 피드백 내용

테스트 케이스ID	TC-20
관련 요구사항	Req-4.4
목적	아이템 추천 확인
입력물	사용자 챔피언 역할 및 상대팀 5명 역할
절차	<ol style="list-style-type: none"> 1. 플레이어 라인 배치 완료 후 저장 선택 2. 사용자 챔피언 역할 및 상대팀 역할 서버로 전송 3. 서버로 부터 추천 아이템 받아서 UI에 출력
기대 결과	추천 아이템 UI에 출력

테스트 케이스ID	TC-21.1
관련 요구사항	Req-4.5
목적	아이템 피드백 확인
입력물	예측한 라인별 승률, 역할이 겹치는 불리한 라인 존재
절차	<ol style="list-style-type: none"> 1. 모든 라인 피드백이 종료됨 2. 승률 45% 이하인 라인 역할 체크

	3. 겹치는 역할이 2개 이상일경우 해당 역할에 대한 피드백 제공
기대 결과	불리한 라인에 맞춰 피드백 제공

테스트 케이스ID	TC-21.2
관련 요구사항	Req-4.5
목적	아이템 피드백 확인
입력물	예측한 라인별 승률, 역할이 겹치는 불리한 라인 없음
절차	<ol style="list-style-type: none"> 1. 모든 라인 피드백이 종료됨 2. 승률 45% 이하인 라인 역할 체크 3. 겹치는 역할이 없어 추가피드백 제공하지 않음
기대 결과	특이사항 없음 출력

● 비기능 요구사항 테스트 케이스

테스트 케이스ID	TC-22
관련 요구사항	Req-NFR-1
목적	모델 정확도 평가
입력물	전적 데이터
절차	<ol style="list-style-type: none"> 1. 모델을 위한 전적 데이터 수집 2. 학습용 데이터와 테스트 데이터 분리 3. 테스트 데이터 정확도 확인
기대 결과	테스트 정확도 평균 80% 이상

테스트 케이스ID	TC-23
--------------	-------

관련 요구사항	Req-NFR-2
목적	뱅크 추천 속도 측정
입력물	유저 라인과 티어
절차	<ol style="list-style-type: none"> 1. 유저 라인과 티어 입력 2. 라인, 티어, 숙려도 데이터로 뱅크 추천 생성 3. 뱅크 내용이 UI로 출력되는 시간 체크 4. 10회 시행후 평균시간 측정
기대 결과	라인, 티어 선택후 뱅크 내용 확인에 걸리는 평균 시간 5초 이내

테스트 케이스ID	TC-24
관련 요구사항	Req-NFR-3
목적	정글 피드백 제공 속도 측정
입력물	전적 데이터
절차	<ol style="list-style-type: none"> 1. 라인 선택화면에서 저장버튼을 누름 2. 저장 버튼을 누르고 난 후 피드백 화면에서 정글 라인 플레이어 idTag가 나올때 까지의 시간 10회 측정 3. 10회의 평균 시간을 기준으로 평가
기대 결과	정글 라인 피드백 제공시간 평균 3분 이내

테스트 케이스ID	TC-25
관련 요구사항	Req-NFR-4
목적	전체 라인 피드백 속도 측정
입력물	전적 데이터
절차	<ol style="list-style-type: none"> 1. 라인 선택화면에서 저장버튼을 누름

	<ol style="list-style-type: none"> 저장 버튼을 누르고 난 후 피드백 화면에서 모든 라인 플레이어 idTag가 나올때 까지의 시간 10회 측정 10회의 평균 시간을 기준으로 평가
기대 결과	모든 라인 피드백 제공시간 평균 10분 이내

테스트 케이스ID	TC-26
관련 요구사항	Req-NFR-5
목적	픽밴 만족도 조사
입력물	사용자 설문 결과
절차	<ol style="list-style-type: none"> 10명 이상의 유저를 대상으로 설문조사 진행 애플리케이션 사용 전 10게임 승률 기록 애플리케이션을 사용해 최소 10게임 이상 진행하고 설문 진행
기대 결과	설문 기반 픽밴 관련 만족도 조사 (CSAT) 70% 이상

테스트 케이스ID	TC-27
관련 요구사항	Req-NFR-6
목적	피드백 만족도 조사
입력물	사용자 설문 결과
절차	<ol style="list-style-type: none"> 10명 이상의 유저를 대상으로 설문조사 진행 애플리케이션 사용 전 10게임 승률 기록 애플리케이션을 사용해 최소 10게임 이상 진행하고 설문 진행
기대 결과	설문 기반 피드백 관련 만족도 조사 (CSAT) 70% 이상

테스트 케이스ID	TC-28
관련 요구사항	Req-NFR-7
목적	아이템 추천 모델 정확도 평가
입력물	랜덤한 경기의 아이템 데이터
절차	<ol style="list-style-type: none"> 1. 테스트 데이터를 입력함 2. 모델에서 추천 아이템 상위 5개 생성 3. 테스트 데이터에 해당 아이템이 있는지 확인
기대 결과	아이템이 포함되어 있는 확률 70% 이상

테스트 케이스ID	TC-29
관련 요구사항	Req-NFR-8
목적	모델 일관성 평가
입력물	티어별 랜덤 유저 20명의 최근 경기 데이터
절차	<ol style="list-style-type: none"> 1. 티어별로 랜덤 유저 20명 추출 2. 해당 유저들의 최근 경기 데이터 분석 3. 수집된 경기의 참여 유저 데이터를 기반으로 모델이 예측한 라인별 유불리(승률) 결과를 계산 4. 실제 경기 결과와, 해당 유저의 최근 10경기 평균 데이터로부터 계산된 라인별 유불리 결과를 비교 5. 일치율(정확도)을 계산하고, 전체 평균 정확도를 산출
기대 결과	모델 예측 정확도 70% 이상

테스트 케이스ID	TC-30
관련 요구사항	Req-NFR-9
목적	동시 접속 인원 테스트

입력물	10명의 클라이언트(실제 디바이스 또는 가상기기)
절차	<ol style="list-style-type: none"> 1. 안드로이드 스튜디오에서 10개의 가상 기기(또는 실제 디바이스)를 실행 2. 각 클라이언트가 동시에 서버에 로그인 및 분석 요청(API 호출)을 수행 3. 각 클라이언트에서 피드백 요청, 아이템 추천 등 주요 기능을 실행 4. 모든 요청이 정상적으로 이루어졌는지 확인
기대 결과	모든 기기에서 정상 작동 확인

테스트 케이스ID	TC-31
관련 요구사항	Req-NFR-10
목적	오류 발생 시 사용자에게 구체적이고 이해하기 쉬운 오류 메시지가 제공되는지 검증
입력물	잘못된 입력(API 키, idTag,...)
절차	<ol style="list-style-type: none"> 1. 아래의 대표 오류 상황 각각에 대해 테스트 수행 <ol style="list-style-type: none"> (A) 잘못된 API 키 입력 (B) 존재하지 않는 ID 태그 입력 (C) 게임중이지 않은 유저를 대상으로 분석 시도 (D) api 호출 제한으로 인한 호출 불가 2. 각 상황에서 시스템이 표시하는 오류 메시지를 확인 3. 메시지 내용이 사용자에게 원인을 이해시킬 수 있는 수준인지 판단
기대 결과	구체적인 오류 메시지 출력

테스트 케이스ID	TC-32
관련 요구사항	Req-NFR-11

목적	동시 접속 시 성능 저하 확인
입력물	10명의 클라이언트(실제 디바이스 또는 가상기기)
절차	<ol style="list-style-type: none"> 1. 안드로이드 스튜디오에서 10개의 가상 기기(또는 실제 디바이스)를 실행 2. 각 클라이언트가 동시에 서버에 로그인 및 분석 요청(API 호출)을 수행 3. 각 클라이언트에서 피드백 요청, 아이템 추천 등 주요 기능을 실행 4. 모든 요청이 제한시간 이내에 이루어졌는지 확인
기대 결과	Req-NFR-3, Req-NFR-4 에서 제시한 시간 이내에 모든 요청 정상 처리

테스트 케이스ID	TC-33
관련 요구사항	Req-NFR-12
목적	UI 만족도 조사
입력물	사용자 설문 결과
절차	<ol style="list-style-type: none"> 1. 10명 이상의 유저를 대상으로 설문조사 진행 2. 애플리케이션을 사용해 보고 설문 진행
기대 결과	설문 기반 UI 관련 만족도 조사 (CSAT) 70% 이상

테스트 케이스ID	TC-34
관련 요구사항	Req-NFR-13
목적	학습 용이성 조사
입력물	사용자 설문 결과
절차	<ol style="list-style-type: none"> 1. 10명 이상의 유저를 대상으로 설문조사 진행 2. 애플리케이션을 사용해 보고 설문 진행

기대 결과	설문 기반 학습 용이성 관련 만족도 조사 (CSAT) 70% 이상
-------	--------------------------------------

테스트 케이스ID	TC-35
관련 요구사항	Req-NFR-14
목적	오류 허용도 조사
입력물	오류 상황
절차	<ol style="list-style-type: none"> 1. 아래의 대표 오류 상황 각각에 대해 테스트 수행 <ol style="list-style-type: none"> (A) 잘못된 API 키 입력 (B) 존재하지 않는 ID 태그 입력 (C) 게임중이지 않은 유저를 대상으로 분석 시도 (D) api 호출 제한으로 인한 호출 불가 2. 오류 발생 후 앱이 정상적으로 복구되는지 확인 (예: 재로그인, 재시도, 이전 화면 복귀 등) 3. 각 오류 상황별 복구 성공 여부를 기록 4. 복구 성공률(복구 가능한 케이스 수 / 전체 오류 케이스 수 × 100)을 계산
기대 결과	오류 발생 후 복구 가능 비율이 80% 이상

7. 기록 및 보고 (Test Records and Reporting)

본 절에서는 시스템 테스트 수행 결과를 체계적으로 기록하고, 문제점을 분석하여 품질 개선에 반영하는 방법을 정의한다.

기록 항목

- 테스트 케이스 ID 및 이름
- 관련 요구사항

- 테스트 수행 일자
- 테스트 수행자
- 실제 수행 결과
- 판정(Pass/Fail)
- 비고(문제 발생 시 원인 및 추가 메모)

기록 방법

- 테스트 수행 후, 각 테스트 케이스별로 위 항목을 기록한다.
- 테스트 실패 시 원인을 간략히 메모하고, 필요 시 스크린샷이나 로그 첨부.