### Project Description, Requirements, Business Rules :

### Description: This application is for users to find rock climbing gyms, outdoor locations, and places to purchase equipment. The users should be able to sign in and add reviews for these locations and add routes. Users can also suggest new gyms, and stores to be approved by an admin. Users can add outdoor locations without admin approval because almost anywhere can be an outdoor location and it doesn’t need to be as heavily regulated.

### Requirements:

1. Users can make accounts with passwords.
2. Users can log in and out.
3. Users can view stores, gyms, and outdoor sites without being logged in.
4. Users can post routes for outdoor locations and gyms.
5. Users can suggest new gyms and stores.
6. Users can add outdoor locations
7. Users can leave reviews.

**Business Rules:**

1. Users can not add gyms or stores themselves.
2. You must log in to post reviews or suggest new additions.
3. You can only rate a location once, trying to add another will overwrite the old review.
4. You do not need to log in to see any of the information.

### Database Design – 10 points

### ER Diagram & Relational Model:

### https://lh4.googleusercontent.com/h4ACiFIA6hk06I4wUxvjxwbvz6tPxVCxzV1N3npUvvhse9WcE75Ca1160cR_1vh_w2ojLlHF5z_BtT1Hdf-UHrh4abKNxaIOBGRlyUqLJe45ox5d3XZpnIzRDY8z0EBhh0NOPfwR

### https://lh4.googleusercontent.com/Tt_zJV2b5lR9OgNYzq7GZST8yeJSfd0omX8LQ58igEuZqYScnTLutCbo0I9PoTSmjfnSyoYM_I7AhB7pCtebqg89KDNIw6OYhDrLYxybhDBGB8fL7NMUguf2__c93Fh1yjN4aRsp

MySQL database tables, views, at least two trigger/procedure/function

**Procedure:**

DELIMITER $$

CREATE DEFINER=`cs340\_zaengled`@`%` PROCEDURE `publishGymOrStore`(IN `object` INT(10))

NO SQL

BEGIN

IF(SELECT gymID FROM Gym WHERE object = gymID) is true THEN

UPDATE Gym SET published = 1;

END IF;

IF(SELECT storeID FROM Store WHERE object = storeID) is true THEN

UPDATE Store SET published = 1;

END IF;

END$$

DELIMITER ;

**Trigger:**

CREATE TRIGGER `updateRating` AFTER INSERT ON `Reviews`

FOR EACH ROW UPDATE Location SET rating = (SELECT ROUND(AVG(stars),2) FROM Reviews WHERE Reviews.objid = new.objid) WHERE Location.objid = new.objid

**Foreign keys:**

Route, Site, Gym, Store, and Reviews all use Location’s objid as a foreign key. Admin and Reviews use User’s username as a foreign key.

**Primary keys are:**

Location: objid

Site: siteID(fk)

Gym: gymID(fk)

Store: storeID(fk)

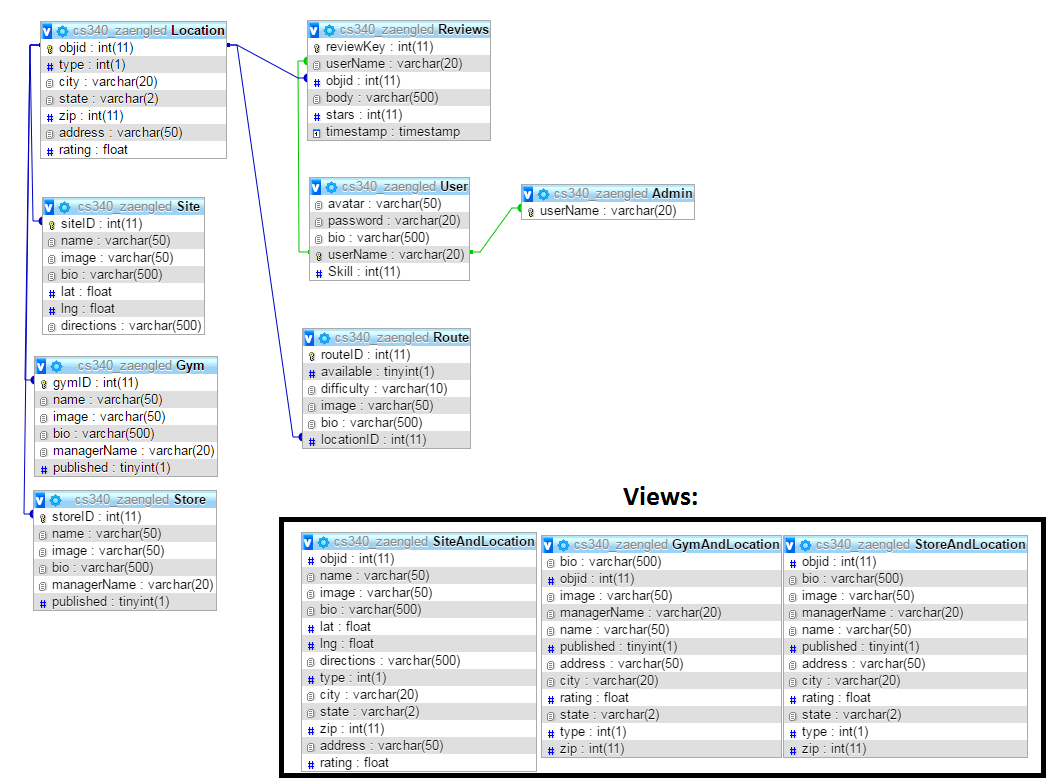
Reviews: reviewKey (autoincremented)

User: userName

Route: routeID(autoincremented)

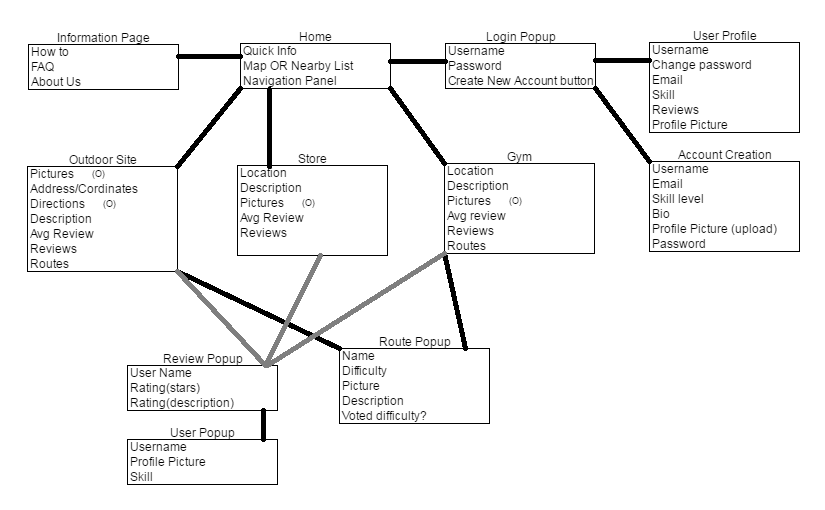
Admin: username(fk)

**Diagram of database tables, pulled from phpMyAdmin using the designer tab.**

****

### Website Design, Layout & User manual – 15 points

### Design: We used Bootstrap to implement the sidebar for navigation. A header and footer included on every page provide persistent navigation no matter where you are on the site.

**Website Layout:**

**To use the site:**

**HOME**:You will start on the homepage which will display nearby locations when you type them in the City box and click submit. Use these cities to get results: Corvallis, Eugene, Susanville. If you add a new location using a different city later it will show up here.

**ANY PAGE:** On any page you can use the navigation bar on the left to move around the website. The gym/outdoor/store links will take you to a list of all the entries of that type in the database and let you suggest a new one if you are logged in.

**LOGGING IN:** In order to log in, click the “login” button in the top right corner. From there you can create a new account or log in with a current one.

**ADMIN LOGIN:** Because admins have the ability to approve suggested gyms and stores we wanted to make sure you could log in and view that as well. Use the account name: **Zaengru** and password: **bad** in order to log in as an admin.

**GYM or OUTDOOR or STORE list pages:** These pages display a full list of every approved location of that type. From these pages you can also suggest a new gym using the “suggest a gym” button at the bottom of the list.

**GYM or OUTDOOR or STORE detail pages:** These pages display the details of the location you have selected. These pages can be reached by clicking a location on the front page OR from any one of the list pages. To view reviews or routes left by other users you can click them and a modal will pop up with all the information they left. From these pages AS A LOGGED IN USER you can leave reviews for any location and add new routes (with optional pictures) for gyms and outdoor sites.

**ABOUT PAGE:** This page just contains some simple instructions and answers to questions we think users might ask.

Application Functionality & Requirements: - 65 points

Below list at least six features of your website and how to test them. All input must be error checked. Functionality implemented without SQL can be listed for credit; however, you must have at least five SQL queries including at least one insert or update generated from user input and at least two complex queries involving multiple tables.

1. To search for locations in a specific city go to the main page and input a city name. Some of the ones that will give results are: Corvallis, Eugene, and Susanville. Please note this input field will strip anything that is not a number or a letter from your input. The text box runs an AJAX query which uses the following

SQL:

"SELECT \* FROM `GymAndLocation` WHERE City = '".$q."'"

Where GymAndLocation is a view combining both the gym table and the Location table. The querys for Site and Store look the same but call a different view. $q is passed in via a GET from the javascript function

1. To see a full list of Outdoor sites/Gyms/Stores click one of those 3 from the menu on the left. The pages use the following SQL respectively:

"SELECT \* FROM `SiteAndLocation`" In site\_list.php

"SELECT \* FROM `GymAndLocation`" In gym\_list.php

"SELECT \* FROM `StoreAndLocation`" In store\_list.php

1. To see details of any Outdoor site/Gym/Store click one of the entries on the preivous page. The pages use the following SQL:

gym\_detail.php

"SELECT \* FROM `GymAndLocation` WHERE `objid` = " . $\_GET["objid"]

"SELECT \* FROM `Reviews` WHERE `objid` = " . $\_GET["objid"]

"SELECT \* FROM `Route` WHERE `locationID` = " . $\_GET["objid"]

site\_detail.php

"SELECT \* FROM `SiteAndLocation` WHERE `objid` = " . $\_GET["objid"]

"SELECT \* FROM `Reviews` WHERE `objid` = " . $\_GET["objid"]

"SELECT \* FROM `Route` WHERE `locationID` = " . $\_GET["objid"]

store\_detail.php

"SELECT \* FROM `StoreAndLocation` WHERE `objid` = " . $\_GET["objid"]

"SELECT \* FROM `Reviews` WHERE `objid` = " . $\_GET["objid"]

"SELECT \* FROM `Route` WHERE `locationID` = " . $\_GET["objid"]

1. To submit a new route WHILE LOGGED IN, from any gym\_detail or site\_detail page: click add route. This goes to submitroute.php which when filled out will send its data via post to route\_upload.php which uses the following SQL:

"INSERT INTO `Route` (`locationID`, `bio`, `difficulty`, `image`, `available`) VALUES ('$OBJID', '$CONTENT', '$DIFFICULTY', '$IMAGE', 1)"

1. To submit a new review WHILE LOGGED IN, from any gym\_detail, site\_detail, or store\_detail page: click add review. This goes to submitreview.php which uses the following SQL to get info about the user:

"SELECT \* FROM `GymAndLocation` WHERE `objid` = ".$\_GET["objid"]

(this uses StoreAndLocation or SiteAndLocation too if you go from those pages)

It fills out the form with user info using that SQL then submits whatever the user inputs using this SQL:

"DELETE FROM `Reviews` WHERE `Reviews`.`objid` = '$OBJID' AND `Reviews`.`userName` = '$USERNAME'" (This makes sure there are no duplicate reviews.)

"INSERT INTO `Reviews` (`userName`, `objid`, `body`, `stars`) VALUES ('$USERNAME', '$OBJID', '$CONTENT', '$STARS')"(This submits the new review after it gets rid of any past reviews that have the same objid and username)

1. To create an account or log in, use the “login” link in the upper right hand corner. Both use login.php and creating a new account will use the SQL:

"INSERT INTO User (userName, password) VALUES('$\_POST[username]', '$\_POST[password]')"

While logging in will use this SQL:

"SELECT \* FROM `User` WHERE `userName` = '$username' AND `password` = '$password'"

Finally, to check if you are an admin this query will be run:

"SELECT \* FROM Admin WHERE userName='$row[userName]'"

Code is in the files submitted to Teach as requested. So I did not include it here which we discussed to be OK in class Friday the 2nd I believe.

Total 100 points

What to submit to Canvas. DO NOT ZIP!

* Project Report
* Grade Sheet
* Screenshots of the content and structure of your database tables.
* In comments URL & onid username of the student that submitted to TEACH

What to submit to TEACH. (If you are working in a group submit a single copy)

* Zipped folder containing all files for website and all exported SQL files for your database.