

DATABASE APPLICATIONS

TERM PROJECT

<세줄 리뷰>

교육학과

2014190004 오형준

서비스개요서 : 서비스개요

서비스개요

세줄 리뷰는 세상의 모든 리뷰를 담은 서비스이다.

세줄 리뷰에서 사용자는 다양한 카테고리의 리뷰 대상들을 직접 만들 수 있고, 각각의 리뷰 대상들에 대해 리뷰(코멘트, 평)을 남길 수 있다.

세줄 리뷰에서의 리뷰는 집중적으로 탐색하여 쓴 글이라기 보다는 3~4줄의 간단한 평이다.

세줄 리뷰에서 사용자들은 리뷰를 남기므로써 포인트를 벌 수 있다. 이 포인트로 상품을 구매할 수 있다.

세줄 리뷰에서 기대하는 것은 사용자들이 자신의 활동을 평과 함께 기록하고, 그 기록을 다른 사용자들이 확인함으로써 더 좋은 선택을 유도하는 것에 있다.

요구사항 명세서 : 전사적

전사적 요구사항

- 어플리케이션으로 접근성을 높인다.
- 본 서비스에 가입된 회원에게만 서비스를 제공한다.
- 누구나 가입된 사람이라면 리뷰대상을 생성할 수 있다.
- 누구나 가입된 사람이라면 리뷰를 남길 수 있다.
- 이 시스템은 이용자에게 24시간 서비스를 제공한다.

요구사항 명세서 : 기능별

기능 명	회원 가입
<p>회원 가입은 회원으로 가입시키는 일을 말한다.</p> <p>회원에는 관리자와 일반의 두 종류의 회원이 있다.</p> <p>1. 일반 회원 가입</p> <p>일반 회원 가입에서는 3가지 정보를 받는다</p> <p>첫째, 이메일</p> <p>둘째, 비밀번호</p> <p>셋째, 이름</p> <p>필요시 비밀번호를 변경 할 수 있다.</p> <p>회원이 탈퇴할 시 회원 정보를 삭제한다.</p> <p>그이외의 회원 정보(가입일, 작성리뷰 개수, 긍정 리뷰수, 부정리뷰수, 보통리뷰수, 포인트, 레벨)는 자동으로 생성 될 수 있도록 한다.</p> <p>이메일은 중복해서 가질 수 없다.</p> <p>비밀번호는 8자 이상이 되어야 한다.</p> <p>비밀번호는 변경이 가능하지만 이외의 정보를 변경할 수 없다.</p> <p>회원 가입 후 로그인을 하지 않고 바로 서비스를 사용할 수있도록 한다.</p> <p>앱 종료후 다시 앱을 켜더라도 앱의 기능을 통해 기존 정보가 남아 있도록 한다.</p> <p>2. 관리자 회원 가입</p> <p>관리자 회원 가입은 직접 디비 입력을 통해 가입시킨다.</p> <p>관리자가 앱을 접속했을 때는 리뷰와 리뷰 대상을 관리할 수 있는 기능을 보여준다.</p> <p>관리자가 가지는 정보는 일반 회원의 정보와 같다.</p>	

관련 Entity Sets 및 Attributes	
Entity Set	Attributes
일반 회원	회원 번호, 회원 이름, 비밀번호, 가입일, 작성리뷰 개수, 긍정 리뷰수, 부정리뷰수, 보통리뷰수, 포인트, 레벨
관리자 회원	회원 번호, 회원 이름, 비밀번호

관련 Functions 및 Processes	
Function	Description
회원 가입	직원의 기본 정보를 입력

요구사항 명세서 : 기능별

기능 명	회원관리
<p>본 학원이 가입된 회원에게만 서비스를 제공한다.</p> <p>회원관리 기능은 회원의 정보를 관리하는 기능이다.</p> <p>가입된 회원의 정보를 입력한다.</p> <p>수정 요청 시 회원의 정보를 수정한다.</p> <p>탈퇴 요청 시 회원의 정보를 삭제한다.</p> <p>비정상적인 행위를 하는 회원을 탈퇴시킬 수 있다.</p> <p>회원의 정보를 조회할 수 있다.</p> <p>이 기능은 관리자만 사용할 수 있다.</p>	

기능 명	리뷰 대상 생성
<p>일반 회원은 리뷰 대상을 자유롭게 생성할 수 있다.</p> <p>리뷰 대상 생성은 리뷰를 작성할 수 있는 대상을 말한다.</p> <p>리뷰 대상 작성에는 다음의 정보를 입력해야한다.</p> <ol style="list-style-type: none">리뷰 대상 이름리뷰 대상 기본 정보리뷰 대상 카테고리리뷰 대상에 대한 리뷰 및 점수 <p>리뷰 대상은 중복될 수 있다.</p> <p>리뷰 점수는 0~10점 사이로 입력이 되어야 한다.</p> <p>기본 정보는 리뷰 대상에 대한 기본 정보를 의미한다.</p> <p>작성 내용은 생성자 명, 작성일과 함께 디비에 저장이 된다.</p>	

기능 명	리뷰 작성
<p>일반 회원은 리뷰 대상에 대한 리뷰를 작성할 수 있다.</p> <p>리뷰 대상 작성에는 다음의 정보를 입력해야한다.</p> <ol style="list-style-type: none">리뷰점수 <p>한 명의 회원은 한 리뷰 대상에 대해 하나의 리뷰만 작성 할 수 있다.</p> <p>회원은 자신이 작성한 리뷰에 대해 수정 및 삭제가 가능하다.</p> <p>리뷰 점수는 0~10점 사이로 입력이 되어야 한다.</p> <p>리뷰를 작성하면 작성일, 작성자와 함께 디비에 저장이 된다.</p>	

요구사항 명세서 : 페이지 별

회원 가입

회원가입

이메일

가입을 위해서 최소한의 정보만을 받는다

이메일은 이메일 형식이 아니면 오류를 출력한다.

비밀번호

비밀번호가 8자 미만이라면 오류를 출력한다.

8자 이상

이메일이 중복된다면 오류를 출력한다.

이름

문제가 없다면 회원을 디비에 등록한 뒤, 자동으로 로그인을 시킨 후에 홈으로 이동 시킨다.

가입

요구사항 명세서 : 페이지 별

로그인



세줄 리뷰

이미 가입한 회원에 대해서는 로그인을 통해 가입 시킨다.

이메일이 일치하는 것이 없으면 에러를 보내준다

비밀번호가 일치하지 않는다면 에러를 표시한다.

정상적인 이메일과 비밀번호라면 홈으로 보낸다

만약 관리자라면 관리자 페이지로 이동 시킨다.

이메일

비밀번호

[로그인](#)

[회원가입](#)



요구사항 명세서 : 페이지 별

홈

리뷰 대상 작성
페이지로 이동



홈에서는 리뷰 대상을 최신순으로 출력한다

리뷰 대상은 이름, 카테고리, 평균 점수, 긍정, 부정, 보통 평가수를 함께 출력한다.

평균 점수에 따라 배경 색상을 다르게 해준다.

리뷰 대상을 탭 하면 리뷰 대상 상세페이지로 이동시킨다.

우측상단의 + 버튼을 클릭하면 리뷰 대상 작성 페이지로 이동 시킨다.



리뷰 대상 상세 페이지로 이동

요구사항 명세서 : 페이지 별

리뷰 대상 작성 페이지

리뷰 대상 생성

리뷰 대상 생성

리뷰 대상 이름

name

대분류

Mountain View
Sunnyvale
Cupertino

Santa Clara
San Jose

소분류

Mountain View
Sunnyvale
Cupertino

Santa Clara
San Jose

기본 정보 작성

점수

리뷰 작성

완료

리뷰 대상 이름 을 작성한다

대분류를 선택하면 대분류에 따라 소분류가 바뀔 수 있도록 한다.

리뷰 대상에 대한 기본 정보를 작성한다

본인 이 생각하는 점수를 기록한다.

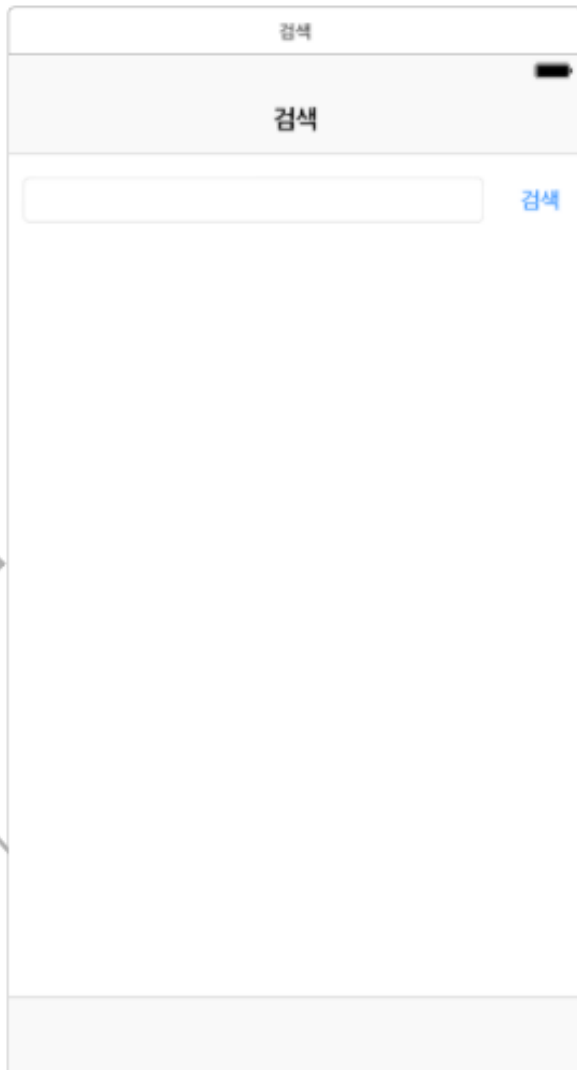
본인의 리뷰를 작성한다.

모든 내용을 작성한 뒤 완료를 누르면 디비에 저장한다.

완료 누르면 리로드 하며 이전페이지로 이동

요구사항 명세서 : 페이지 별

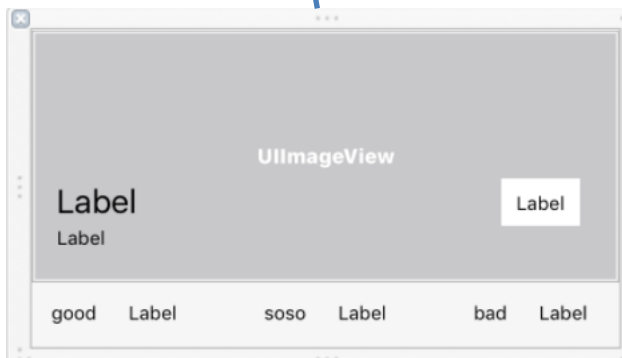
검색



버튼 누르면 검색 진행

상단 검색 바에서 키워드를 넣은 뒤 검색 버튼을 누르면 아래에 리뷰 대상 간략히 표시한 cell이 출력된다.

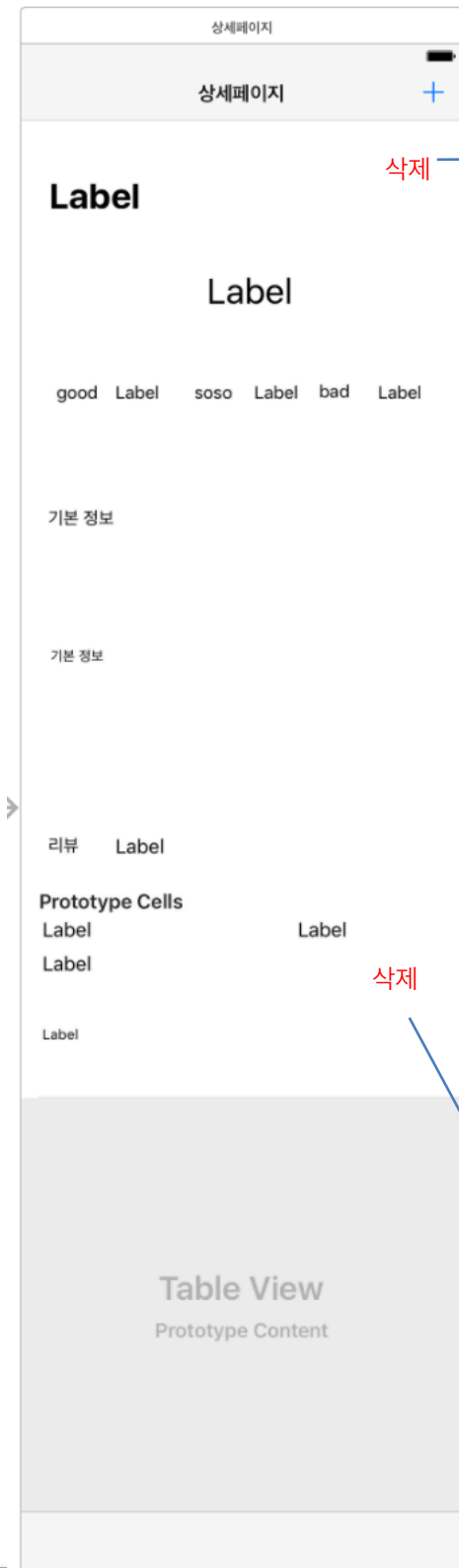
cell 을 터치하면 리뷰 대상 상세페이지로 이동시킨다.



리뷰 대상 상세 페이지로 이동

요구사항 명세서 : 페이지 별

리뷰 대상 상세 페이지



리뷰 작성 페이지로 이동

삭제

관리자로 접속하면 삭제 버튼이 보일 수 있도록 하며, 디비에서 리뷰 대상을 삭제하도록 한다.

최 상단에 리뷰대상 이름을 출력한다

그 아래 중간에 평균을 출력한다.(소수점 1자리)

그 아래 긍정, 보통, 부정 리뷰수를 출력한다.

그 아래 기본 정보를 출력한다.

그 아래 리뷰 대상에 대한 리뷰를 출력한다.

리뷰에는 생성 날짜, 작성자 이름, 리뷰 내용, 점수를 출력한다.

리뷰가 10개 이상이면 페이징을 통해 출력한다.

삭제

관리자로 접속하면 삭제 버튼이 보일 수 있도록 하며, 디비에서 리뷰를 삭제하도록 한다.

리뷰 대상 상세 페이지로 이동

요구사항 명세서 : 페이지 별

리뷰 작성 페이지

Write Review View Controller

Label

점수

작성완료

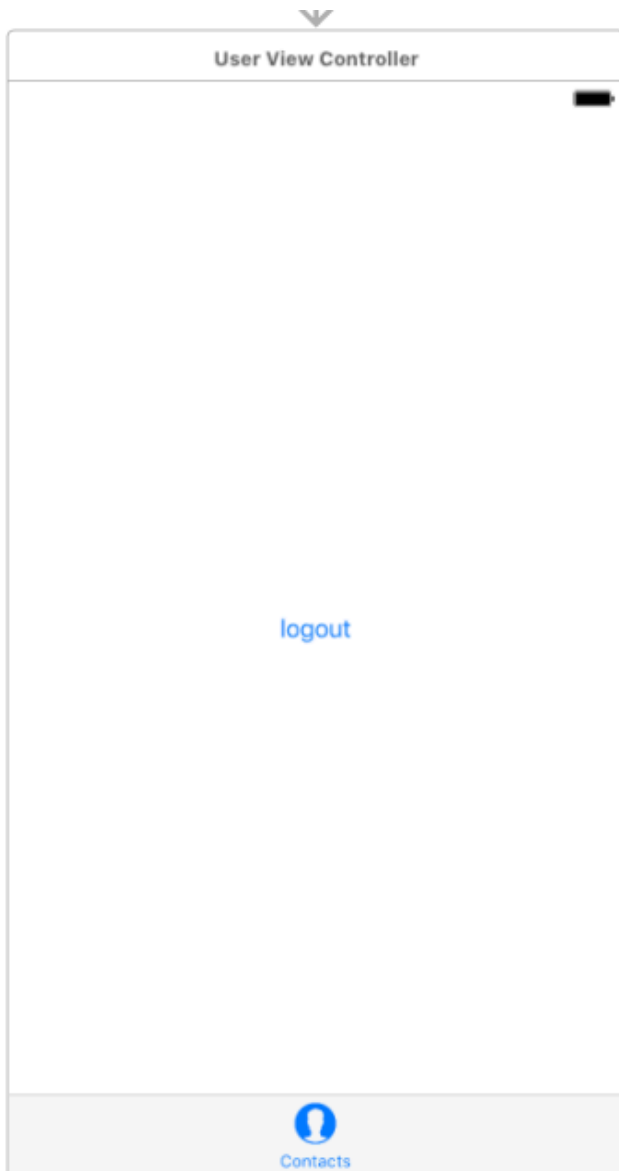
상단에는 작성하는 리뷰
대상을 출력한다.

본인이 생각하는 점수를 기록한 뒤
리뷰를 작성하도록 한다.

작성완료버튼을 누르면
디비에 저장한 뒤 페이지를
내리도록 한다.

요구사항 명세서 : 페이지 별

마이페이지

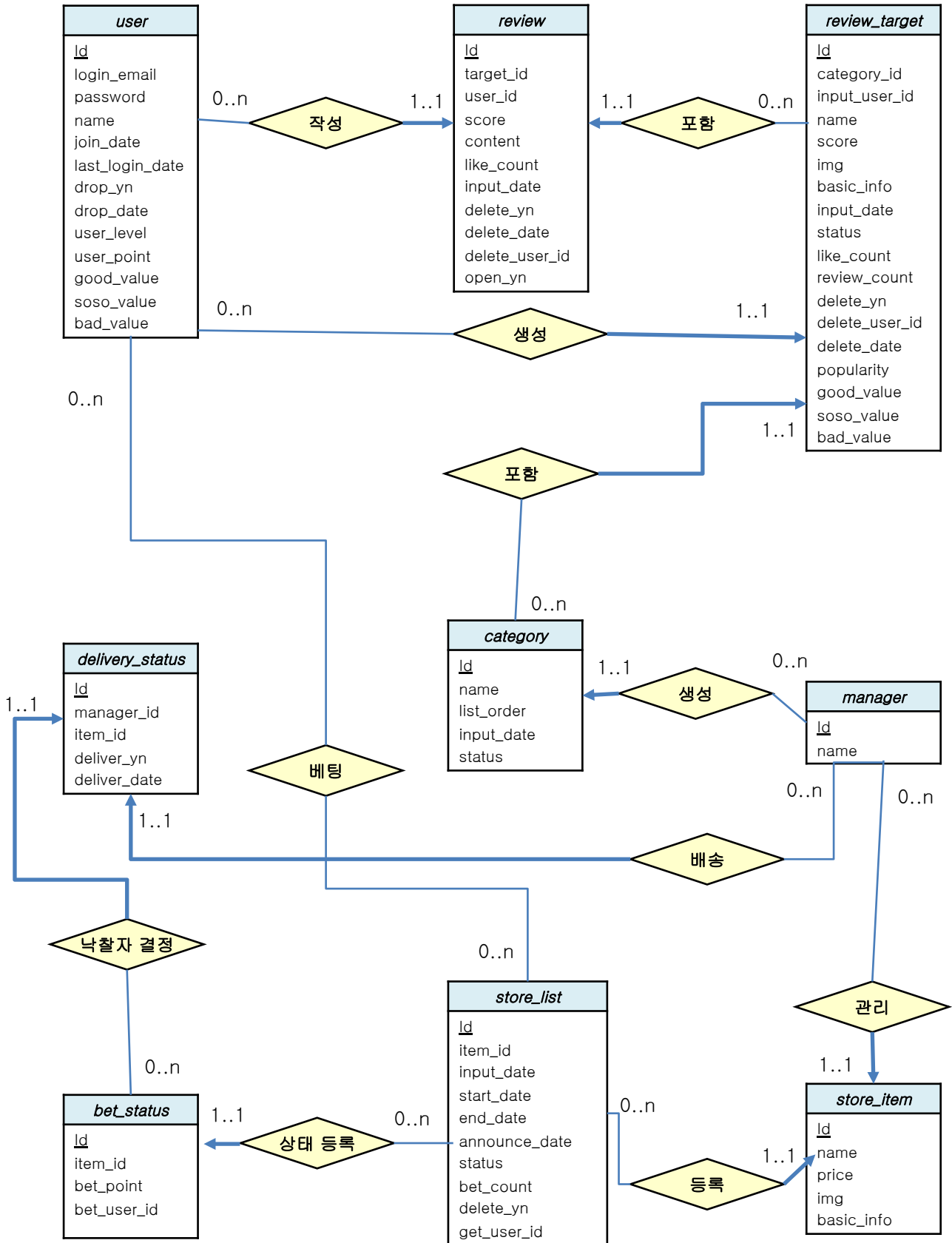


자신이 작성한 리뷰를
출력할 수 있도록 한다.

로그아웃 버튼을 누르면 로그아웃 시킨뒤
로그인 페이지로 이동시킨다.

앱의 userdefault 정보를 삭제한다.

ERD



스키마 정의 코드

User Table : 사용자 정보

```
CREATE TABLE `user` (
```

`id`	int	NOT NULL AUTO_INCREMENT,	## 인덱스(PK)
`login_email`	char(30)	NOT NULL,	## 로그인 이메일
`password`	varchar(100)	NOT NULL,	## 비밀번호
`name`	varchar(20)	NOT NULL,	## 이름
`join_date`	datetime	NOT NULL,	## 회원 가입일
`last_login_date`	datetime	DEFAULT NULL,	## 마지막 로그인 일시
`drop_yn`	char(1)	DEFAULT 'N',	## 탈퇴 여부
`drop_date`	datetime	DEFAULT NULL,	## 탈퇴 일시
`user_level`	smallint	DEFAULT 1,	## 회원 레벨
`user_point`	int	DEFAULT 1,	## 회원 포인트
`good_rate`	int	DEFAULT 0,	## 긍정 평가수
`soso_rate`	int	DEFAULT 0,	## 보통 평가수
`bad_rate`	int	DEFAULT 0,	## 부정 평가수
PRIMARY KEY (`id`)			
)			

Category : 카테고리

```
CREATE TABLE `category` (
```

`id`	int	NOT NULL AUTO_INCREMENT,	## 카테고리 id(10000 대분류, 100 중분류, 1 소분류)
`name`	varchar(50)	DEFAULT NULL,	## 카테고리 이름
`list_order`	smallint	NOT NULL,	## 순서
`input_date`	datetime	NOT NULL,	## 작성일
`status`	smallint	DEFAULT 0,	## 상태 0=미사용, 1=사용
PRIMARY KEY (`id`)			
)			

스키마 정의 코드

Review_Target: 리뷰 대상

```
CREATE TABLE `review_target` (  
  `id` int NOT NULL AUTO_INCREMENT, ## 인덱스  
  `category_id` int DEFAULT NULL, ## 카테고리 인덱스(FK)  
  `input_user_id` int DEFAULT NULL, ## 생성자 아이디  
  `name` varchar(50) DEFAULT NULL, ## 카테고리 이름  
  `score` float DEFAULT 0, ## 카테고리 점수  
  `img` varchar(255) DEFAULT NULL, ## 대표 이미지  
  `basic_info` text DEFAULT NULL, ## 기본 정보  
  `input_date` datetime NOT NULL, ## 작성일  
  `status` int DEFAULT 1, ## 상태 0=미사용, 1=사용, 2=신고  
  `like_count` int DEFAULT 0, ## 좋아요 수  
  `review_count` int DEFAULT NULL, ## 리뷰 수  
  `delete_yn` char(1) DEFAULT 'N', ## 삭제 여부  
  `delete_user_id` bigint DEFAULT NULL, ## 삭제자 아이디  
  `delete_date` datetime DEFAULT NULL, ## 삭제일  
  `popularity` bigint DEFAULT NULL, ## 인기도  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (category_id) REFERENCES category(id)  
)
```

Review : 리뷰

```
CREATE TABLE `review` (  
  `id` int NOT NULL AUTO_INCREMENT, ## 인덱스  
  `target_id` int DEFAULT NULL, ## 리뷰대상 인덱스  
  `user_id` int DEFAULT NULL, ## 작성자 아이디  
  `score` float NOT NULL, ## 리뷰점수  
  `content` text DEFAULT NULL, ## 내용  
  `like_count` int DEFAULT 0, ## 좋아요 수  
  `input_date` datetime NOT NULL, ## 작성일  
  `delete_yn` char(1) DEFAULT 'N', ## 삭제 여부  
  `delete_date` datetime DEFAULT NULL, ## 삭제일  
  `delete_user_id` bigint DEFAULT NULL, ## 삭제자 아이디  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (target_id) REFERENCES review_target(id),  
  FOREIGN KEY (user_id) REFERENCES user(id)  
)
```


스키마 정의 코드

Store_item: 스토어 물품 목록

```
CREATE TABLE `store_item` (  
  `id` int NOT NULL AUTO_INCREMENT, ## 아이템인덱스  
  `name` varchar(50) DEFAULT NULL, ## 상품 이름  
  `price` int DEFAULT 0, ## 상품 원가  
  `img` varchar(255) DEFAULT NULL, ## 대표 이미지  
  `basic_info` text DEFAULT NULL, ## 기본 정보  
  `input_date` datetime NOT NULL, ## 생성일  
  `start_date` datetime NOT NULL, ## 경매 시작 시간  
  `end_date` datetime NOT NULL, ## 경매 종료 시간  
  `announce_date` datetime NOT NULL, ## 경매 결과 발표 시간  
  `status` int DEFAULT 1, ## 상태 0=준비중, 1=경매중, 2=집계  
  중, 3 = 완료  
  `bet_count` int DEFAULT 0, ## 경매 참여자 수  
  `delete_yn` char(1) DEFAULT 'N', ## 삭제 여부  
  `get_user_id` int DEFAULT NULL, ## 상품 수령자 id  
  `store_maneger` text DEFAULT NULL, ## 상품관리자 정보  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (get_user_id) REFERENCES user(id)  
)
```

Bet_status:배팅 상태

```
CREATE TABLE `bet_status` (  
  `id` int NOT NULL AUTO_INCREMENT, ## 인덱스  
  `item_id` int DEFAULT NOT NULL, ## 아이템 인덱스  
  `bet_point` int DEFAULT NOT NULL, ## 베팅 포인트  
  `bet_user_id` int DEFAULT NOT NULL, ## 베팅한 사람  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (item_id) REFERENCES store_item(id),  
  FOREIGN KEY (bet_user_id) REFERENCES user(id)  
)
```

Join 결과 캡처

리뷰 대상에서 카테고리 이름을 가져오기 위한 Join
category와 review_target을 Join 시킴

```
SELECT review_target.id, category.name as category_id , review_target.name, score, basic_info, review_target.input_date, like_count, review_count, create_user_id, good_value, soso_value
FROM review_target left Join category
on review_target.category_id = category.id
ORDER BY review_target.id DESC;
```

id	category_id	name	score	basic_info	input_date	like_count	review_count	create_user_id	good_value	soso_value
21	지역서비스	카페 713	7	염창동의 작은 카페	2017-06-28 12:23:00	0	0	14	0	0
20	지역서비스	tset	5	Tset	2017-06-28 12:11:14	0	0	14	0	0
9	기타	인도 - 맥그로드 간즈	0	인도 히말라야 중턱에 위치한 마을. 티벳명정부가 있다.	2017-06-28 08:07:17	0	0	14	0	0
8	웹툰	복학왕	0	네이버 수요 웹툰 중 하나	2017-06-28 07:58:00	0	0	14	0	0
7	스포츠	롯데 자이언츠	0	KBO 프로야구 팀 중 하나	2017-06-28 07:54:59	0	0	14	0	0
6	TV	신서유기 4	0	나영석이 만든 TVN 예능, 강호동, 이수근 등 출연	2017-06-28 07:49:43	0	0	14	0	0
5	TV	신서유기 4	0	NULL	2017-06-28 07:44:27	0	0	14	0	0
4	문화	바람과 함께 사라지다	0		2017-05-14 17:43:33	0	NULL	1	0	0
3	문화	바람과 함께 사라지다	0		2017-05-14 16:49:51	0	NULL	1	0	0

Join 결과 캡처

검색 페이지에서
리뷰 대상에서 카테고리 이름을 가져오기 위한 Join
category와 review_target을 Join 시킴

```
SELECT review_target.id, category.name as category_id , review_target.name, score, basic_info, review_target.input_date, like_count, review_count, create_user_id, good_value, soso_value
FROM review_target left Join category
on review_target.category_id = category.id
ORDER BY review_target.id DESC;
```

id	category_id	name	score	basic_info	input_date	like_count	review_count	create_user_id	good_value	soso_value
21	지역서비스	카페 713	7	염창동의 작은 카페	2017-06-28 12:23:00	0	0	14	0	0
20	지역서비스	tset	5	Tset	2017-06-28 12:11:14	0	0	14	0	0
9	기타	인도 - 맥그로드 간즈	0	인도 히말라야 중턱에 위치한 마을. 티벳명정부가 있다.	2017-06-28 08:07:17	0	0	14	0	0
8	웹툰	복학왕	0	네이버 수요 웹툰 중 하나	2017-06-28 07:58:00	0	0	14	0	0
7	스포츠	롯데 자이언츠	0	KBO 프로야구 팀 중 하나	2017-06-28 07:54:59	0	0	14	0	0
6	TV	신서유기 4	0	나영석이 만든 TVN 예능, 강호동, 이수근 등 출연	2017-06-28 07:49:43	0	0	14	0	0
5	TV	신서유기 4	0	NULL	2017-06-28 07:44:27	0	0	14	0	0
4	문화	바람과 함께 사라지다	0		2017-05-14 17:43:33	0	NULL	1	0	0
3	문화	바람과 함께 사라지다	0		2017-05-14 16:49:51	0	NULL	1	0	0

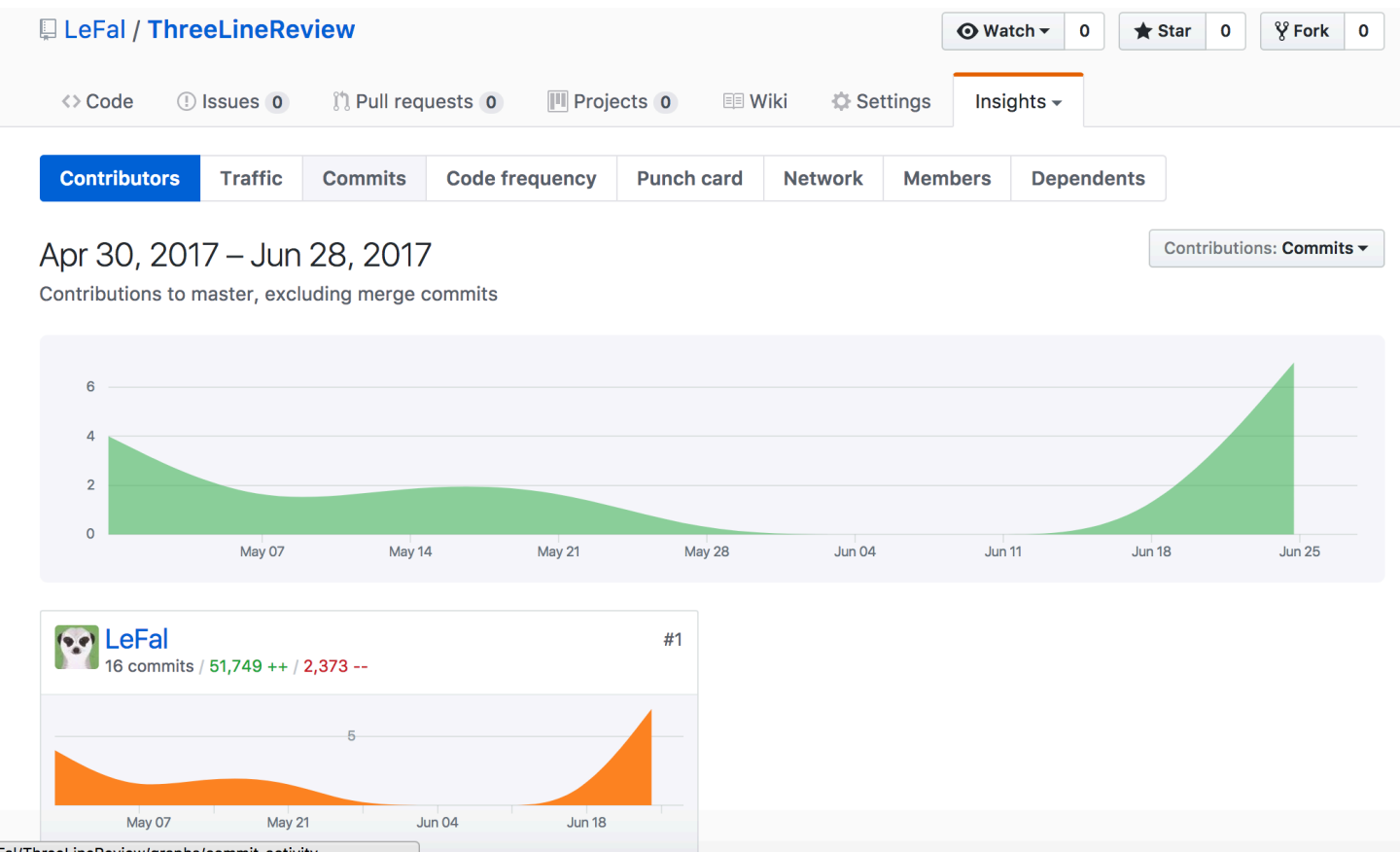
리뷰에서 작성자를 출력하기 위한 Join,
review를 기준으로 right Join

```
ORDER BY review_target.id DESC;
SELECT content, name as writer , score, target_id, input_date,review.id
FROM user right Join review
on user.id = review.user_id
order by review.id desc;
```

content	writer	score	target_id	input_date	id
오래오래	Leo	6	31	2017-06-28 18:57:25	36
55	Leo	5	31	2017-06-28 18:57:01	35
한반도	Leo	10	31	2017-06-28 18:48:07	34
Helloworld	Leo's	1	31	2017-06-28 13:25:35	33
	Leo's	3	30	2017-06-28 13:25:03	32
ㅇㅇ	Leo's	8	29	2017-06-28 13:24:43	31
Test	Leo's	4	28	2017-06-28 13:15:16	30
..	Leo's	9	26	2017-06-28 13:12:19	29
..	Leo's	9	26	2017-06-28 13:12:16	28
..	Leo's	2	25	2017-06-28 13:10:42	27

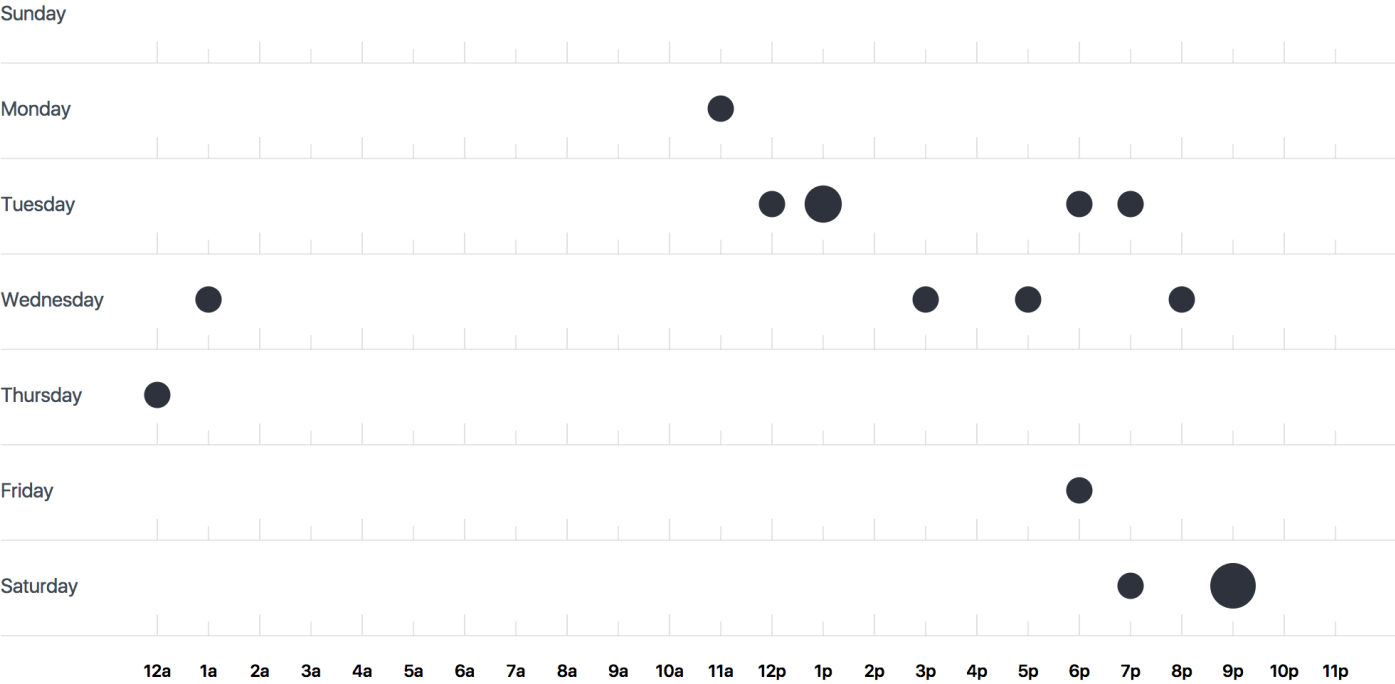
commit Log

<https://github.com/LeFal/ThreeLineReview>



commit Log

Contributors	Traffic	Commits	Code frequency	Punch card	Network	Members	Dependents
--------------	---------	---------	----------------	------------	---------	---------	------------



LeFal / ThreeLineReview

Watch 0

Star 0

Fork 0

<> Code

Issues 0

Pull requests 0

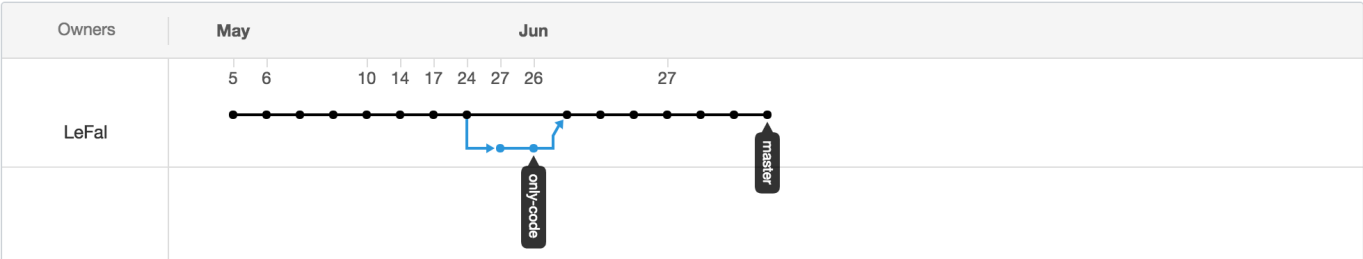
Projects 0

Wiki

Settings

Insights

Contributors	Traffic	Commits	Code frequency	Punch card	Network	Members	Dependents
--------------	---------	---------	----------------	------------	---------	---------	------------



요구사항 만족 확인

회원 가입

iPhone 7 - iOS 10.3 (14E8301)

Carrier 10:45 PM

회원가입

이메일

비밀번호

8자 이상

이름

가입

구현 완료

iPhone 7 - iOS 10.3 (14E8301)

Carrier 10:45 PM

회원가입

이메일

오류
중복된 이메일이 있습니다.

OK

이름

가입

iPhone 7 - iOS 10.3 (14E8301)

Carrier 10:46 PM

회원가입

이메일

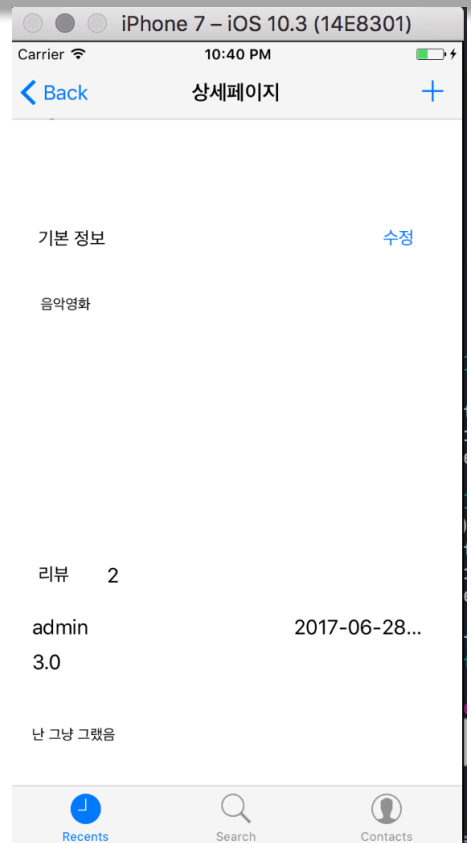
오류
패스워드가 올바르지 않습니다.

OK

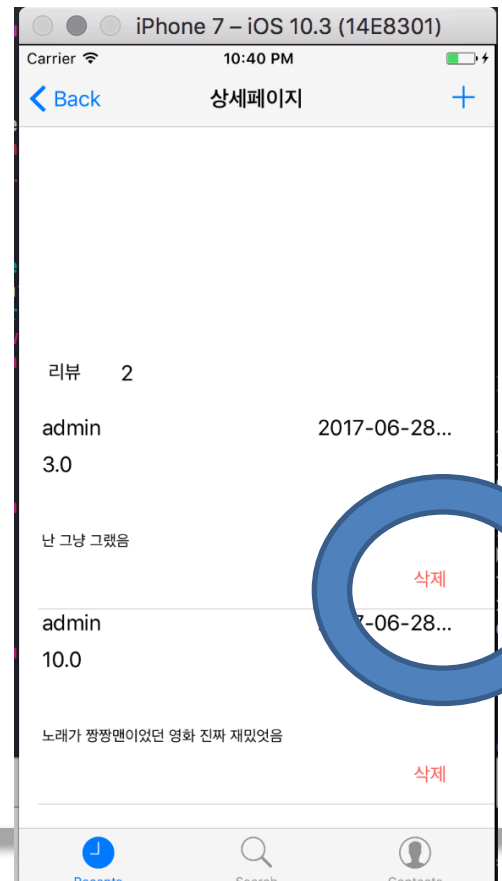
이름

가입

관리자 기능과 일반 사용자 기능



위 : 일반 사용자 로그인 / 아래 : admin 로그인



쿠키(iOS에서는 UserDefaults)

LoginViewController

```
case " success" :
    UserDefaults.standard.set(username, forKey: "id")
    if let gvc = self.storyboard?.instantiateViewController(withIdentifier:
        "mainVC") as? UITabBarController {
        self.present(gvc, animated: false)
    }
```

로그인 시 userdefault 저장

HomeController

```
if UserDefaults.standard.string(forKey: "id") == "" {
    if let gvc = self.storyboard?.instantiateViewController(withIdentifier: "LoginVC") as?
        LoginViewController {
        self.present(gvc, animated: false)
    }
}
```

home에서 UserDefaults 확인

(UserDefaults 는 앱이 종료되어도 값이 유지됨)

핵심 데이터 삽입

```
echo "success";
$sql = "INSERT into review_target (category_id, name, input_date, create_user_id,score,basic_info)
      values ('".$category_id."','".$name."',now(),'".$user_id."','".$score."','".$basic_info."')";
//echo($sql);
$result = mysqli_query($conn, $sql);

$sql = "SELECT id FROM review_target WHERE name = '".$name.'" AND basic_info = '".$basic_info.'";
$result = mysqli_query($conn, $sql);
$review_target_id = mysqli_fetch_row($result)[0];

$sql = "INSERT into review (target_id, user_id,score, input_date, content)
      values ('".$review_target_id."','".$user_id."','".$score."',now(),'".$review."')";
//echo($sql);
$result = mysqli_query($conn, $sql);
```

iPhone 7 – iOS 10.3 (14E8301)

Carrier 10:51 PM

리뷰 대상 생성

리뷰 대상 이름

name

대분류

여행
사회
지역
문화

소분류

국내
일본
중국
동남아

기본 정보 작성

리뷰 대상 생성(INSERT)

핵심 데이터 삽입

```
echo "success";  
$sql = "INSERT into review (target_id, user_id, input_date, content, score)  
      values (".$target_id.", ".$user_id.", now(), ' ".$content."', ".$score.")";  
//echo($sql);  
$result = mysqli_query($conn, $sql);
```

The screenshot shows a mobile application interface on an iPhone 7. At the top, the status bar displays 'Carrier', signal strength, time '10:52 PM', and battery level. The app header is '아는형님'. Below the header, there is a label '점수' (Score) followed by a text input field. Underneath this is a large rectangular text area for writing the review content. At the bottom of the screen, there is a blue button labeled '작성완료' (Finish Writing).

리뷰 생성(INSERT)

핵심 데이터 수정

```
<?php
require ("config/config.php");
require ("lib/db.php");
$conn = db_init($config["host"],$config["duser"],$config["psw"],$config["dname"]);
//echo "string";
if(!isset($_POST['target_id']) || !isset($_POST['basic_info'])) exit;

$target_id = mysqli_real_escape_string($conn, $_POST['target_id']);
$basic_info = mysqli_real_escape_string($conn, $_POST['basic_info']);

$sql = "UPDATE review_target SET basic_info = '". $basic_info ."'
      where id = ".$target_id;
$result = mysqli_query($conn, $sql);

echo "success";
exit();
?>
```

Carrier

10:53 PM



아는형님

JTBC 예능

수정

리뷰 기본 정보 수정(UPDATE)

핵심 데이터 삭제

```
<?php
require ("config/config.php");
require ("lib/db.php");
$conn = db_init($config["host"],$config["duser"],$config["psw"],$config["dname"]);
//echo "string";
if(!isset($_POST['target_id'])) exit;

$target_id = mysqli_real_escape_string($conn, $_POST['target_id']);

$sql = "DELETE FROM review where target_id = ".$target_id;
$result = mysqli_query($conn, $sql);

$sql = "DELETE FROM review_target where id = ".$target_id;
$result = mysqli_query($conn, $sql);

echo "success";
exit();
```

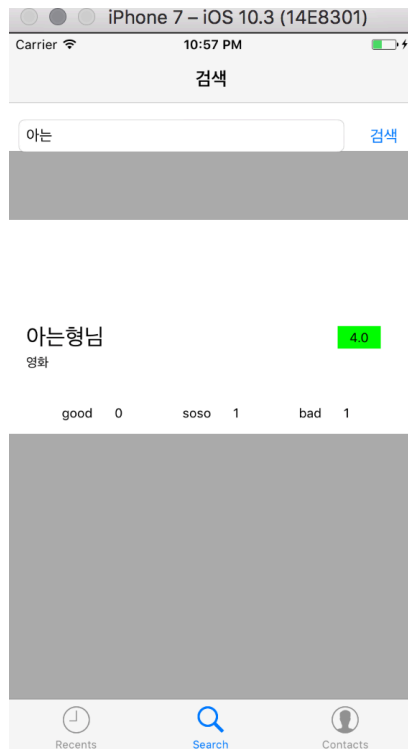


리뷰대상 삭제 (DELETE)

핵심 데이터 검색

```
@IBAction func search(_ sender: Any) {
    let keyword = searchBar.text
    fetchSearches(keyword: keyword!)
    self.collectionView.reloadData()
}

fileprivate func fetchSearches(keyword : String) {
    let parameters: [String: Any] = [
        "search": keyword
    ]
    Alamofire
        .request("http://localhost:8080/search.php", method: .post, parameters: parameters)
        .responseJSON { response in
            switch response.result {
            case .success(let value):
                guard let json = value as? [String: Any] else { return }
                let reviewTargetsJSONArray = json["data"] as? [[String: Any]] ?? []
                self.reviewTargets = [ReviewTarget](JSONArray: reviewTargetsJSONArray) ?? []
                self.collectionView.reloadData()
            case .failure(let error):
                print(error)
            }
        }
}
```



리뷰대상 검색 (SEARCH)

간략정보 & 상세정보



좌 : 간략데이터 / 우: 상세데이터

Join

```
$sql = "SELECT content, name as writer , score, target_id, input_date,review.id
FROM user right Join review
on user.id = review.user_id
where target_id = ".$target_id."
order by review.id desc;";
```

ORDER BY review_target.id DESC;

```
SELECT content, name as writer , score, target_id, input_date,review.id
FROM user right Join review
on user.id = review.user_id
order by review.id desc;
```

100% 37:13 1 error found

content	writer	score	target_id	input_date	id
오래오래	Leo	6	31	2017-06-28 18:57:25	36
55	Leo	5	31	2017-06-28 18:57:01	35
한반도	Leo	10	31	2017-06-28 18:48:07	34
Helloworld	Leo's	1	31	2017-06-28 13:25:35	33
	Leo's	3	30	2017-06-28 13:25:03	32
ㅇㅇ	Leo's	8	29	2017-06-28 13:24:43	31
Test	Leo's	4	28	2017-06-28 13:15:16	30
..	Leo's	9	26	2017-06-28 13:12:19	29
..	Leo's	9	26	2017-06-28 13:12:16	28
As	Leo's	9	26	2017-06-28 13:10:42	27

리뷰에서 사용자 이름을 가져오기 위한 JOIN

리뷰대상에서 카테고리를 가져오기 위한 JOIN

```
$sql = "SELECT review_target.id, category.name as category_id , review_target.name, score, basic_info, review_target
.input_date, like_count, review_count, create_user_id, good_value, soso_value, bad_value, review_target.status,
img
FROM review_target left Join category
on review_target.category_id = category.id
ORDER BY review_target.id DESC;";
```

```
SELECT review_target.id, category.name as category_id , review_target.name, score, basic_info, review_target.input_date, like_count, review_count, create
FROM review_target left Join category
on review_target.category_id = category.id
ORDER BY review_target.id DESC;
```

100% 45:6

id	category_id	name	score	basic_info	input_date	like_count	review_count	create_user_id	good_value	soso_value
21	지역서비스	카페 713	7	염창동의 작은 카페	2017-06-28 12:23:00	0	0	14	0	0
20	지역서비스	tset	5	Tset	2017-06-28 12:11:14	0	0	14	0	0
9	기타	인도 - 맥그로드 간츠	0	인도 히말라야 종턱에 위치한 마을. 티벳명정부가 있다.	2017-06-28 08:07:17	0	0	14	0	0
8	웹툰	북학왕	0	네이버 수요 웹툰 중 하나	2017-06-28 07:58:00	0	0	14	0	0
7	스포츠	롯데 자이언츠	0	KBO 프로야구 팀 중 하나	2017-06-28 07:54:59	0	0	14	0	0
6	TV	신서유기 4	0	나영석이 만든 TVN 예능, 강호동, 이수근 등 출연	2017-06-28 07:49:43	0	0	14	0	0
5	TV	신서유기 4	0	NULL	2017-06-28 07:44:27	0	0	14	0	0
4	문화	바람과 함께 사라지다	0		2017-05-14 17:43:33	0	NULL	1	0	0
3	문화	바람과 함께 사라지다	0		2017-05-14 16:49:51	0	NULL	1	0	0