

통합테스트					
순번	요구사항	확인 내용	테스트 확인	결과	
	1 한 사용자는 장소마다 리뷰를 1개만 작성할 수 있다.	(장소 정보를 저장한다.) 1. 로그인 한다. 2. 에디터에 내용을 작성한다. 3. 리뷰를 전송한다. 4. 같은 장소에서 리뷰를 다시 전송한다.	첫번째 리뷰는 저장이 되고 두번째 리뷰는 저장이 안된다.	O	
	리뷰는 수정 할 수 있다.	(장소 정보를 저장한다.) 1. 로그인 한다. 2. 작성한 리뷰를 로드한다. 3. 텍스트를 수정한다. 4. 수정 버튼을 누른다	수정된 텍스트가 저장되어야 한다.	O	
	리뷰는 삭제 할 수 있다.	(장소 정보를 저장한다.) 1. 로그인 한다. 2. 리뷰를 작성한다. 3. 리뷰를 삭제한다.	작성한 리뷰가 삭제되어야 한다.	O	
	사용자가 1자 이상의 텍스트를 작성하면 포인트 1점 부여한다.	(장소 정보를 저장한다.) 1. 로그인 한다. 2. 에디터에 텍스트만 입력한다. 3. 리뷰를 전송한다.	포인트가 1점 더해져야 한다. ex) 포인트가 2점이었을때 3점이 되어야 함	O	
	사용자가 1자 이상의 텍스트와 1개 이상의 이미지를 업로드하면 포인트 2점 부여한다.	(장소 정보를 저장한다.) 1. 로그인 한다. 2. 에디터에 텍스트를 입력하고 이미지를 업로드 한다. 3. 리뷰를 전송한다.	포인트가 2점 더해져야 한다. ex) 포인트가 2점이었을때 4점이 되어야 한다.	O	
	특정 장소에 첫 리뷰를 작성하면 보너스 점수 1점을 부여한다.	(장소 정보를 저장한다.) 1. 로그인 한다. 2. 에디터에 텍스트를 입력하고 이미지를 업로드 한다. 3. 리뷰를 전송한다.	장소 정보를 저장 할때 새로운 장소 등록 메시지를 확인 한다. 포인트가 3점 더해져야 한다. ex) 포인트가 2점이었을때 5점이 되어야 한다.	O	
	사용자마다 현재 시점의 포인트 총점을 조회하거나 계산 할 수 있어야 한다.	(장소 정보를 저장한다.) 1. 로그인 한다.	화면에 사용자의 현재 포인트 정보를 확인 한다.	O	
	리뷰를 작성했다가 삭제하면 해당 리뷰로 부여한 내용 점수와 보너스 점수는 회수	(장소 정보를 저장한다.) 1. 로그인 한다. 2. 사용자의 현재 포인트를 확인 한다. 3. 에디터에 텍스트를 입력한다. 4. 리뷰를 전송 한다. 5. 해당 리뷰를 삭제 한다.	1 포인트가 줄어 드는 지 확인 한다.	O	
	리뷰를 수정하면 수정한 내용에 맞는 내용 점수를 계산하여 점수를 부여하거나 회수	(장소 정보를 저장한다.) 1. 로그인 한다. 2. 에디터에 텍스트를 작성한다. 3. 리뷰를 전송한다. 4. 해당 리뷰에 그림을 추가 한다. 5. 리뷰를 전송한다. (수정버튼)	1 포인트가 늘어 나는지 확인 한다.	O	
	사용자 A가 리뷰를 남겼다가 삭제하고, 삭제된 이후 사용자 B가 리뷰를 남기면 사용자 B에게 보너스를 준다.	1. A로 로그인 한다. 2. 장소를 저장 한다. (새로운 장소라는 알림이 뜬다.) 3. 리뷰를 남겼다가 삭제 한다. 4. B가 로그인 한다. 5. 장소를 저장한다. (이미 등록된 장소라는 알림이 뜬다.) 6. 리뷰를 남긴다.	B에 보너스 포인트가 부여 된다. (장소는 이미 place table에 저장되어 있지만 해당 place에	O	
	사용자 A가 리뷰를 남겼다가 삭제하는데, 삭제되기 이전 사용자 B가 리뷰를 남기면 사용자 B에게 보너스 점수를 부여하지 않는다.	1. A로 로그인 한다. 2. 장소를 저장 한다. (새로운 장소라는 알림이 뜬다.) 3. B가 로그인 한다. 4. 장소를 저장한다. (이미 등록된 장소라는 알림이 뜬다.) 5. 리뷰를 남긴다.	B에 보너스 포인트가 부여 되지 않는다.	O	
	포인트 부여 API 구현에 필요한 SQL 수행 시, 전체 테이블 스캔이 일어나지 않는 인덱스가 필요		index 확인 완료	O	
	포인트 증감이 있을 때마다 이력이 남아야 한다.		DB table user_aud에서 포인트 히스토리를 확인 할 수 있다.	O	