

## 요구사항 명세서 (SRS)



201811223 조찬형

202213524 오택호

# 목차

1. 소개 (Introduction)
  - 1.1. 목적 (Purpose)
  - 1.2. 범위 (Scope)
  - 1.3. 용어 정의 및 약어 (Definitions, Acronyms, and Abbreviations)
  - 1.4. 참조 문서 (References)
  - 1.5. 문서 개요 (Overview)
2. 전반적인 설명 (Overall Description)
  - 2.1. 제품 관점 (Product Perspective)
  - 2.2. 제품 기능 (Product Functions)
  - 2.3. 사용자 특성 (User Characteristics)
  - 2.4. 제약 사항 (Constraints)
  - 2.5. 가정 및 의존성 (Assumptions and Dependencies)
3. 구체적인 요구사항 (Specific Requirements)
  - 3.1. 외부 인터페이스 요구사항 (External Interfaces)
  - 3.2. 기능 요구사항 (Functional Requirements)
  - 3.3. 성능 요구사항 (Performance Requirements)
  - 3.4. 논리 데이터베이스 요구사항 (Logical Database Requirements)
  - 3.5. 설계 제약사항 (Design Constraints)

# 1. 소개 (Introduction)

## 1.1. 목적 (Purpose)

이 문서의 목적은 리그 오브 레전드 유저를 위한 AI 기반 승패 예측 및 피드백 시스템의 요구사항을 정의하는 것이다. 해당 프로그램은 라이엇 API 와 외부 게임 데이터 사이트(OP.GG 등)를 활용하여 유저의 전적을 분석하고 라인전 및 후반 성과 바탕의 승패 예측, 피드백 제공, 밴픽 및 조합에 따른 아이템 추천, 룬 추천을 제공한다. 이 시스템은 신규 및 초보 유저의 게임 이해도 증진, 중급 유저의 실력 향상을 목표로 한다.

## 1.2. 범위 (Scope)

이 소프트웨어는 전적 데이터와 현재 플레이중인 게임 참가자들 데이터를 기반으로 우리 팀과 상대팀을 분석하고 플레이 보조 프로그램이다.

우선사용자가 직접 현재 티어(Tier)와 플레이하고 싶은 라인(Lane)을 입력해야 하며, 시스템은 티어나 라인을 자동으로 인식하지 않습니다. 사용자의 숙련도(Riot API) + 앱 내부 데이터베이스(티어×라인의 승률/픽률/밴률/상대전적)를 기반으로,랭크 픽 단계에서 챔피언 추천(Pick)과 밴(Ban) 제안을 제공합니다.

게임이 시작되면 프로그램에서 API를 이용해 유저 데이터를 가져오고 승패 예측 및 분석을 시작한다. 입력된 라인에 맞춰 우선적으로 분석을 시작하고, 내 라인에 영향을 줄 수 있는 정글을, 마지막으로 탑, 미드, 바텀 순으로 분석을 해 피드백과 승패 예측 결과를 제공한다.

10명의 챔피언에 대한 정보가 들어오면 해당 챔피언들의 역할 군을 확인 후 자신의 라인과 챔피언에 맞춰 아이템 방향성을 제시한다.

해당 소프트웨어는 모바일에서 동작하며 실력 향상을 원하는 일반 유저들이 손쉽게 분석과 피드백을 받을수 있도록 설계되었다.

라이엇에서 제공하는 익명모드를 사용하는 유저의 경우 API를 통해서도 정보를 가져올 수 없기에 분석과 피드백 제공에 어려움이 있다.

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

롤	리그 오브 레전드의 약어
DB	데이터베이스
API	라이엇 API
밴픽	챔피언 밴과 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 나타낸 것
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 나타냄
익명모드 가려	라이엇에서 제공하는 모드로 실시간 게임에서 자신의 닉네임을 정보를 알 수 없게 함

### 1.4. 참조 문서 (References)

<https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>

<https://developer.riotgames.com/apis>

<https://www.op.gg>

### 1.5. 문서 개요 (Overview)

본 문서는 승률 예측 및 피드백 시스템에 대한 소프트웨어 요구사항 명세서로 프로그램이 제공해야 하는 주요 기능, 성능, 외부 시스템과의 인터페이스 등을 포함하며 기능적 요구사항과 비기능적 요구사항을 정의하고 설명한다.

## 2. 전반적인 설명 (Overall Description)

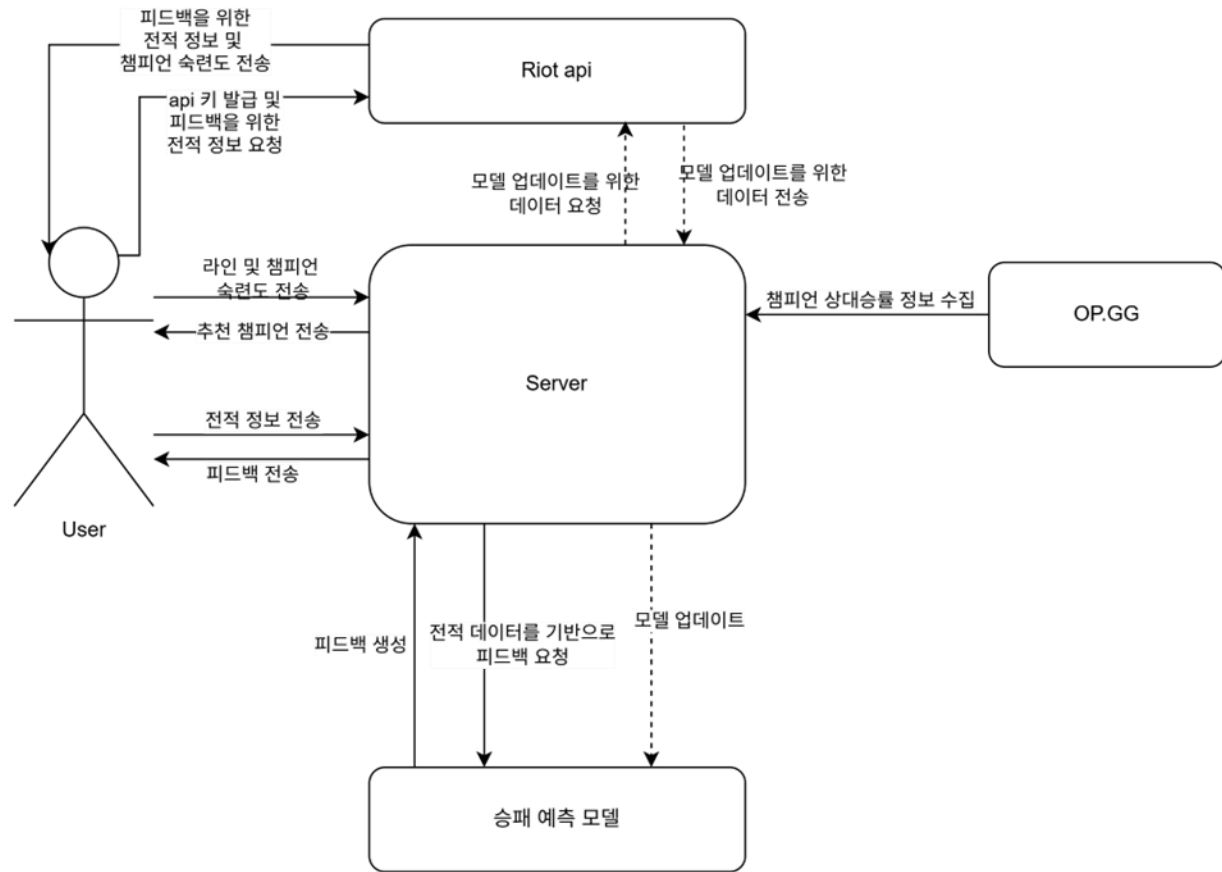
### 2.1. 제품 관점 (Product Perspective)

본 소프트웨어는 롤을 대상으로한 보조 분석 도구로, 신규 유저의 초기 적응을 돕기 위해 고안된 프로그램이다. 라이엇 공식 API 및 OP.GG같은 외부 사이트로부터 데이터를 수집하여 게임 참가자의 정보를 분석하고 유저 맞춤형 피드백을 제공한다.

자주 변경되는 패치 환경에서 신규 및 하위권 유저가 게임에 쉽게 적응할 수 있도록 돕는다.

### 2.2. 제품 기능 (Product Functions)

본 소프트웨어는 아래와 같은 구조를 갖는다.



본 소프트웨어는 아래와 같은 기능을 갖는다.

- API 설정
  - Riot API 발급 사이트로 이동하여 키 발급
  - 발급받은 API 키 입력
- 롤 아이디 태그 입력하기
  - 현재 플레이 중인 롤 아이디 입력(예:user#123)
- 챔피언 추천 시스템
  - 숙련도와 챔피언 티어,라인 데이터를 이용하여 적절한 챔피언 추천
- 아이템 추천
  - 내 챔피언과 상대 챔피언들을 고려하여 아이템 추천

- 게임 피드백 받기
  - 현재 경기에서 팀원과 상대방의 상태를 분석하여 피드백 제공
  - 익명모드 유저의 데이터는 제공하지 않음

### 2.3. 사용자 특성 (User Characteristics)

본 소프트웨어의 주요 사용자는 리그 오브 레전드를 자주 플레이하며 자신의 실력 향상을 원하는 유저입니다. 사용자는 전략적 밴픽 및 실력 향상을 위한 피드백을 목적으로 서비스를 이용하며 간단한 모바일 또는 ~~컴퓨터~~ 조작이 가능한 수준의 지식을 필요로 합니다.

사용자는 자신의 챔피언 숙련도와 전적을 기반으로 챔피언 추천 및 밴 추천, 그리고 피드백을 제공받을 수 있고, 현재 게임 상황에 맞는 아이템을 추천받을 수 있습니다.

### 2.4. 제약 사항 (Constraints)

본 소프트웨어는 라이엇에서 제공하는 API를 사용하기 때문에 API 호출 횟수 제한이나 응답 지연이 발생할 수 있습니다. 또한 라이엇의 데이터 사용 정책을 준수해야 합니다.

### 2.5. 가정 및 의존성 (Assumptions and Dependencies)

본 소프트웨어는 사용자가 라이엇 계정을 가지고 있다는 것을 가정하고 설계되었습니다. 시스템은 라이엇 API에 의존하며 응답 속도는 상황에 따라 달라질 수 있습니다. 본 소프트웨어는 지속적인 인터넷 연결이 가능한 상태에서의 실행을 전제로 합니다.

## 3. 구체적인 요구사항 (Specific Requirements)

### 3.1. 외부 인터페이스 요구사항 (External Interfaces)

#### 3.1.1. 사용자 인터페이스

본 시스템의 기본 화면 구성은 다음과 같다

- API 키 입력 및 아이디, 태그 입력 화면
- 챔피언 픽(Pick)과 밴(Ban) 추천 화면
- 모든 플레이어 ID 태그 표시 및 라인 확인 화면
- 피드백 및 아이템 방향 제시 화면

### 3.1.2. 소프트웨어 인터페이스

본 시스템은 클라이언트-서버 구조로 동작한다.

- 서버 : Python 기반 **FastAPI** 프레임 워크를 사용하여 **API** 요청을 처리한다. 내부적으로 사용자 게임 데이터를 가공한다.

- 클라이언트 : **Kotlin(Android)** 으로 개발된 모바일 애플리케이션으로 **REST API**를 통해 서버와 통신한다.

- 주요 연동 API :

- **Summoner-V4**: 닉네임으로 사용자 정보 조회

- **Champion-Mastery-V4**: 챔피언 숙련도 조회

- **Match-V5**: 최근 전적 분석

- **Spectator-V5**: 진행중인 게임 관전

### 3.1.3. 통신 인터페이스

본 시스템은 클라이언트(**Android** 앱) 와 서버(**FastAPI**) 간에 통신을 위해 **HTTP** 기반 **RESTful API**를 사용하며, **json** 포맷으로 응답을 반환한다.

## 3.2. 기능 요구사항 (Functional Requirements)

### 1. API 및 아이디 입력

구분	내용
요구사항ID	Req-1.1
요구사항명	API 키 발급
개요	API 키 발급받기 버튼을 클릭해 라이엇 개발자 사이트로 이동
입력물	API 키 발급받기 버튼 클릭
사전조건	최근 <b>24시간</b> 내에 <b>API</b> 를 입력한 기록이 없음
시나리오	1. 유저가 앱, 또는 웹을 실행함 2. <b>API</b> 발급받기 버튼 클릭



	3. 라이엇 개발자 사이트로 이동
산출물	API 키
기타 요구사항	사용자가 라이엇 계정을 보유하고 있어야함
예외 시나리오	

구분	내용
요구사항ID	Req-1.2.1
요구사항명	API 입력
개요	발급받은 API 키 입력
입력물	발급받은 API 키
사전조건	Req-1.1 에서 API 키를 발급받음
시나리오	<ol style="list-style-type: none"> <li>1. 라이엇 개발자 사이트에서 발급받은 API 키 복사</li> <li>2. API 키 입력란에 붙여넣기</li> <li>3. 저장 버튼 클릭</li> </ol>
산출물	입력된 API 키
기타 요구사항	사용자가 라이엇 계정을 보유하고 있어야함
예외 시나리오	

구분	내용
요구사항ID	Req-1.2.2
요구사항명	API 입력 유지
개요	로그인 상태 유지 체크박스를 체크함
입력물	로그인 상태 유지 체크박스
사전조건	API 키를 발급받고 입력한 기록
시나리오	<ol style="list-style-type: none"> <li>1. 사용자가 앱을 실행함</li> <li>2. 로그인 상태 유지 체크박스가 체크되어있음</li> <li>3. 기존 API 키 값 자동 입력</li> </ol>

산출물	자동으로 입력된 API 키
기타 요구사항	사용자가 라이엇 계정을 보유하고 있어야함
예외 시나리오	

구분	내용
요구사항ID	Req-1.3.1
요구사항명	사용자 ID 입력
개요	플레이 할 자신의 게임 ID 입력
입력물	게임 ID와 태그
사전조건	사용자가 자신의 게임 ID와 태그를 알고 있음 API 키가 입력 되었음
시나리오	<ol style="list-style-type: none"> <li>1. 사용자가 게임 ID와 태그를 입력</li> <li>2. 확인 버튼을 클릭</li> <li>3. 통신을 통해 API 키와 ID 및 태그가 유효한지 확인</li> <li>4. 유효한 경우 뱅크 화면으로 이동</li> </ol>
산출물	뱅크 화면
기타 요구사항	사용자가 라이엇 계정을 보유하고 있어야함
예외 시나리오	

구분	내용
요구사항ID	Req-1.3.2
요구사항명	사용자 ID 입력 유지
개요	로그인 상태 유지 체크박스를 체크함
입력물	로그인 상태 유지 체크박스
사전조건	사용자가 ID를 입력함
시나리오	<ol style="list-style-type: none"> <li>1. 사용자가 앱을 실행함</li> <li>2. 로그인 상태 유지 체크박스가 체크되어있음</li> </ol>

	<b>3. 기존 ID 자동 입력</b>
산출물	뱅크 화면
기타 요구사항	사용자가 라이엇 계정을 보유하고 있어야함
예외 시나리오	

구분	내용
요구사항ID	<b>Req-1.4</b>
요구사항명	로그인 확인
개요	사용자가 로그인을 시도함
입력물	게임 ID와 태그, API 키, 로그인 확인 버튼
사전조건	ID 와 태그, API 키가 유효함
시나리오	<ol style="list-style-type: none"> <li>1. API를 통해 사용자의 챔피언 속련도 데이터 수집</li> <li>2. 로컬 데이터 파일에 저장</li> </ol>
산출물	사용자의 챔피언 속련도 데이터 파일
기타 요구사항	사용자가 라이엇 계정을 보유하고 있어야함
예외 시나리오	사용자 ID가 유효하지 않음->아이디 존재하지 않는다고 알림 API가 만료됨->만료된 API라고 알림

## 2. 뱅크 추천

구분	내용
요구사항ID	<b>Req-2.1</b>
요구사항명	라인 티어 입력
개요	자신의 티어와 라인을 입력함
입력물	자신의 라인 티어
사전조건	올바른 api와 idTag를 입력
시나리오	<ol style="list-style-type: none"> <li>1. 자신의 라인과 티어를 입력</li> </ol>

	2. 라인, 티어, 숙련도 데이터에 맞는 밴픽 추천
산출물	추천 챔피언 및 해당 챔피언과 관련된 밴(Ban) 추천.
기타 요구사항	사용자가 자신의 티어와 라인을 정확하게 입력해야 함
예외 시나리오	

구분	내용
요구사항ID	Req-2.2
요구사항명	게임중 확인
개요	티어와 라인 입력 후 게임 시작 버튼 클릭
입력물	게임 시작 버튼
사전조건	라인과 티어를 입력함
시나리오	<ol style="list-style-type: none"> <li>1. 게임 매칭이 잡힘</li> <li>2. 픽밴창에서 자신의 티어와 라인을 확인함</li> <li>3. 해당 티어와 라인을 입력</li> <li>4. 게임 시작 클릭</li> </ol>
산출물	플레이어 라인 선택 화면
기타 요구사항	매칭이 랭크게임이어야 함 사용자가 자신의 티어와 라인을 정확하게 입력해야 함
예외 시나리오	

### 3. 라인 확인 저장

구분	내용
요구사항ID	Req-3.1
요구사항명	플레이어 라인 선택
개요	플레이어 10명의 라인 데이터를 확인 및 저장
입력물	플레이어 라인 정보
사전조건	밴픽 화면에서 게임시작 버튼을 누른 상태

시나리오	<ol style="list-style-type: none"> <li>1. 사용자가 '게임 시작' 버튼을 탭한다.</li> <li>2. 플레이어 정보 표시 화면에 두 팀(Blue/Red)의 10명 플레이어 idTag를 표시합니다.</li> <li>3. 라인 교환 플레이어가 라인을 바꿀 경우, 해당 플레이어가 위치한 칸을 클릭하여 서로 위치를 교환할 수 있습니다.</li> <li>4. 라인 저장 저장 버튼 클릭 플레이어 idTag 최종 라인 배치를 Top → Jungle → Mid → Bottom → Support 순서로 정렬하여 저장합니다.</li> <li>5. 초기화 버튼을 클릭하면, 플레이어 라인이 교환되기 전의 초기 상태로 복원됩니다.</li> <li>6. 서버 전송 라인업 및 분석 요청을 서버로 전송합니다.</li> </ol>
산출물	<ol style="list-style-type: none"> <li>1. 10명의 플레이어가 속한 팀과 idTag</li> <li>2. 10명의 플레이어 라인</li> </ol>
기타 요구사항	입력한 플레이어 idTag는 반드시 게임 내에 존재해야 함
예외 시나리오	플레이어에 익명 모드를 사용하는 유저 존재->유저 id 대신 챔피언 이름으로 표시

#### 4. 피드백 제공 및 아이템 제시

구분	내용
요구사항ID	Req-4.1
요구사항명	내 라인 피드백 제공
개요	플레이어 10명의 데이터를 수집 후 내 라인 피드백 제공
입력물	플레이어 분석 데이터

사전조건	플레이어가 라인 정보를 입력하고 저장버튼 누른 상태
시나리오	<ol style="list-style-type: none"> <li>1. 사용자가 저장 버튼을 누름</li> <li>2. api를 통해 사용자의 라인 우선 분석</li> <li>3. 분석 후 데이터 서버로 전송</li> <li>4. 서버로부터 피드백 결과 수신</li> <li>5. UI에 피드백 내용 출력</li> </ol>
산출물	내 라인 피드백 데이터
기타 요구사항	저장 버튼 버튼을 선택해야 함 해당 라인 피드백을 서버로부터 받아야 함
예외 시나리오	데이터가 부족하거나 존재하지 않음->id 옆에 정보부족 출력

구분	내용
요구사항ID	Req-4.2
요구사항명	순차적 피드백 제공
개요	다른 라인 피드백 제공
입력물	플레이어 분석 데이터
사전조건	사용자의 라인 분석이 완료됨
시나리오	<ol style="list-style-type: none"> <li>1. 사용자 라인 분석 완료됨</li> <li>2. 정해진 순서에 따라 다음 라인 분석 시작</li> <li>3. 분석 후 데이터 서버로 전송</li> <li>4. 서버로부터 피드백 결과 수신</li> <li>5. UI에 피드백 내용 출력</li> </ol>
산출물	다른 라인 피드백 데이터
기타 요구사항	실시간으로 UI가 업데이트 되어야함
예외 시나리오	사용자 라인이 분석중->다른 라인 로딩 표시 데이터가 부족하거나 존재하지 않음->id 옆에 정보부족 출력

구분	내용
요구사항ID	Req-4.3

요구사항명	타 라인 피드백 확인
개요	다른 라인을 클릭해 타 라인 피드백 확인
입력물	타 라인 선택
사전조건	선택한 라인이 피드백이 완료된 상태
시나리오	<ol style="list-style-type: none"> <li>1. 사용자가 타 라인을 선택</li> <li>2. UI에 보여주는 피드백 정보 변경</li> </ol>
산출물	타 라인들의 피드백 데이터
기타 요구사항	해당 라인 피드백을 서버로부터 받아야 함
예외 시나리오	데이터가 부족하거나 존재하지 않음->id 옆에 정보부족 출력

구분	내용
요구사항ID	Req-4.4
요구사항명	아이템 추천
개요	챔피언 역할에 따라 아이템 추천
입력물	내 챔피언과 상대팀 챔피언 5명의 역할
사전조건	10명의 챔피언 정보가 들어온 상태
시나리오	<ol style="list-style-type: none"> <li>1. 10명의 챔피언 정보 수집</li> <li>2. 그중 내 챔피언과 상대 챔피언 5명의 역할정보를 서버로 전송</li> <li>3. 서버로부터 추천 아이템 수신</li> <li>4. UI에 추천 아이템 업데이트</li> </ol>
산출물	추천 아이템
기타 요구사항	
예외 시나리오	

구분	내용
요구사항ID	Req-4.5

요구사항명	아이템 피드백
개요	라인별 예측 승률에 따라 피드백 제공
입력물	라인별 예측 승률
사전조건	모든 라인 피드백이 완료된 상태
시나리오	1. 라인별 예측 승률 확인 2. 불리한 라인 역할 체크 3. 해당 역할에 맞는 피드백 제공
산출물	아이템 피드백
기타 요구사항	
예외 시나리오	

### 3.3. 비기능 요구사항 (Non-functional Requirements)

#### 1. 성능

항목	요구사항	요구사항ID
모델 정확도	승률 예측 모델의 정확도 80% 이상	Req-NFR-1
뱅크 추천 속도	라인 입력 시 추천 결과 제공 5초 이내	Req-NFR-2
정글 포함 분석	내 라인 + 정글 분석 결과 제공 3분 이내	Req-NFR-3
전체 분석 시간	전체 라인 분석 및 피드백 제공 10분 이내	Req-NFR-4
<del>픽밴 사용자 만족도</del>	<del>설문 기반 픽밴 관련 만족도 조사 (CSAT) 70% 이상</del>	<del>Req-NFR-5</del>
<del>피드백 사용자 만족도</del>	<del>설문 기반 피드백 관련 만족도 조사 (CSAT) 70% 이상</del>	<del>Req-NFR-6</del>
아이템 예측 모델 정확도	모델이 추천한 상위 5개 아이템 중 실제 사용된 아이템이 포함될 확률 70% 이상	Req-NFR-7
모델 일관성	시뮬레이션 테스트를 통해 측정한 모델의 일관성(정확도) 70% 이상	Req-NFR-8
동시 접속 인원	10명이 동시접속시 모든 기능 정상 작동	Req-NFR-9



2. 신뢰성

항목	요구사항	요구사항ID
오류 안내	오류 발생 시 사용자에게 구체적인 메시지 제공	Req-NFR-10
동시 접속 성능	10명 동시접속시 분석기능 성능저하 없음	Req-NFR-11

3. 사용성

항목	요구사항	요구사항ID
UI 사용자 만족도	설문 기반 UI 관련 만족도 조사 (CSAT) 70% 이상	Req-NFR-12
학습 용이성	설문 기반 학습 용이성 관련 만족도 조사(CSAT) 70% 이상	Req-NFR-13
오류 허용도	오류 후 정상 상태 복구 가능 비율 80% 이상	Req-NFR-14

3.4. 논리 데이터베이스 요구사항 (Logical Database Requirements)

항목	내용
데이터 항목	챔피언 정보(이름, 역할) 경기 전적(게임 ID, 킬, 데스, 어시스트 등) 챔피언 티어 정보(이름, 라인, 승률, 티어 등) 챔피언 상대 승률 정보(챔피언 정보, 라인, 승률 등)
주요 엔티티	챔피언, 게임 ID, 승률, 라인, 티어 등

3.5. 설계 제약사항 (Design Constraints)

서버

항목	내용
모델 의존성	리그래션 모델 사용, TensorFlow 프레임워크에 의존
개발 언어	Python 3.10 이상

주요 라이브러리	pandas, numpy, scikit-learn, joblib, tensorflow 등
프레임워크	FastAPI (RESTful API 제공)
데이터 포맷	입력 및 출력은 JSON 형식
네트워크 프로토콜	HTTP 기반 통신
리소스 환경	모델 실행을 위한 CPU 또는 GPU 환경 필요

#### 클라이언트

항목	내용
모델 의존성	단일 액티비티 + Jetpack Navigation, MVVM + Repository 구조
개발 언어	Kotlin 2.0 이상
지원 플랫폼	Android 14(API 34+)
주요 라이브러리	Coroutines/Flow, Lifecycle/ViewModel KTX, Navigation-Compose, Retrofit2+OkHttp3(또는 Ktor Client), Moshi/Gson(JSON), Coil 3(이미지), DataStore(설정), File I/O(JSON 저장), Room(선택)
프레임워크	Jetpack Compose + MVVM + Repository 패턴
외부 API 연동	서버의 FastAPI REST 엔드포인트 호출(분석/추천 수신). 클라이언트에서 Riot Games API 직접 호출
데이터 포맷	요청/응답: JSON. 로컬 저장:
네트워크 프로토콜	HTTP 기반 통신
리소스 환경	
호출 제한	라이엇 API 요청 제한 (Rate Limit) 고려 필요