Nicolas Brown

lnl

at2 rad

Contents

[Sprint one 3](#_Toc43369590)

[Project Management Plan for Sprint one 4](#_Toc43369591)

[Source Control snapshot 4](#_Toc43369592)

[Trello 4](#_Toc43369593)

[Github 4](#_Toc43369594)

[Analysis Report 5](#_Toc43369595)

[Mission of CITE 5](#_Toc43369596)

[CITE business rules for software development 5](#_Toc43369597)

[CITE Managed Services Quality Assurance 5](#_Toc43369598)

[Acme Entertainment Pty Ltd development requirements 6](#_Toc43369599)

[Software Development Testing Plan 7](#_Toc43369600)

[Introduction 7](#_Toc43369601)

[Functional Requirements 7](#_Toc43369602)

[Non-Functional Requirements 7](#_Toc43369603)

[Quality Objective 7](#_Toc43369604)

[Member Roles 7](#_Toc43369605)

[Test Methodology 8](#_Toc43369606)

[Overview 8](#_Toc43369607)

[Test Levels 8](#_Toc43369608)

[Suspension Criteria and Resumption Requirements 8](#_Toc43369609)

[Test Completeness 8](#_Toc43369610)

[Test Deliverables 8](#_Toc43369611)

[Resource and Environment Needs 9](#_Toc43369612)

[Testing Tools 9](#_Toc43369613)

[Testing Environment 9](#_Toc43369614)

[Test Cases 9](#_Toc43369615)

[Multi-Platform Report 13](#_Toc43369616)

[Adaptive Web Design 13](#_Toc43369617)

[Responsive Web Design 14](#_Toc43369618)

[Sprint Two 15](#_Toc43369619)

[Project Management Plan for Sprint one 16](#_Toc43369620)

[Source Control snapshot 16](#_Toc43369621)

[Trello 16](#_Toc43369622)

[Github 16](#_Toc43369623)

[Software Review Plan 17](#_Toc43369624)

[Goal of the Software Review plan 17](#_Toc43369625)

[Process of Software review 17](#_Toc43369626)

[Software Peer Review 18](#_Toc43369627)

[Benefits of the Software Review 18](#_Toc43369628)

[Software Performance testing 19](#_Toc43369629)

[Testing types 19](#_Toc43369630)

[Various testing software 20](#_Toc43369631)

[WAPT 20](#_Toc43369632)

[AppOptics 20](#_Toc43369633)

[Testing For Sprint Two 21](#_Toc43369634)

[Testing Tools 21](#_Toc43369635)

[Testing Environment 21](#_Toc43369636)

[Test Cases 21](#_Toc43369637)

[Sprint Three 23](#_Toc43369638)

[Project Management Plan for Sprint one 24](#_Toc43369639)

[Source Control snapshot 24](#_Toc43369640)

[Trello 24](#_Toc43369641)

[Github 24](#_Toc43369642)

[Optimization Report 25](#_Toc43369643)

[PEAR test 25](#_Toc43369644)

[Commenting 30](#_Toc43369645)

# Sprint one

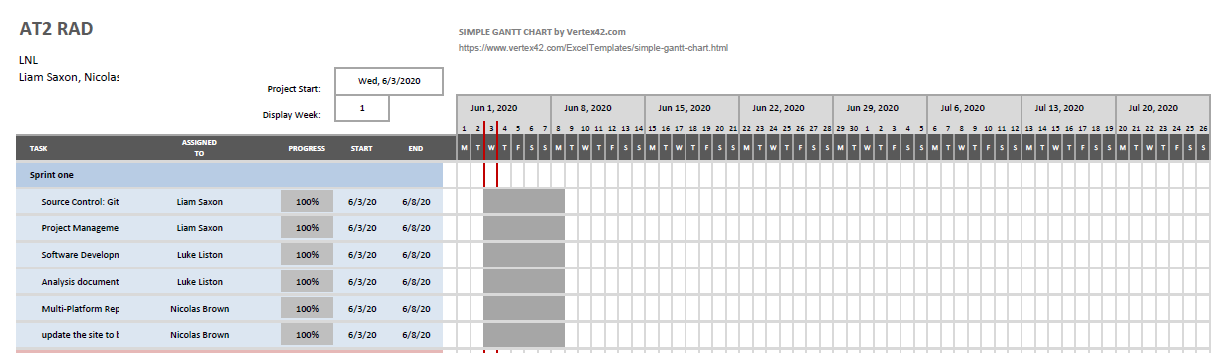
Team Name: **LNL**

Scrum Master (1st team leader): **Liam Saxon**

Team Member (#2): **Nicholas Brown**

Team Member (#3): **Luke Liston**

# Project Management Plan for Sprint one



# Source Control snapshot

## Trello

<https://trello.com/b/16EiOtwb/rad>

## Github

<https://github.com/Liam-Saxon/RAD>

# Analysis Report

This analysis report is being written to analyse requirements of both CITE and Acme Entertainment Pty Ltd, while undergoing this project to develop a movie database for CITE.

The following will be included

* CITE business rules for software development,
* CITE Managed Services Quality Assurance,
* Acme Entertainment Pty Ltd development requirements

## Mission of CITE

Cite operates on a group of core principals which include the following.

* Clients
* Accountability
* People and Reputation
* Professionalism and quality
* Innovation, creativity
* Teamwork
* Size
* Anticipation
* Growth
* Integrity

These core principals make up their mission they seek to achieve in their line of work.

## CITE business rules for software development

Documented on CITE’s website are their coding standards that each of their developers must follow to ensure that code is written in a preferred style and not a myriad of different styles that will affect later readability and usability.

Common aspects of the CITE coding standards:

* Naming Conventions
* File Naming and Organization
* Formatting and Indentation
* Comments and Documentation
* Classes, Functions and Interfaces
* Pointer and Reference Usage
* Testing

Coding standards will reflect the context of the language and their client’s requirements.

In the absence of the following criteria CITE developers will default to industry standards.

All the systems and projects are covered by the ISO Standard:

ISO/IEC/IEEE 12207:2017 Systems and software engineering — Software life cycle processes

(Services C. M., n.d.)

## CITE Managed Services Quality Assurance

CITE managed services has implemented a Quality Management System (QMS) comprising of a complex set of engineering and managerial activities

The QMS has a set of tasks and objectives in place to hopefully maximise quality assurance.

* Elaboration and implementation of procedures and regulations for software development process based on industry standards and best practices;
* Product lifecycle monitoring to ensure compliance with established processes and guidelines
* Product quality verification and validation to ensure that it complies with clients’ business needs and expectations;
* Establishment of an effective collaboration between all project team members

With a comprehensive approach to quality through more refined steps such as:

**Quality planning**

CITE Managed Services puts together quality plans that govern the applicable set of standards, regulations, procedures, guidelines and tools during the development lifecycle in each project.

**Quality Assurance**

We have established processes that evaluate project performance and aim to assure that quality standards are being followed and that the deliverables comply with customer requirements.

**Quality Control**

We measure performance trends to identify defective pieces of code, verify that deliverables are of high quality and that they are complete and correct.

(Services, n.d.)

## Acme Entertainment Pty Ltd development requirements

Acme requirements as per the AT2 Project sprint One outline the following requirements.

Acme requires a Movie database prototype that they wish to review and update so it may work on all major digital platforms. They require a Multi-Platform Report comparing the two design options currently used; adaptive and responsive. The team must choose an existing movie database and modify it to meet these requirements to later be presented to the client at the end of the first sprint.

The site can either be hosted locally or on the cloud, ensuring that our lecturer is aware of which one. Conducting and recording suitable testing of the completed spring development in the testing plan.

# Software Development Testing Plan

## Introduction

This project is designed to be worked on in a team of three team members. The aim is to use the Agile workflow method to deliver a quality product within scope and on time following the standards of quality defined by CITE.

## Functional Requirements

* The list of movies must be displayed as a list.
* Searches will filter out movies that do not match the criteria of the search.
* Specific titles, ratings, years and genres must be able to be searched for by the user.
* A list of the top searched movies must be displayed in a graph in a separate page.

## Non-Functional Requirements

* Search results may be segmented into multiple pages when there are many results for the sake of performance on low end hardware that can only show few results at a time.
* The top searched movies graph may be displayed as a coloured bar graph.
* The graph may also display a legend to indicate the titles more clearly.
* All pages may load within 10 seconds of the last user input, making the website responsive and quick to use.
* Pages must be properly rendered on larger and smaller screens without parts of the website being off screen or of an inappropriate size.

## Quality Objective

An objective of quality testing is to assure the features of the final product are in alignment with the functional and non-functional requirements. High priority features concerning the correct functionality and usability of the website will be addressed.

Testing of the project is conducted throughout the development of the project. Errors and other outstanding issues will be fixed before deployment of the program as per the Agile SDLC the team will be following.

The project will meet the standards of quality specified by CITE at this page: <http://www.citems.com.au/?page_id=93>.

## Member Roles

* The boss is the overarching manager of all projects currently being worked on. The only member of the development teams that communicate with the boss are the leaders of those groups.
* The role of scrum master will be moved to a new team member every week. They are the main communication between the boss and the development team and manage how the rest of the team operates based on what the boss says.
* The development team, including the scrum master and the rest of the team members, must work based on the input of the scrum master and communicate between each other to accomplish the work for the current sprint.

# Test Methodology

## Overview

The project will be using the Agile testing methodology. It is the standard for most professional projects.

## Test Levels

The following types of test will be conducted:

* Unit testing, when the separate parts of the project are tested individually.
* Integration Testing, that is performed on individual units that are combined to be tested as a group.
* System Testing, which is done on a complete integrated system to ensure correspondence with the laid-out requirements.
* Acceptance Testing, when the implementation of the project is complete, it will then be evaluated whether it is suitable for deployment.

## Suspension Criteria and Resumption Requirements

In the testing of the project, team members find that %40 of the tests result in failure, testing will be suspended until development has fixed all failed cases.

## Test Completeness

The pass rate of tests will be %80. Tests are not considered to have passed if they fall below that threshold. All decided manual or automated tests must be completed.

## Test Deliverables

The following test deliverables will be produced:

Before the testing phase:

* Testing plan document
* Test cases document
* Test design specifications

During the testing:

* Test data
* Testing logs and error logs

After the testing:

* Results of the testing
* Defect report
* Installation and test procedures guide
* Release note

# Resource and Environment Needs

## Testing Tools

* Automated testing tools that can conduct tests and auto generate test data.
* Debugging tools used by the development to manually find bugs and errors in the code.

## Testing Environment

* A modern Windows 10 computer.
* An internet network connection.
* A server to host the website and database. A solution such as USBWebServer may also be satisfactory.
* The Google Chrome web browser to navigate the website.

## Test Cases

These are the test cases for the current version of the project.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Description | Expected Outcome | Evidence |
| Case 1 | Open the website/Click the Home button in the header | Opens the home page (index) |  |
| Case 2 | Click the Search button in the header | Opens the search page, there will be no search results at first |  |
| Case 3 | Click the Top Searches button in the header | Opens the top 10 searches page, the graph will be empty at first |  |
| Case 4 | Search for a movie title only | Displays movies containing the term in its title | Searched for “monkey” in title textbox |

|  |  |  |  |
| --- | --- | --- | --- |
| Case 5 | Search for a movie genre only | Displays movies containing the term in its genre | Searched for “horror” in genre textbox |
| Case 6 | Search for a movie rating only | Displays movies containing the term in its rating | Searched for “PG” in rating textbox |
| Case 7 | Search for a movie year only | Displays movies containing the term in its year | Searched for “1999” in year textbox |
| Case 8 | Search using multiple textboxes | Displays movies containing the term in the specified columns | Searched for “12 Monkeys, R, 1995, Drama” |
| Case 9 | Search with no matching results | Displays a “No results” message | Searched for “Movie, G, 2000, Horror” |
| Case 10 | View top searches page after making several searches | Displays a graph showing the top 10 searched movies |  |
| Case 11 | Different browsers | Tested on chrome, firefox and Edge | Chrome  Firefox    Edge |
| Case  12 | Different sizes for scaling purposes | 100% and 150% | 100%    150% |

# Multi-Platform Report

## Adaptive Web Design

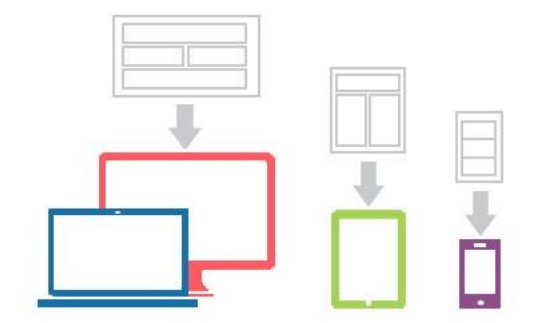
Adaptive Web Design was first introduced by web designer Aaron Gustafson in his book, Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement which released in 2011. It is also known as progressive enhancement of a website.

Adaptive Web Design involves creating fixed layouts. When the site detects the available space, it selects the layout most appropriate for the screen. So, when you open a browser on the desktop, the site chooses the best layout for that desktop screen; resizing the browser has no impact on the design.

Due to Adaptive Web Design having fixed layouts it has many benefits these include allowing designers to build the best user experience for the appropriate device, mobile devices can sense their user’s environment, and designers can optimize advertisements based on user data from smart devices. Although there are many benefits to using Adaptive Web Design there are a lot more cons to using it, these include labour-intensive to create – most adaptive designs are retrofitting traditional sites to make them more accessible, tablets and netbooks can have trouble with site configuration tending to be smartphone or desktop oriented, challenging to SEO — Search engines have trouble appreciating identical content on multiple sites.

As a result of the negatives of Adaptive Web Design it is less used that Responsive web Design for modern web pages

An example of Adaptive Web Design is



## Responsive Web Design

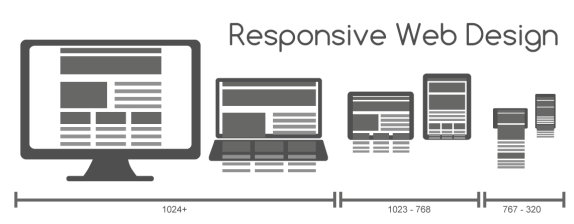
Responsive Design was first coined by the web designer and developer Ethan Marcotte in his book, Responsive Web Design. Responsive designs respond to changes in browser width by adjusting the placement of design elements to fit in the available space.

A responsive website shows content based on the available browser space. If you open a responsive site on the desktop and then change the size of the browser window, the content will move dynamically to arrange itself optimally for the browser window. On mobile phones, this process is automatic; the site checks for the available space and then presents itself in the ideal arrangement.

Due to Responsive Web Design having fluid layouts it has many benefits these include it being uniform and seamless leading to good user experience, abundance of templates to use, SEO friendly, often easier to implement Although there are many benefits to using Responsive Web Design there are also a lot of cons to using it, these include less screen size design control, elements can move around, advertisements may be lost on screen, longer mobile download times.

As a result of the fewer negatives of Responsive Web Design it is more used that Adaptive web Design for modern web pages

An example of Responsive Web Design is



# Sprint Two

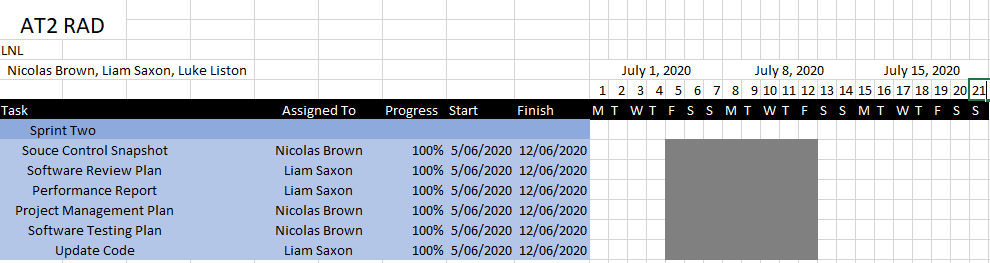
Team Name: **LNL**

Scrum Master (1st team leader): **Nicholas Brown**

Team Member (#2): **Liam Saxon**

Team Member (#3): **Luke Liston**

# Project Management Plan for Sprint one



# Source Control snapshot

## Trello

<https://trello.com/b/16EiOtwb/rad>

## Github

<https://github.com/Liam-Saxon/RAD>

# Software Review Plan

A software review plan is a systematic inspection of code by one or more individuals who work together to correct errors and defects in the software during its early stages of (SDLC) Software development life cycle. Software review is integral to ensuring that the client’s needs are met and that everything is operating how it should be.

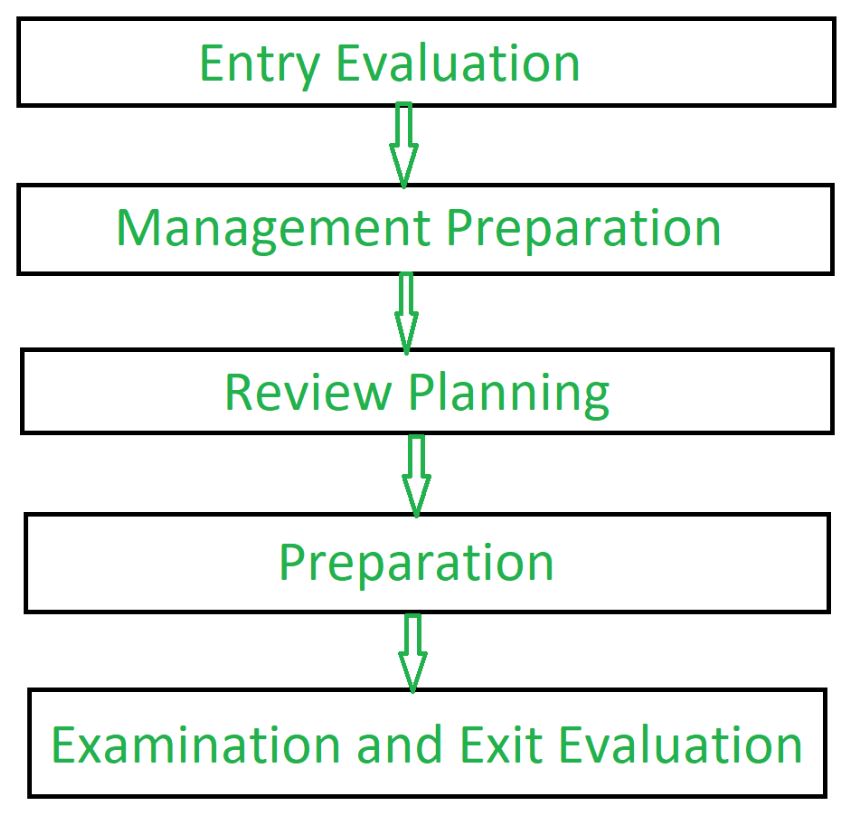
Usually performed manually, software review is used to verify various documents like requirements, system designs, codes, test plans and test cases.

(geeksforgeeks, n.d.)

## Goal of the Software Review plan

* To improve the productivity of the development team.
* To make the testing process time and cost effective.
* To make the final software with fewer defects.
* To eliminate the inadequacies.

## Process of Software review



## **Software Peer Review**

The following form of Software review is what we will be closely following to ensure code is reviewed and inspected for functionality.

Software Peer Review:

Peer review is the process of assessing the technical content and quality of the product and it is usually conducted by the author of the work product along with some other developers.

Peer review is performed in order to examine or resolve the defects in the software, whose quality is also checked by other members of the team.

Peer Review has following types:

1. **Code Review**:

Computer source code is examined in a systematic way.

1. **Pair Programming**:

It is a code review where two developers develop code together at the same platform.

1. **Walkthrough:**

Members of the development team is guided by author and other interested parties and the participants ask questions and make comments about defects.

1. **Technical Review:**

A team of highly qualified individuals examines the software product for its client’s use and identifies technical defects from specifications and standards.

1. **Inspection:**

In inspection the reviewers follow a well-defined process to find defects

## Benefits of the Software Review

* Defects and issues can be found early and corrected
* Earlier detection minimizes the cost both in time and money
* It can be used improve upon technical documentation
* Remove inferior ways of documenting and developing code.

# Software Performance testing

Software performance testing is a general practise performed to gauge how a system performs in terms of stability and responsiveness under a workload. It can also be used to measure, validate or verify attributes of the system, such as scalability, reliability and resource usage.

## Testing types

Below we will briefly go over the types of testing.

* Load testing

Load testing is the simplest test, where the system is put under a certain amount of load e.g., a set a number of online users.

* Stress testing

A stress test is a test used to push the software to its maximum limits and see if it can still operate at those limits

* Soak testing

A soak test is to see if the system can sustain a continuous expected load, memory utilization is monitored to check for memory leaks.

* Spike testing

Spike testing is done by suddenly increasing or decreasing the load generated by a very large number of users, and observing the behaviour of the system

* Breakpoint testing

Breakpoint testing is like stress testing. An incremental load is applied over time while the system is monitored for predetermined failure conditions. Breakpoint testing is sometimes referred to as Capacity Testing because it can be said to determine the maximum capacity below which the system will perform to its required specifications or Service Level Agreements

* Configuration testing

Rather than testing for performance from a load perspective, tests are created to determine the effects of configuration changes to the system's components on the system's performance and behaviour. A common example would be experimenting with different methods of load-balancing.

* Isolation testing

Isolation testing is not unique to performance testing but involves repeating a test execution that resulted in a system problem.

* Internet testing

This is a relatively new form of performance testing when global applications such as Facebook, Google and Wikipedia, are performance tested from load generators that are placed on the actual target continent whether physical machines or cloud VMs. These tests usually require an immense amount of preparation and monitoring to be executed successfully.

## Various testing software

Some of the below are various Software performance testing software.

## WAPT

WAPT is a load testing tool for websites

* Create tests using a browser or a mobile application
* Remote, distributed and cloud-based testing
* Server and database performance monitoring
* Flexible error handling and adjustable pass/fail criteria
* Charts with instant test results available at run time
* Customizable test report with over 20 tables
* Support of all major web development frameworks

## AppOptics

AppOptics APM is a performance monitor for many different frameworks and libraries as mentioned below.

* Out-of-the-box support for dozens of frameworks and libraries including Go, Java, .NET, PHP, Ruby, Python, Scala, and Node.js
* Distributed tracing, live code profiling, and exception tracking follow requests across processes, hosts, and data centre, providing detailed transaction performance visualization by service and further drill-down to the code—no manual instrumentation required

# Testing For Sprint Two

## Testing Tools

* Automated testing tools that can conduct tests and auto generate test data.
* Debugging tools used by the development to manually find bugs and errors in the code.

## Testing Environment

* A modern Windows 10 computer.
* An internet network connection.
* A server to host the website and database. A solution such as USBWebServer may also be satisfactory.
* The Google Chrome web browser to navigate the website.

## Test Cases

These are the test cases for the current version of the project.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Description | Expected Outcome | Evidence |
| Case 1 | Singing up using only a first name | Error given |  |
| Case 2 | Singing up using a full name | Email and name added to database |  |
| Case 4 | Singing up using an already existing account | Error given |  |
| Case 3 | Singing up using a number in the name | Error given |  |
| Case 4 | Singing up using an incorrect email | Error given |  |
| Case 5 | Singing up using a correct email | Email and name added to database |  |
| Case 6 | Singing up and not selecting either Send an email or Show notifications | Error given |  |
| Case 7 | Singing up and selecting only Send an email | Email and name added to database only for sending emails |  |
| Case 8 | Singing up and selecting only Show notifications | Email and name added to database only for notifications |  |
| Case 9 | Choosing to unsubscribe | Email sent to admin |  |
| Case 10 | Adding an admin | Adds a new admin to the database |  |
| Case 11 | Signing in as an admin | Allows access to admin controls |  |
| Case 12 | Unsubscribing a user | Changes the user’s status to not receive emails |  |
| Case 13 | Removing a user | Removes them from the database |  |

# Sprint Three

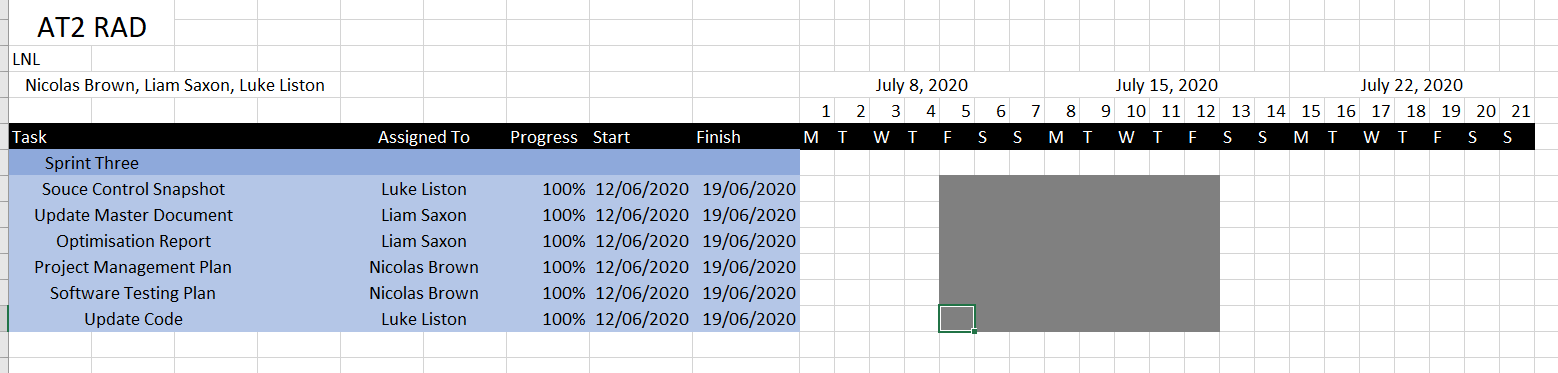
Team Name: **LNL**

Scrum Master (1st team leader): **Luke Liston**

Team Member (#2): **Liam Saxon**

Team Member (#3): **Nicholas Brown**

# Project Management Plan for Sprint one



# Source Control snapshot

## Trello

<https://trello.com/b/16EiOtwb/rad>

## Github

<https://github.com/Liam-Saxon/RAD-LNL>

Optimization Report

This optimization report is being done to show that the code meets requirements, in a few areas.

* Commenting
* Readability
* PEAR Validation on PHP code

In having the above three done correctly it will mean that the code is optimized enough for readability and future modifications for any team or individuals who may come along and have to modify it.

PEAR test

adminControls.php



connect.php



graph\_rated\_scr.php



graph\_scr.php



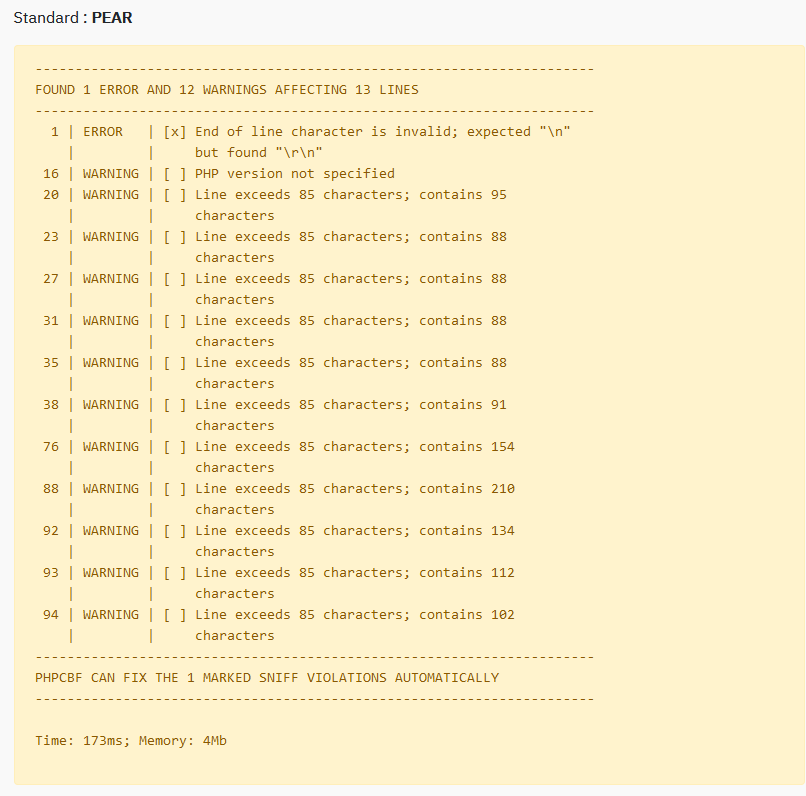
login.php



loginAdmin.php



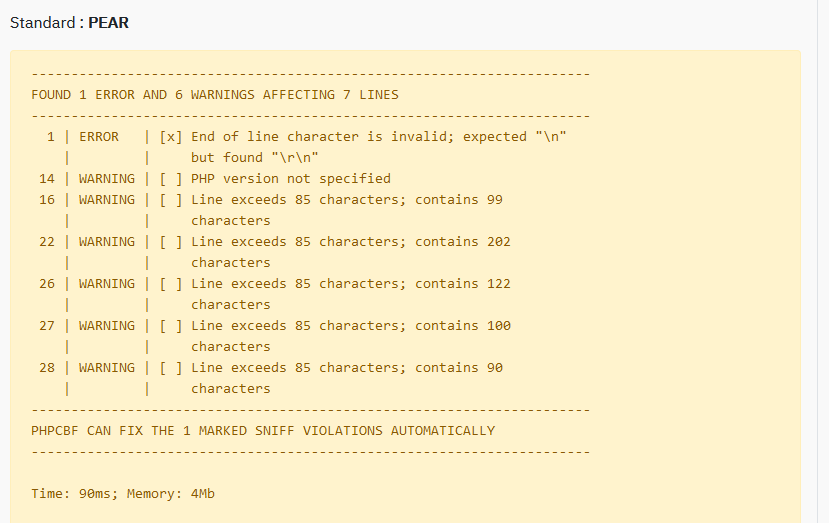
moviesSearch\_scr.php



ratedGraph.php



ratingHistroy.php



searchGraph.php

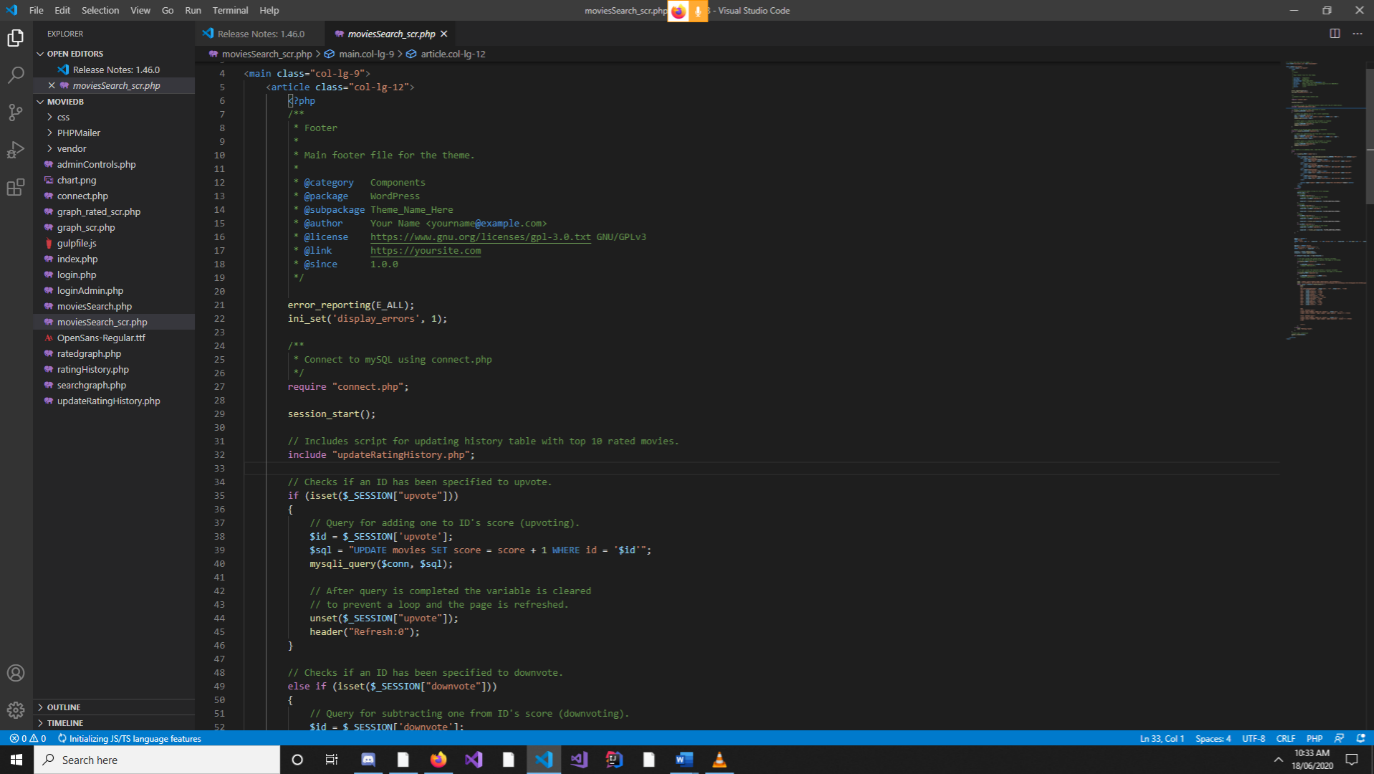


updateRatingHistory.php



Commenting

Through assessment of the code, most if not all methods or areas of complex code are commented as to allow future individuals who need to assess the code to be able to do so without complications.



Above is an example of one of the many pages with commenting. Due to the sheer complexity and size of the program the above image will just be used as an example.