

# 2023 年度ウェブプログラミング実習 総合実習レポート (公園さ・が・し！！)

班番号	A	研究室名	森勢研
組	4	番号	5
氏名	岩月 雷英		
提出日時	2024 年 1 月 17 日		
共同実習者名	石田翔		
	水上遼人		
	古島妙		

## 1. 作成概要

本課題で作成したシステム「公園さ・が・し！！」は、公園に特化した口コミサイトである。バーベキューや球技の練習など、何らかの目的をもって公園を探している人が利用することを想定しており、公園データの追加・検索の他、地図表示やデータ編集などを行える。データの追加や編集にはユーザー登録が必要であるが、公園一覧と各公園情報の閲覧はログインなしで利用できる。本作品では、ユーザーが送信したデータを全てサーバーのデータベースに登録し、書き込み、読み込みは全て **SQL** を用いて行っている(図 1)。

ログイン画面では、登録したユーザー情報を入力したログインの他、サインアップから新しくアカウント登録を行うこともできる。ログイン送信が完了したら、公園一覧画面に遷移する。

公園一覧画面では、都道府県ごとに公園の一覧を表示し、公園名をクリックすることで、各公園情報画面に遷移することができる。この時、検索機能を用いて公園を絞り込むこともでき、例えば「**BBQ**」と検索すると、情報内に「**BBQ**」の単語を含んでいる公園のみを表示することができる。

公園情報画面では、ユーザーが登録した情報の他、地図上の位置表示や、他のユーザーからのレビューも見ることができる。ログインしたユーザーは、このページから、新たにレビューを投稿したり、既に登録されている情報を編集したりすることもできる。

公園追加画面では、ログインしたユーザーが、公園名や住所、その他の情報などをフォー

ムに入力し、サーバーのデータベースに登録することができる。住所登録では郵便番号から住所検索する機能が備わっている。公園編集画面においても同様のフォームを表示し、ユーザーは情報を編集し、登録しなおすことができる。本システムでは Google Maps API を使用しており、追加された住所情報をもとに地図を表示する。

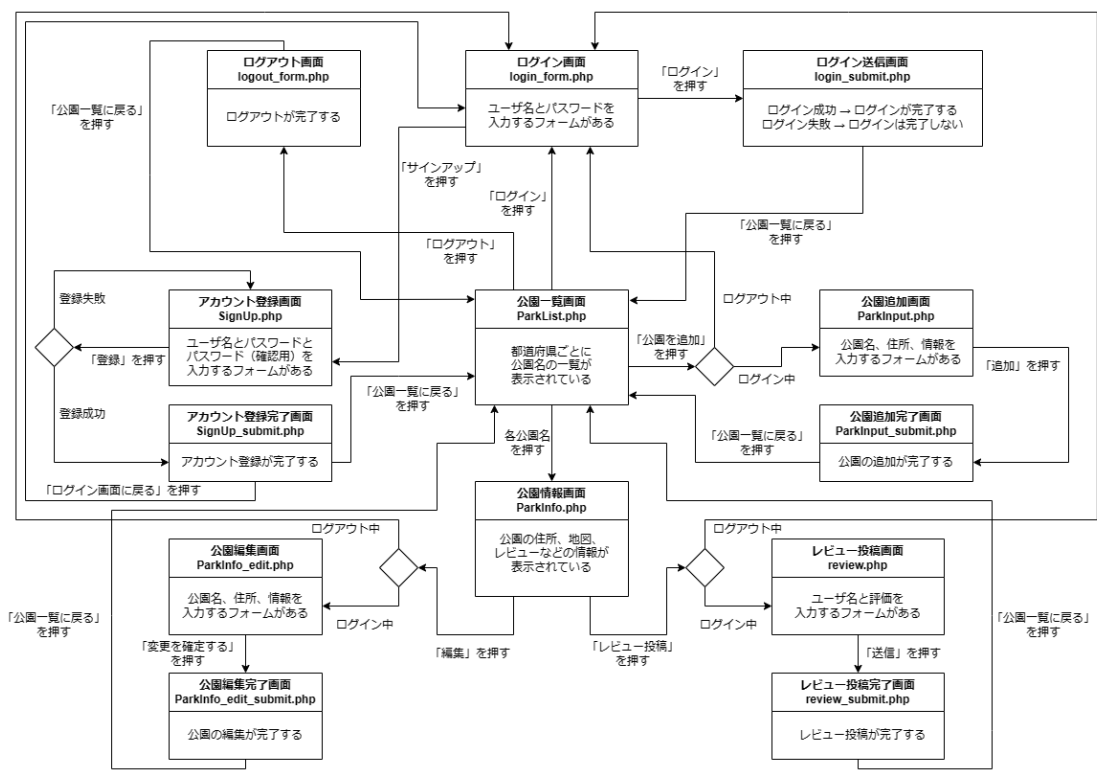


図 1 システム画面遷移図

2. 担当箇所について

表 1 グループ内での担当箇所

担当者	備考
石川翔	公園データ追加、プレゼン用パワーポイントの作成、発表
水上遼人	CSS、郵便番号検索機能の作成
岩月雷英	Google Maps API、公園データの編集機能、レビュー機能、ログイン機能、サインアップ機能
古島妙	SQL データベースの作成、公園データ入力のプロトタイプ、公園データ表示のプロト

	タイプ、公園一覧ページの作成
--	----------------

```
<img src = https://maps.googleapis.com/maps/api/staticmap?
center=".h($parklist["pref"])."."h($parklist["address"]).",CA&zoom=16&size=400x400&scale=2
&key=AlzaSyDx2HM7XyxyLnRJL9N22VIcdvO_pbZgQkw class='image'></img>";
```

図 2 API のソースコード

都道府県 ▼

市区町村以降の住所

図 3 住所入力欄

```
sqlite> .schema
CREATE TABLE parklist(id integer primary key autoincrement, parkname text, pref text, data text, address
text);
CREATE TABLE review(id integer primary key autoincrement, park_id integer, name text, rate integer, body
text);
CREATE TABLE user(id integer primary key autoincrement, username text, password text);
```

図 4 各テーブルのスキーム

```
$st = $pdo->query("SELECT * FROM review WHERE park_id = '$parklist[id]' ORDER BY id DESC");
$data2 = $st->fetchAll();
print "<h2>レビュー</h2>";
foreach($data2 as $review) {
    $sum = $sum + h($review["rate"]);
    $count = $count + 1;
    print '<h3>星' . h($review["rate"]) . '</h3>';
    print '<p>' . h($review["name"]) . '</p>';
    print '<p>' . h($review["body"]) . '</p>';
    print '</div>';
}
if($sum!=0 && $count !=0)print '<h3>平均:星' . $sum/$count . '</h3>';
```

図 5 レビューの平均値のソースコード

```

if(empty($username))
{
    header("Location: SignUp.php?error=3");
    exit;
}
else if(empty($passwd) || empty($passwd2))
{
    header("Location: SignUp.php?error=5");
}
else if ($passwd != $passwd2){
    header("Location: SignUp.php?error=2");
    exit;
}

else if ($user_on_db)
{
    header("Location: SignUp.php?error=4");
    exit;
}
else if(!$user_on_db && $passwd == $passwd2) {
    $st2 = $pdo->prepare("INSERT INTO user(username, password) VALUES(?, ?)");
    $st2->execute(array($username, $passwd));

    $result = "登録しました。";
}
else{
    header("Location: SignUp.php?error=1");
    exit;
}

```

図 6 サインアップ時のエラー処理のソースコード

私は共同実習者の古島妙が作成したプロトタイプをもとに改良と機能追加をした。追加した機能は Google Maps API、公園データの編集機能、レビュー機能、ログイン機能、サインアップ機能の 5 つである。レビュー機能、ログイン機能、サインアップ機能に必要な review、user テーブルを作成した(図 4)。

Google Maps API は公園情報画面(ParkInfo.php)で公園の住所を地図で表示する機能である。公園追加時にデータベースに書きこんだ住所情報をもとに API を呼び込むことで地図を表示している(図 2)。工夫した点は公園追加時のデータをそのままデータベースから取り出すだけで API から地図を表示できるように、住所入力欄のフォーマットが整えられている点である(図 3)。都道府県は parklist の pref に、市区町村以降の住所は parklist の address に書き込まれる(図 4)。Google Maps Platform の center パラメータの説明より「文字列の住所(例: city Hall, new york, ny)で、地表上の一意の場所を示します」<sup>1)</sup>と書いてあるため、データベースの住所情報を緯度経度に変換せず地図を表示できる。苦労した点は API の仕

様を学ぶ必要があった点である。

公園データ編集機能は公園編集画面(ParkInfo\_edit.php)で公園情報を後から変更することができる機能である。例えば、”公園情報の中に BBQ 可能と書かれていたが、本当は禁止だった”、”住所が間違っていた”など情報を修正、追加したい時に使う機能である。公園修正画面は公園追加画面と同じ UI である。工夫した点は公園編集画面を開いたとき、入力欄には修正前の情報が既に入力されている点である。これにより修正したくない部分まで自分で書き直す必要がなくなり、簡単に編集をすることができる。苦労した点は公園情報追加のときは parklist テーブルに INSERT コマンドで書き込んでいたが、編集時は UPDATE コマンドを使う必要があるということを忘れていたためにエラー修正に時間がかかった点である。SQLite 入門より「テーブルに格納されているデータを新しい値に更新するには UPDATE 文を使います。」<sup>2)</sup>と学んだためこの問題を解決することができた。

レビュー機能(review.php)はユーザーが公園を星 1~5 で評価し、コメントすることができる機能である。この機能によってユーザーがどの公園が良い公園なのか知ることができる。レビュー内容は review に書き込まれる。どの公園に対するレビューなのか判別するために park\_id というカラムがある(図 4)。工夫した点はレビューの点数の平均値を出すようにした点である。レビュー内容を 1 つずつ表示すると同時に点数の合計を求めて、最後に総レビュー数で割ることで平均値を求めている(図 5)。

ログイン機能(login\_form.php)はユーザーに公園追加、公園情報編集、レビューを許可するための機能である。ログインしていない状態でこれらの機能を使用すると強制的にログイン画面へ遷移する。工夫した点はパスワードが間違っている場合やユーザーが存在しない場合にはエラーを表示するという点である。これによりユーザーがパスワードを間違えたのか、そもそもサインアップしていなかったのか理解できる。

サインアップ機能(SignUp.php)はユーザーを新規登録するための機能である。ユーザー情報は user に書き込まれる(図 4)。パスワードは確認のために 2 回書かせるようにしている。工夫した点は細かくエラーの分類を分けた点である。エラー処理は SignUp\_submit.php で行われている。ユーザー名が空欄だった場合、パスワードが入力されていない場合、パスワードが 1 回目と 2 回目で違う場合、すでに登録済みだった場合、自動で SignUp.php に戻りそれぞれ別のエラー文が表示される(図 6)。これによりユーザーはどうしてエラーになったか理解できる。苦労した点はユーザー名やパスワードを空白で入力されたときのエラー処理がうまくできなかった点である。単なる空白であれば empty としてエラー処理できるが、スペースキーを入力された場合の空白を上手く処理することができなかった。しかし原因は特定できないがスペースキーのみで登録されたユーザーは、データベースには情報が書き込まれているのにログインをすることはできなかった。

## 参考文献

- 1) Google Maps Platform, Maps Static API, (<https://developers.google.com/maps/documentation/maps-static/start?hl=ja#Locations>, 2024 年 12 月参照)
- 2) Tasuo Ikura, SQLite 入門, (<https://www.javadrive.jp/sqlite/>, 2024 年 12 月参照)