

System Test Plan(STP)

KU.GG

LoL 웹기반 전적검색 & 분석 시스템

Team 브론즈에게 희망을
(LoL api를 이용한 유저 데이터 분석)

컴퓨터공학과 201714170 조해성
컴퓨터공학과 201714167 양현영
컴퓨터공학과 201714168 유호원
컴퓨터공학과 201714169 조영래

1. Functional Requirements Test Cases

test number	test 항목	description	expected result	system function
1	페이지 이동	<ol style="list-style-type: none"> 메뉴에서 페이지 탭 중 하나를 클릭한다. 메인 배너를 클릭한다. 	페이지 탭을 누르면 선택한 페이지로 이동하고 메인 배너를 누르면 홈페이지로 이동한다.	3.2.1.1
2	챔피언명 or 소환사명 검색	<ol style="list-style-type: none"> 메뉴에서 검색창에 챔피언명 또는 소환사명을 검색한다. 입력한 챔피언 정보 또는 소환사 정보 페이지로 이동한다. 	DB에 해당 챔피언 또는 소환사 정보가 있다면 해당 챔피언 또는 소환사 정보가 보인다.	3.2.1.2
3	존재하지 않는 챔피언명 or 소환사명 검색	<ol style="list-style-type: none"> 메뉴에서 검색창에 챔피언명 또는 소환사명을 검색한다. 챔피언 정보 또는 소환사 정보 페이지로 이동한다. 	DB에 해당 챔피언 또는 소환사 정보가 없으면 '존재하지 않는 챔피언 or 소환사입니다.'가 출력된다.	3.2.1.2
4	유저 챔피언 정보	<ol style="list-style-type: none"> 챔피언을 하나 선택한다. 	유저가 선택한 챔피언과 일치하는 라인과 아이템 트리를 보여준다	3.2.2.1
5	유저 챔피언 정보 취소 후 재선택	<ol style="list-style-type: none"> 챔피언을 하나 선택한다. 선택을 취소하고 다른 챔피언을 선택한다 	선택한 챔피언과 일치하는 라인과 아이템 트리를 보여주고 유저가 다른 챔피언을 선택하면 그에 맞추어 라인과 아이템 트리도 바뀐다.	3.2.2.1
6	상대 챔피언 정보	<ol style="list-style-type: none"> 챔피언과 라인을 선택한다. 	선택한 챔피언과 라인이 표시되고 상대승률과 승리전략을 보여준다.	3.2.2.2
7	상대 챔피언 정보 취소 후 재선택	<ol style="list-style-type: none"> 챔피언과 라인을 선택한다. 선택을 취소하고 다른 챔피언과 라인을 선택한다. 	선택한 챔피언과 라인이 표시되고 상대승률과 승리전략을 보여주고 유저가 다른 챔피언과 라인을 선택하면 그에 맞추어 상대승률과 승리전략이 바뀐다.	3.2.2.2
8	아이템 트리 시뮬레이션	<ol style="list-style-type: none"> 챔피언을 선택한다. 상대 챔피언을 선택한다. 새로운 테스트 데이터를 대입하여 승패여부를 예측한다. 	선택한 챔피언과 상대 챔피언이 표시되고 아이템마다 승리할 확률이 나타난다.	3.2.2.3

9	소환사 검색	검색 창에 소환사명을 입력하고 검색 버튼을 누른다.	입력한 소환사명이 DB에 있다면 해당 소환사의 전적 정보 페이지로 이동한다.	3.2.3.1
10	존재하지 않는 소환사 검색	검색 창에 소환사명을 입력하고 검색 버튼을 누른다.	입력한 소환사명이 DB에 없을 경우 '존재하지 않는 소환사명입니다.'가 출력된다.	3.2.3.1
11	소환사 분석	소환사 분석 페이지에서 소환사명을 검색한다.	입력한 소환사명의 정보를 다이어그램으로 나타내고 다른 챔피언을 추천한다. 입력한 소환사명의 자주 선택하는 챔피언, 아이템트리, 특성, 자주 가는 라인을 출력한다.	3.2.4.1, 3.2.4.2
12	존재하지 않는 소환사 분석	소환사 분석 페이지에서 소환사명을 검색한다.	입력한 소환사명이 DB에 없을 경우 '존재하지 않는 소환사명입니다.'가 출력된다.	3.2.4.1, 3.2.4.2
13	첫번째 오브젝트 승률 예측	<ol style="list-style-type: none"> 1. 드롭다운에서 티어 하나를 선택한다. 2. 첫번째로 획득한 오브젝트의 체크박스를 클릭한다. 3. 승률 예측 버튼을 클릭한다. 	선택된 티어와 첫번째 오브젝트에 대해 예측된 승률 정보와, 이번 시즌 가장 승률이 높은 첫번째 오브젝트를 출력된다.	3.2.5.1 3.2.5.2
14	오브젝트 킬 당 승률 예측	<ol style="list-style-type: none"> 1. 드롭다운에서 티어 하나를 선택한다. 2. 각 오브젝트 드롭다운에서 킬 수를 선택한다. 3. 승률 예측 버튼을 클릭한다. 4. 승률과 추천 오브젝트 출력을 확인한다. 	선택된 티어와 오브젝트 킬 수에 대해 예측된 승률과 추천 오브젝트가 화면에 출력된다.	3.2.5.3 3.2.5.4
15	첫번째 오브젝트 & 오브젝트 킬 당 승률 예측	<ol style="list-style-type: none"> 1. 드롭다운에서 티어 하나를 선택한다. 2. 첫번째로 획득한 오브젝트 체크박스를 선택한다. 3. 각 오브젝트 드롭다운에서 킬 수를 선택한다. 4. 승률 예측 버튼을 클릭한다. 5. 승률 출력을 확인한다. 	선택된 티어와 첫번째 오브젝트, 그리고 오브젝트 킬 수에 대해 예측된 승률이 화면에 출력된다.	3.2.5.5 3.2.5.6

16	유저 티어 정보 요청	<ol style="list-style-type: none"> 1. 시스템이 티어정보를 불러오기 위해 api에 request를 보낸다. 2. request의 status 코드가 정상적인지 확인한다. 3. 정상적인 코드가 아닐 경우 다시 request를 보낸다. 4. request를 통해 얻은 데이터를 메모리에 저장한다. 	<p>시스템이 DB의 유저 티어 정보 데이터 중 아직 request를 보내지 않는 유저 티어 정보 데이터에 대해 request를 보낸다.</p> <p>request에 대한 status 코드가 200이 나온다.</p> <p>request에 대한 status 코드가 200이 아닐경우 다시 request를 요청한다.</p> <p>request를 통해 얻은 데이터를 정상적으로 메모리에 올린다.</p> <p>데이터를 DB에 저장할 때 중복되는 것은 제외하고 저장한다.</p>	3.2.6.1
17	유저 티어 정보 DB 저장	<ol style="list-style-type: none"> 1. 시스템은 DB에 연결을 요청한다. 2. 시스템은 메모리에 있는 유저 티어 데이터를 DB에 저장한다. 	<p>DB에 정상적으로 연결된다.</p> <p>메모리에 있는 유저 티어 정보 데이터가 정상적으로 DB에 저장된다.</p>	3.2.6.1
18	유저 상세정보 요청	<ol style="list-style-type: none"> 1. 시스템이 유저 티어 정보 DB에 새롭게 추가된 유저들을 뽑아낸다. 2. 시스템이 새롭게 추가된 유저들의 상세정보를 얻기 위해 api에 request를 보낸다. 3. request의 status 코드가 정상적인지 확인한다. 4. 정상적인 코드가 아닐경우 다시 request를 보낸다. 5. request를 통해 얻은 데이터를 메모리에 저장한다. 	<p>시스템이 DB의 유저 상세정보 데이터 중 아직 request를 보내지 않는 유저 상세정보 데이터에 대해 request를 보낸다.</p> <p>request에 대한 status code가 200이 나온다.</p> <p>request를 통해 얻은 데이터를 정상적으로 메모리에 올린다.</p> <p>데이터를 DB에 저장할 때 중복되는 것은 제외하고 저장한다.</p>	3.2.6.2
19	유저 상세정보 DB 저장	<ol style="list-style-type: none"> 1. 시스템은 DB에 연결을 요청한다. 2. 시스템은 메모리에 있는 유저 상세정보 데이터를 DB에 저장한다. 	<p>DB에 정상적으로 연결된다.</p> <p>메모리에 있는 유저 상세 정보 데이터가 정상적으로 DB에 저장된다.</p>	3.2.6.2
20	유저 matchlist	<ol style="list-style-type: none"> 1. 시스템은 유저들이 최근에 플레이한 게임을 	<p>시스템이 DB의 matchlist데이터 중 아직 request를 보내지 않는</p>	3.2.6.3

	데이터 요청	<p>DB에서 가져온다.</p> <ol style="list-style-type: none"> 2. 시스템은 가장 최근에 플레이한 데이터 이후의 게임들에 대한 리스트를 받기 위해 api에 request를 보낸다. 3. request의 status 코드가 정상적인지 확인한다. 4. 정상적인 코드가 아닐경우 다시 request를 보낸다. 5. request를 통해 얻은 유저의 matchlist 데이터를 메모리에 저장한다. 	<p>matchlist에 대해 request를 보낸다.</p> <p>request에 대한 status code가 200이 나온다.</p> <p>request를 통해 얻은 데이터를 정상적으로 메모리에 올린다.</p> <p>데이터를 DB에 저장할 때 중복되는 것은 제외하고 저장한다.</p>	
21	유저 matchlist 데이터 DB 저장	<ol style="list-style-type: none"> 1. 시스템은 DB에 연결을 요청한다. 2. 시스템은 메모리에 있는 matchlist 데이터들을 DB에 저장한다. 	<p>DB에 정상적으로 연결된다.</p> <p>메모리에 있는 유저 matchlist 정보 데이터가 정상적으로 DB에 저장된다.</p>	3.2.6.3
22	match 상세정보 요청	<ol style="list-style-type: none"> 1. 시스템은 메모리에 올라가있는 matchlist들에서 최근 match들만을 뽑아낸다. 2. 시스템은 최근 match에 대해서 해당 match들의 상세정보에 대해 api에 request를 보낸다. 3. request의 status 코드가 정상적인지 아닌지 확인한다. 4. 정상적인 코드가 아닐경우 다시 request를 보낸다. 5. request를 통해 얻은 match 상세정보 데이터를 메모리에 저장한다. 	<p>시스템이 DB의 match 상세 정보 데이터 중 아직 request를 보내지 않는 match에 대해 request를 보낸다.</p> <p>request에 대한 status code가 200이 나온다.</p> <p>request를 통해 얻은 데이터를 정상적으로 메모리에 올린다.</p> <p>데이터를 DB에 저장할 때 중복되는 것은 제외하고 저장한다.</p>	3.2.6.4
23	match 상세정보 DB 저장	<ol style="list-style-type: none"> 1. 시스템은 DB에 연결을 요청한다. 2. 시스템은 메모리에 있는 match 상세정보 데이터들을 DB에 저장한다. 	<p>DB에 정상적으로 연결된다.</p> <p>메모리에 있는 match 상세 정보 데이터가 정상적으로 DB에 저장된다.</p>	3.2.6.4
24	match 타임라인 데이터 요청	<ol style="list-style-type: none"> 1. 시스템은 메모리에 올라가있는 matchlist들에서 최근 	<p>시스템이 DB의 match 타임라인 데이터 중 아직 request를 보내지 않는 match 타임라인에</p>	3.2.6.5

		<p>match들만 뽑아낸다.</p> <ol style="list-style-type: none"> 1. 시스템은 최근 match에 대해서 해당 match들의 타임라인 정보에 대해 api에 request를 보낸다. 2. request의 status 코드가 정상적인지 아닌지 확인한다. 3. 정상적인 코드가 아닐경우 다시 request를 보낸다. 4. request를 통해 얻은 match 타임라인 데이터를 메모리에 저장한다. 	<p>대해 request를 보낸다.</p> <p>request에 대한 status code가 200이 나온다.</p> <p>request를 통해 얻은 데이터를 정상적으로 메모리에 올린다.</p> <p>데이터를 DB에 저장할 때 중복되는 것은 제외하고 저장한다.</p>	
25	match 타임라인 데이터 DB 저장	<ol style="list-style-type: none"> 1. 시스템은 DB에 연결을 요청한다. 2. 시스템은 메모리에 있는 match 타임라인 데이터들을 DB에 저장한다. 	<p>DB에 정상적으로 연결된다.</p> <p>메모리에 있는 match 타임라인 정보 데이터가 정상적으로 DB에 저장된다.</p>	3.2.6.5
26	데이터 전처리	<ol style="list-style-type: none"> 1. 시스템이 LoL api에 정상적으로 request를 보냈는지 확인한다. 2. 시스템이 request를 통해 얻은 데이터에 전처리 작업을 정상적으로 수행했는지 확인한다. 3. 시스템이 전처리 작업을 거친 데이터를 데이터베이스에 저장했는지 확인한다. 	<p>결측치나 이상치들을 제외한 유용한 데이터들만 선별하고 필요하면 왜곡된 데이터 수정도 마친 데이터들이 데이터 베이스에 저장이 된다.</p>	3.2.6.6
27	모델 입출력	<ol style="list-style-type: none"> 1. 데이터 모델이 오버피팅이나 언더피팅이 되진 않았는지 확인하기 위해 여러가지 데이터 input을 준비한다. 2. 데이터 모델에 input을 입력한다. 	<p>모델별로 input에 맞는 output 데이터를 오버피팅이나 언더피팅없이 높은 정확도로 결과를 예측한다.</p>	3.2.6.7

2. Non-functional requirements

1	정적수치 요구사항 test	시스템은 웹브라우저가 실행 가능한 메모리를 가진 Mac,	각 OS에서 웹브라우저를 통해 KUGG가 정상적으로	3.3.1
---	----------------	---------------------------------	------------------------------	-------

		Windows, Linux 기기에서 사용 가능하며 모바일 기기도 사용 가능하다.	동작한다.	
2	동적수치 요구사항 test	시스템은 검색 및 분석 결과의 98%를 5초 안에 처리한다.	KUGG웹에서 검색과 분석 기능을 실행하고 5초 이내에 실행된다.	3.3.2
3	Reliability test	시스템은 사용자가 KUGG를 사용 중 crash가 발생하면 홈 화면 부터 재시작시킨다.	유저가 KUGG를 사용하다 오류가 발생하면 홈 화면에서 재시작한다.	3.4.1
4	Availability test	시스템은 사용자가 요청한 전적 정보 또는 데이터 분석 정보를 가능한 빠르게 전달해준다.	유저가 데이터를 요청 시 가능한 빠르게 전달 받는다.	3.4.2
5	Security test	DB 서버에 접속하려면 정해진 user와 password를 입력해야 한다.	DB 서버에 접속 요청시 user와 password가 일치하면 정상적으로 DB 서버에 접속할 수 있다.	3.4.3
6	Maintainability test	KUGG는 유지보수에 용이하게 모듈식으로 구성되고, 각 모듈은 독립적이어야 한다.	LOL api에서 실시간으로 데이터를 받아와서 DB 서버에 저장할 때 웹 서버는 영향을 받지 않는다.	3.4.4
7	Portability test	개발은 python 3.6버전으로 하고 특정 DB 및 라이브러리들을 사용한다.	다른 pc에서 KUGG 웹 서버를 실행시켜도 정상적으로 동작한다.	3.4.5

3. Success criteria

1	게임 실력 향상 여부	<ol style="list-style-type: none"> 1. LOL을 플레이하는 유저들에게 KUGG를 사용하게 한다. 2. 일정 기간 이후 그들의 게임 실력 향상 여부를 확인한다. 	실제 유저 대상으로 실력 향상에 도움이 되었다고 증언받는다.	Success criteria 1
2	최신 정보 출력	<ol style="list-style-type: none"> 1. 소환사명을 검색한다. 2. 검색된 소환사의 최신 정보가 출력된 것을 확인한다. 	유저가 개인의 소환사명을 검색하면 5초 내에 최신정보를 받아온다.	Success criteria 2

4. System Traceability



