CSci492 Senior Project Semester Paper

Grant Haataja
UND Computer Science
Grand Forks, ND, USA
grant.haataja@und.edu

Seth Thoelke
UND Computer Science
Grand Forks, ND, USA
seth.thoelke@und.edu

***Abstract*—This document details the progress made during Fall 2019 semester on the cinema project for Grand Forks startup company Project Rebel. The cinema project is the first iteration of Project Rebel's vision, geared toward small-scale individually-owned move theaters. The idea is to create a service for small business owners to easily construct their pre-movie advertisements to better generate revenue, without having to hire an expensive professional company to shoot and setup the advertisements for them.**

I. Introduction

Project Rebel was born as a result of a unique need, noticed amongst small-business movie theater owners in small towns. Such owners, who lack decent budgets to pay for advertisement development for themselves to generate revenue, became the target for the first iteration of a vast project to make a more user-friendly, clean, and powerful video editing web application.

The first iteration targets the movie theater owners, with the only functionality being to compile Digital Cinema Packages for their pre-movie advertisement runs. The key is that this service would allow them to easily add existing videos to a project that would run for them before their movie showings, without them having to try to build a way to deal with the raw files themselves, or pay outrageous amounts of money to professional advertisement-creating companies.

Eventually, Project Rebel plans to work toward becoming more versatile and including many more types of clients. And once the startup company is generating enough revenue to be sustainable, perhaps move toward developing an online video editor that is more user-friendly and effective than other similar services.

II. Development of the Project

A. Back-end Development

For development of back-end processes, PHP was used within the Laravel framework. Using Laravel allowed the team to focus on developing the product right away rather than focusing on setting everything up from scratch. [1] This required a lot of learning initially, the team watched a lot of tutorial videos to understand workings of Laravel, but eventually became comfortable enough to begin developing the functionality. Using Test-Driven Development, the team wrote specific tests for each new piece of functionality and worked to success from a point of failure. Utilizing TDD allowed for greater understanding of each part of the project added, as well as higher confidence in the results after the tests passed successfully. The following is an example of a test program for functionality of the Video object in our model:

```php
<?php

namespace Tests\Unit;

use App\User;
use App\Video;
use App\Library;
use Tests\TestCase;
use Illuminate\Database\Eloquent\Collection;
use Illuminate\Foundation\Testing\WithFaker;
use Illuminate\Foundation\Testing\RefreshDatabase;

class VideoTest extends TestCase
{
    use WithFaker, RefreshDatabase;

    public function testItHasAUser()
    {
        $video = factory(Video::class)->create();

        $this->assertInstanceOf(User::class, $video->user);
    }


    public function testItRetrievesItsLibraries()
    {
        $libraries = factory(Library::class, 5)->create();
        $video = factory(Video::class)->create();

        $libraries->first()->videos()->attach($video);
        $libraries->last()->videos()->attach($video);

        $this->assertInstanceOf(Collection::class, $video->libraries);
        $this->assertCount(2, $video->libraries);
        $this->assertInstanceOf(Library::class, $video->libraries->first());
        $this->assertEquals($libraries->first()->title, $video->libraries->first()->title);
    }
}
```

Listing 1. VideoTest.php

In test programs there are a number of assertions which are a way of simulating an action that a user of the platform might take, and testing that the results of that action are as expected.

For the first iteration of the project, the bare-bones platform geared for setting up Digital Cinema Packages for theater

owners who want to cut down on paying for their ad revenue, the back-end development was completed in November. The main objects involved are classes for Video, Library, and Project. A library can contain videos by themselves, as well as projects. A project can contain several videos, with the idea that a project would be one pre-movie advertisement run, with all the videos to be played included in that project. Here is the code for the Project model, which is quite similar in nature to the Video and Library models:

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;
use Illuminate\Support\Facades\DB;


class Project extends Model
{
    use SoftDeletes;

    protected $guarded = [];


    public function library()
    {
        return $this->hasOne(Library::class);
    }

    public function user()
    {
        return $this->belongsTo(User::class);
    }

    public function videos()
    {
        return $this->belongsToMany(Video::class);
    }

    public function add($video_id)
    {
        $max = DB::table('project_video')
                    ->where('project_id', $this->id)
                    ->max('order');

        ProjectVideo::create([
            'video_id' => $video_id,
            'project_id' => $this->id,
            'order' => $max ? $max + 1 : 1
        ]);
    }

    public function remove($video_id)
    {
        $this->videos()->detach($video_id);
    }
}
```

Listing 2. Project.php

B. Front-end Development

Upon moving from back-end to front-end development, another steep learning curve came into play. The team spent a lot of time learning JavaScript from Codecademy, and learning the Vue.js framework [2] through Laracasts. [3]

Learning Vue.js took longer for the team to get comfortable with than Laravel did. Vue.js is incredibly powerful, but contains so much additional functionality and new concepts that are unnatural for those accustomed to traditional coding methods.

III. Further Work

A. Upcoming Weeks

In the final weeks of the semester, Project Rebel's team intends to continue training in Vue.js in preparation for continued front-end development. Two of the team members not involved in this class plan to take a tour through the Midwest to visit potential customers they have been in contact with to showcase the product.

B. Next Semester

The primary goal for next semester is to finish the front-end user interface and launch the product. This will be primarily completed using the Vue.js framework, with additional functionality being added to the back-end as needed. The team will continue to hone in on potential additions to the product to iterate on the design and flesh out its functionality.

IV. Conclusion

Overall, the team learned many new concepts and development procedures over the Fall 2019 semester. The entire back-end framework for the platform was finished, with the team then moving to learning about and working toward finalizing the front-end functionality and user interface for the website.

References

[1] Laravel, "The PHP Framework for Web Artisans," [Online]. Available: https://laravel.com/. [Accessed: 05-Dec-2019].

[2] Vue.js, "The Progressive JavaScript Framework," [Online]. Available: https://vuejs.org/. [Accessed: 05-Dec-2019].

[3] Laracasts, "Screencasts for the Modern Developer," [Online]. Available: https://laracasts.com/. [Accessed: 05-Dec-2019].