CSci493 Senior Project Final 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 from September, 2019 through April, 2020 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, 2019. The main objects involved are classes for Video, Library, and Project. A library contains collections of videos for organization, while projects have a reel of specific 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 a snippet of 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 a great deal of additional functionality and introduces new concepts and ways of approaching problems that are unnatural for those accustomed to traditional programming methods.

Over the course of Spring 2020 semester, the team built up the front end functionality and user interface of the web app using Vue.js, with Tailwind CSS for styling. [4] Vue components were developed for every necessary page and element, including videos, the project page, library page, and dashboard page; user settings, app navigation areas, getting feedback from users, payments from users, uploading files, rendering projects, and much more. Below is shown the code for the user-profile component. Note that there are two `<router-link>` components embedded in this component, Account Settings and Downloads. Account Settings contains the functionality for users to change their name, email, and password, as well as upload a profile picture. Downloads is the page where the advertisement video reels will be posted after Project Rebel renders them, allowing the users to download the reel as a properly-formatted DCP.

```vue
<template>
  <div class="flex">
    <aside class="w-1/6">
      <section class="mb-12">
        <h6 class="mb-2">Profile</h6>
          <ul class="leading-loose">
            <li><router-link to="/app/profile/account-settings">Account Settings</router-link></li>
            <li><router-link to="/app/profile/downloads">Downloads</router-link></li>
          </ul>
      </section>
      <section class="mb-8 flex flex-col">
        <h6>Session</h6>
          <a href="/logout" @click.prevent="logout">Log out</a>
      </section>
    </aside>
  </div>
</template>

<script>
export default {
  methods: {
    logout() {
      axios.post('/logout').then(response => {
        window.location.replace('/');
      });
    },
  },
};
</script>
```

Listing 3. UserProfile.vue

The main draw of using Vue is its reactivity, so the team built the entire web app with that property in mind. This means that the app immediately reacts to everything the user does, eliminating the need for page refreshes or traditional HTML forms throughout most of the product.

To achieve this effect on a large scale, every single element is contained in a Vue component; the main App component is always loaded into the DOM, and each other component is a

child of the App and can be switched between via Vue routing. This allows only new and important information to be loaded by the front end and any given time, increasing the efficiency of the app. It also gives the user a better overall experience, seeing their actions directly affecting their projects in the app in real time; it is a similar idea to autosaving documents (which is indeed part of the functionality).

III. Further Work

A. Upcoming Weeks

In the final weeks of the semester, Project Rebel's team intends to continue fixing bugs and fleshing out the finer details of the web application. Although the application was launched to beta testers in early March, due to the unforeseen global conditions, the broader product launch has been indefinitely postponed as movie theaters across the country are shut down. Since that time, Project Rebel has been working on continuing to develop the web app, as well as reaching out to potential clients to stay in contact and provide support during the current crisis. The team has been researching and working on ways to serve theaters that have maintained contact while their buildings are closed, exploring options such as online film rentals, curbside popcorn sales, virtual community movie nights, and more.

IV. Conclusion

Overall, the team learned many new concepts and development procedures over the two semesters. The entire back-end framework for the platform was finished, the front-end was made fully-functional on a base level, and lots of work was done in the area of client relations. As soon as the movie theaters across the country begin to reopen, our web app will be ready for use.

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].

[4] Tailwind, "A utility-first CSS framework for rapidly building custom designs," [Online]. Available: https://tailwindcss.com/#what-is-tailwind. [Accessed: 29-Apr-2020].