RAD

AT2

Mitchel King

2019

Table of Contents

[1 Introduction 2](#_Toc10026884)

[1.1 Resources 2](#_Toc10026885)

[1.1.1 Human 2](#_Toc10026886)

[1.1.2 Physical 2](#_Toc10026887)

[2 Analysis 2](#_Toc10026888)

[3 Conclusion 2](#_Toc10026889)

# Sprint One

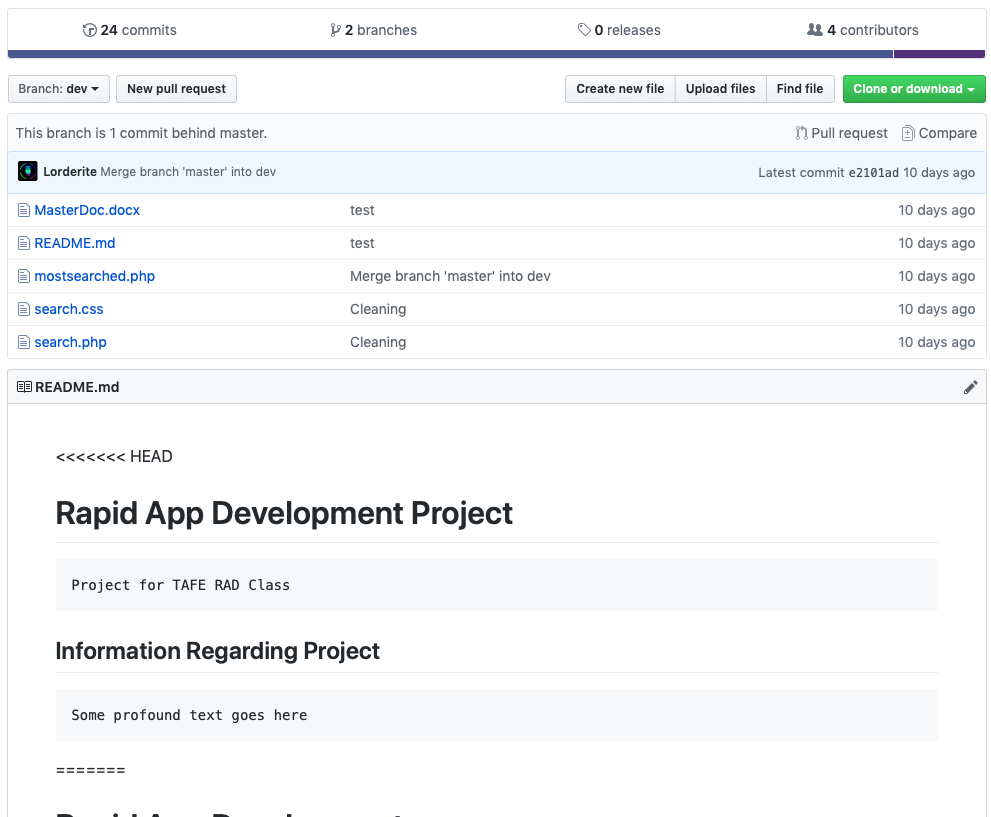
## Source Control

Source control for this project is GitHub, working with a local copy each, and merging with the central repo when changes are completed.

GitHub lets you see different versions of the source code and the differences between copies, issues and bugs can be created and tracked and assigned to team members.

Git Hub: <https://github.com/mkjking/RAD>

Snapshots;



## Project management plan

## Software Development Testing Plan

#### Scope

The scope of this testing plan is focused on the responsive design. The responsive design should be done before the end of the sprint so the testing plan for this sprint is to get the responsive design working.

#### In Scope

* Basic Functionality (searching, most searched)
* Responsive Design

#### Out of Scope

* Additional Features

#### Quality Objectives

This sprint requires that the responsive design fully works. Therefore, our Quality Objectives are:

* A fully functional responsive design implemented by the Second Sprint start.
* All basic functionality (searches etc.) is fully implemented and works as intended.

### Test Methodology

#### Overview

Due to the nature of responsive design, the testing will be done on multiple devices at various resolutions and browser sizes. The focus of the testing will be making sure there are no major issues with the responsive design, and that the page responds as intended.

#### Test levels

**Unit Test**

Each individual component should be tested separately from the whole project first, making sure that the code intention matches the result. This means that components such as image resizing, table resizing, font size changes and more should be tested individually first.

**Integration Test**

The tested components, i.e. image resizing, should then be integrated into the main system and tested to make sure the intended results do not change due to change in environment variables.

**System Test**

The entire system should be tested to see if the new integration has affected any other functionality, for example does the movie search still work, can you still search by rating etc.

**Acceptance Test**

After all components have been added and finalised, a final testing session should be done to make sure that the system is in acceptable state and works as intended.

#### Bug triage

*The bug triage shall use the following format:*

1. **Initial screenshot and description**

Post a screenshot of the issue occurring, and describe what the issue is, how it occurred and on what system it occurred.

1. **Confirmation of issue**

Recreate the issue and test in another environment/system to confirm the issue and whether it is system dependent.

1. **Issue address**

Address what fixes you have put in place/what changes you have made to fix the issue.

1. **Conclusion of issue**

Give a final statement on the condition of the bug and whether it was fixed or needs addressing later down the pipeline.

#### Test completeness

To finalise any level of testing, the goal requirements being solved must be achieved and/or furthered towards. This requires proper reading of the software requirements to make sure the product fits those requirements.

### Test deliverables

The first sprint has only two deliverables, to be shown at the end of the sprint to the relevant stakeholder:

* Basic functionality of the website (searching for the movies, displaying a table etc.)
* Working responsive design across the product.

These should be tested before showing the stakeholder, and then may be presented.

### Resource and environment needs

#### Testing tools

The web program is being hosted on a local server, notably XAMPP Apache server. Any development features of the browsers that the product is being tested on (Chrome, Firefox, Edge) to test the product.

#### Test environment

The product will be tested multiple popular browsers, namely Chrome and Firefox, on multiple systems.

|  |  |
| --- | --- |
| **Environment Variable Type** | **Environment Variable Used** |
| Operating System (64 Bit) | Windows 10, Mac OS |
| Web Browser | Chrome, Firefox, Edge |
| Resolutions | 1920x1080, 1366x768, 1280x720, 640x480 |

### Terms/Acronyms

## Analysis documentation

### CITE business rules for software development

CITE has a few standards and rules it follows for development

Coding standards;

* Naming conventions
* File naming and organization
* Formatting and indentation
* Comments and documentation
* Classes, Functions Interfaces
* Pointer and reference usage
* Testing

IP and Security;

* Intellectual property protection
  + Your software assets for your business have economic value, this value depends on the intellectual property rights involved. CITE uses industry best practices, all appropriate legal and physical guidelines to protect to protect your IP.
* Information Non-disclosure
  + A non-disclosure agreement is signed with all customers to provide peace of mind with all security and confidentiality issues.
  + All employees also sign a confidentiality agreement and must acknowledge understanding of CITE security policies.
* Distributed environment
  + All our online services run on CITE’s global network, not a local repository or server.
  + All data is distributed among this shared infrastructure with CITE server locations all around the world.
* Title transfer
  + We understand you want full ownership of the IP that we may develop.
  + CITE have a system in place that all employees and contractors must sign. This system enables CITE to own all IP as soon as it is created, waiving all moral rights to the individual.
* Data destruction
  + When data or services are retired from our system, we make sure the disks follow the data destruction procedure.
  + The procedure involves wiping the disks with the IT security team’s policies. If issues arise from wiping, then the disks are physically destroyed.
  + All data is destroyed if the disks are to leave CITE property.
* Physical security
  + Swipe cards
  + Role based access
  + CCTV
  + Fire protection
  + UPS systems, backup power

### CITE managed services quality assurance

Cite has a complex quality management system which follows a strict set of tasks and procedures to ensure quality is always maintained.

QMS;

* Procedures and regulations based on industry standards and best practices.
* Monitoring throughout lifecycle of the project to ensure consistent compliance with the regulations and practices.
* Product quality and verification against client needs and expectations.
* Create a collaboration environment for the project team to communicate efficiently
* Quality planning
  + CITE services makes custom plans that inform the client of standards, regulations, procedures guidelines and tools
* Quality assurance
  + CITE have processes in place to regularly check how the project is going to make sure it is meeting customer requirements.
* Quality control
  + Performance data is taken to decide whether the code is efficient or defective, to make sure quality is always high.
* Independent QA department
  + The QA department that perform these processes and tests are an independent branch of CITE, with its own dedicated QA engineers. Having a QA structure like this allows CITE to be extremely flexible with many ongoing projects.

### Acme entertainment development requirements

# Sprint two

# Conclusion