Assignment 2 Report

All of the criteria items for this assessment has been successfully fulfilled except for *pagination, Image,* and *Recommendation*. The approach used to design this assignment product was to create migrations and seeder and model files for each table schema to check the functionality of the controller through routes in web.php as shown below.

```
Route::get('/follow', function () {
    $follows = Follow::all();
    //dd($votes);
    foreach ($follows as $follow){
        echo "User Name:  {$follow->user->name} follows {$follow->follows->name} <br>";
        echo "<br><br>";
    }
});

// https://s2921450.elf.ict.griffith.edu.au/webAppDev/week9(20)/review/public/follow2
Route::get('/follow2', function () {
    $users = User::all();
    foreach($users as $user){
        $id = $user->id;
        $following = Follow::whereHas('user',function($query)use($id){
            return $query->whereRaw('user_id = ?',array($id));
        })->get();
        echo "{$user->name} is following:<br>";
        foreach($following as $follow){
            echo "<ul>";
                echo "<li>{$follow->follows->name}</li>";
            echo "</ul>";
        }
    }
});

Route::get('/follow3', function () {
    $users = User::all();
    foreach($users as $user){
        // $id = $user->id;
        // $following = Follow::whereHas('user',function($query)use($id){
        //    return $query->whereRaw('user_id = ?',array($id));
        // })->get();
        echo "{$user->name} is following:<br>";
        foreach($user->following as $follow){
            if(is_null($follow)){
                echo "<p>Not following anyone</p>";
            }else{
                echo "<ul>";
                echo "<li>{$follow->name}</li>";
                echo "</ul>";
            }
        }
    }
});
```

In order to effectively fulfil the criteria the design method chosen in this assignment was to create relational functions that use eloquent ORM capability rather than store related data in every table. The approach chosen in this assignment is illustrated by the code snippets below.

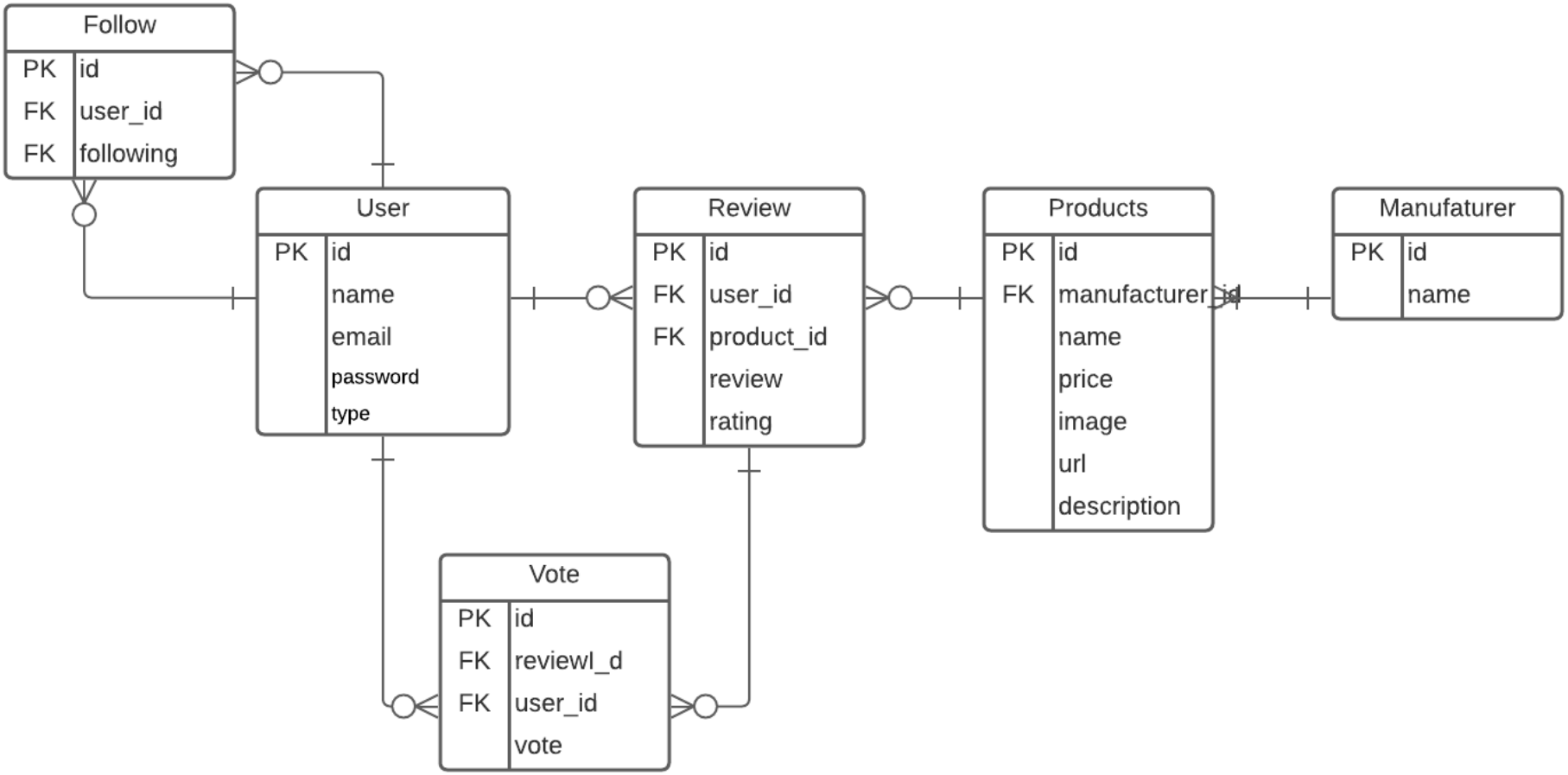
```php
Route::get('/votes0', function () {
    $votes = Vote::all();
    //dd($votes);
    foreach ($votes as $vote){
        echo "Review ID:  {{$vote->review->user->id}} <br>";
        echo "Reviewer Name:  {{$vote->review->user->name}} <br>";
        echo "Comments:  {{$vote->review->review}} <br>";

        echo "Voter ID:  {{$vote->review->user_id}} <br>";
        echo "Voter Name:  {{$vote->user->name}} <br>";
        if($vote->vote == 1){
            echo "Vote : 👍";
        }else{
            echo "Vote : 👎";
        }
        echo "<hr>";
        echo "<br><br>";
    }
});
```

```php
public function show($id)
{
    //
    $product = Product::find($id);
    $manufacturer = Product::find($id)->manufacturer;
    $reviews = Review::whereRaw('product_id = ?',array($id))->get();
    //dd($reviews);
    $votes = [];
    $results = [];
    //foreach($reviews as $review){
        //$votes = Vote::whereRaw('review_id = ?',array($review->id))->get();
        $votes = Vote::whereHas('review',function($query)use($id){
            return $query->whereRaw('product_id = ?',array($id));
        })->get();
        //$results = DB::select("select votes.id,votes.vote,votes.created_at,votes.updated_at,votes.user_id,votes.review_id,products.id a
        //dd($results);
    //}
    //dd($votes);
    return view('products.show')->with('product',$product)->with('reviews',$reviews)->with('votes',$votes)->with('results',$results);
    //return view('products.show')->with('product',product)->with('manufacturer',manufacturer);
}
```

Test first development using models and routes was used to develop the controller functions initially. Views were added to obtain instant feedback rather than reloying on SQL commands on the terminal to test CRUD commands.

Before any code was written sequential pseudo-code design was made to avoid the program from crashing due the foreign key violation constraints. Algorithms were designed such that when deleting a product first the reviews and the votes (likes and dislikes) would be deleted first. Likewise, when deleting reviews the code ensured that the votes with the foreign keys referencing to the reviews are deleted first. This relationship is illustrated in the Entity relationship diagram below.


Follow
PK id
FK user_id
FK following
User
PK id
name
email
password
type
Review
PK id
FK user_id
FK product_id
review
rating
Products
PK id
FK manufacturer_id
name
price
image
url
description
Manufaturer
PK id
name
Vote
PK id
FK reviewl_d
FK user_id
vote