AT2

Movie Database Application

Software Development Master Document

Team Name:

Team Bare Maximum

Members:

Reece Pieri, Jose Rico Imbang, Say Hon Lee

Diploma of Software Development - RAD:

Diploma of Software Development

Rapid Application Development

Contents

[1 Introduction 5](#_Toc57221809)

[1.1 Purpose of the Software Development Master Document 5](#_Toc57221810)

[2 Team 5](#_Toc57221811)

[2.1 Team Information 5](#_Toc57221812)

[3 Sprint One 6](#_Toc57221813)

[3.1 Team 6](#_Toc57221814)

[3.2 Source Control Snapshot 6](#_Toc57221815)

[3.3 Project Management Plan 7](#_Toc57221816)

[3.4 Software Development Testing Plan 8](#_Toc57221817)

[1 Introduction 8](#_Toc57221818)

[1.1 Purpose of the Software Development Testing Plan 8](#_Toc57221819)

[1.2 Scope 8](#_Toc57221820)

[1.3 Quality Objective 9](#_Toc57221821)

[1.4 Roles and Responsibilities 9](#_Toc57221822)

[2 Test Methodology 10](#_Toc57221823)

[2.1 Overview 10](#_Toc57221824)

[2.2 Test Levels 10](#_Toc57221825)

[2.3 Bug Triage 11](#_Toc57221826)

[2.4 Test Completeness 11](#_Toc57221827)

[3 Test Deliverables 12](#_Toc57221828)

[4 Resources & Environment 12](#_Toc57221829)

[4.1 Testing Tools 12](#_Toc57221830)

[4.2 Test Environment 12](#_Toc57221831)

[3.5 Analysis Report 13](#_Toc57221832)

[CITE Business Rules for Software Development 13](#_Toc57221833)

[CITE Managed Services Quality Assurance 14](#_Toc57221834)

[Acme Entertainment Pty Ltd Development Requirements 14](#_Toc57221835)

[3.6 Multi-platform Report 15](#_Toc57221836)

[1 Introduction 15](#_Toc57221837)

[1.1 Purpose of the Multi-platform Report 15](#_Toc57221838)

[2 Adaptive Design 16](#_Toc57221839)

[2.1 Definition of Adaptive Design 16](#_Toc57221840)

[2.2 Advantages and disadvantages of Adaptive Design 16](#_Toc57221841)

[3 Responsive Design 17](#_Toc57221842)

[3.1 Definition of Responsive Design 17](#_Toc57221843)

[3.2 Advantages and disadvantages of Responsive Design 17](#_Toc57221844)

[4 Conclusion 18](#_Toc57221845)

[3.7 Test Document 19](#_Toc57221846)

[Test Data 19](#_Toc57221847)

[4 Sprint Two 26](#_Toc57221848)

[4.1 Team 26](#_Toc57221849)

[4.2 Source Code Snapshot 26](#_Toc57221850)

[4.3 Project Management Plan 26](#_Toc57221851)

[4.4 Software Review Plan 27](#_Toc57221852)

[1 Introduction 27](#_Toc57221853)

[2 Type Of Software Review 28](#_Toc57221854)

[2.1 Software peer review 28](#_Toc57221855)

[2.2 Software Audit Review 28](#_Toc57221856)

[3 Formal Review VS Informal Review 29](#_Toc57221857)

[3.1 Formal review 29](#_Toc57221858)

[3.2 Informal review 29](#_Toc57221859)

[4 Review User 30](#_Toc57221860)

[5 Software Review Plan to be implemented 30](#_Toc57221861)

[6 Software Review Checklist 31](#_Toc57221862)

[4.5 Performance Report 32](#_Toc57221863)

[1 Introduction 32](#_Toc57221864)

[2 Code Optimization 32](#_Toc57221865)

[2.1 About code optimization 32](#_Toc57221866)

[2.2 Advantages of code optimization 32](#_Toc57221867)

[2.3 Code optimization tool 32](#_Toc57221868)

[3 Software Requirement 33](#_Toc57221869)

[4 Performance Testing 34](#_Toc57221870)

[4.1 What is performance testing 34](#_Toc57221871)

[4.2 Type of performance testing 34](#_Toc57221872)

[4.3 Common performance issue 35](#_Toc57221873)

[4.4 Performance Testing Process 36](#_Toc57221874)

[4.5 Performance Testing Metrics 37](#_Toc57221875)

[4.6 Test table 38](#_Toc57221876)

[5 Sprint Three 44](#_Toc57221877)

[5.1 Team 44](#_Toc57221878)

[5.2 Source Code Snapshot 44](#_Toc57221879)

[5.3 Project Management Plan 44](#_Toc57221880)

[5.4 Optimisation Report 45](#_Toc57221881)

[1 Introduction 45](#_Toc57221882)

[2 Code Optimisation Done for HTML 45](#_Toc57221883)

[2.1 Index 45](#_Toc57221884)

[2.2 Admin 45](#_Toc57221885)

[3 Code Optimisation Done for PHP 46](#_Toc57221886)

[3.1 Admin 46](#_Toc57221887)

[3.2 Connectors 50](#_Toc57221888)

[3.3 Index 51](#_Toc57221889)

[3.4 Mail 57](#_Toc57221890)

[5.5 Software Testing Plan 59](#_Toc57221891)

[1 Introduction 59](#_Toc57221892)

[1.1 Purpose of the Software Development Testing Plan 59](#_Toc57221893)

[1.2 Scope 59](#_Toc57221894)

[1.3 Quality Objective 60](#_Toc57221895)

[1.4 Roles and Responsibilities 60](#_Toc57221896)

[2 Test Methodology 61](#_Toc57221897)

[2.1 Overview 61](#_Toc57221898)

[2.2 Test Levels 61](#_Toc57221899)

[3 Test Deliverables 62](#_Toc57221900)

[3.1 Test Table 62](#_Toc57221901)

[4 Resources & Environment 67](#_Toc57221902)

[4.1 Testing Tools 67](#_Toc57221903)

[4.2 Test Environment 67](#_Toc57221904)

# 1 Introduction

## 1.1 Purpose of the Software Development Master Document

The purpose of the Software Development Master Document is to serve as a central source of information pertaining to all aspects of the development of the Movie Database Application.

# 2 Team

## 2.1 Team Information

Team Name: Team Bare Maximum

Members: Reece Pieri (M087496)

Jose Rico Imbang (30019932)

Say Hon Lee (30003628)

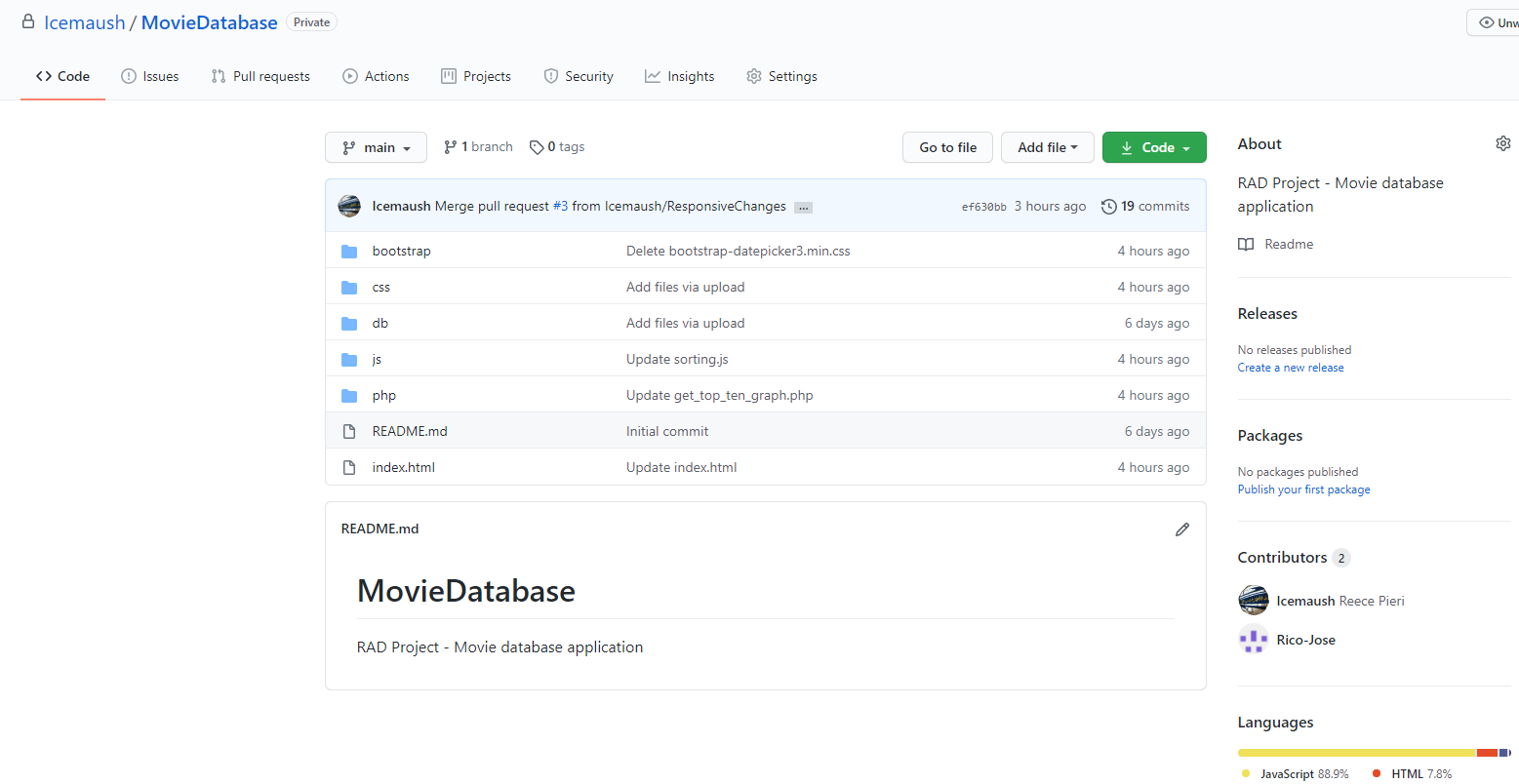
# 3 Sprint One

## 3.1 Team

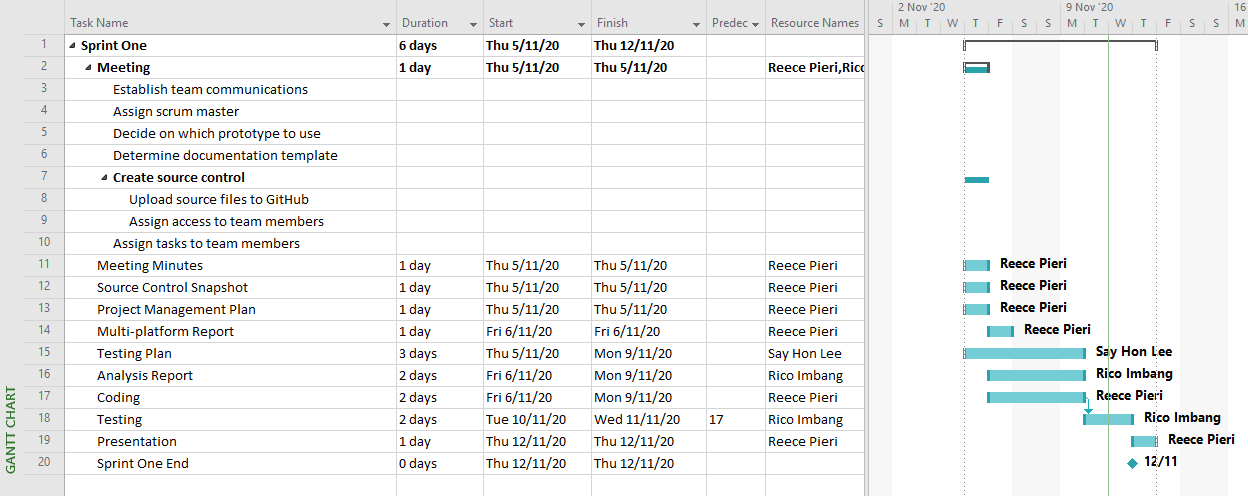
Scrum Master: Reece Pieri

Members: Jose Rico Imbang, Say Hon Lee

## 3.2 Source Control Snapshot



## 3.3 Project Management Plan



## 3.4 Software Development Testing Plan

1 Introduction

1.1 Purpose of the Software Development Testing Plan

The purpose of the Software Development Testing Plan is to outline the scope and strategies that will be applied to the testing of the Movie Database application.

1.2 Scope

The functional requirements of the Movie Database application include:

* Displaying all available movies
* Connection to database
* Searching of specific movies
* Searching movies by name, genre, released year or ratings
* Displaying the top 10 movies searched by user in graph
* Compatible with multiple device such as computers, tablet and mobile devices.

The non-functional requirements of the Movie Database application include:

* Responsive time of the website
* Maintainability of the website
* Security of the database
* Software flexibility (moving from one operating system to other operating system shouldn’t crash the website)

1.3 Quality Objective

* Ensure the application functional and non-functional requirements meets the client expectations.
* Ensure the AUT meets the quality specifications requested by the client.
* Bugs and issues are identified and fixed before releasing the product into the market.

1.4 Roles and Responsibilities

All team members will be contributing to all aspects of the development of the application. Tasks and responsibilities will rotate weekly with each sprint.

2 Test Methodology

2.1 Overview

Due to the client’s requirement not being clearly outlined at the beginning of the project, the testing methodology best suited for this project is the Agile methodology due to its flexibility and the need to present a working prototype to the client frequently. This allows the improvement of the product quality in a progressive manner.

2.2 Test Levels

Four stages of testing will be applied to the Movie Database application. These include:

* Unit testing – it is a smallest testable portion of the systems ensuring it could be compiled, loaded and executed.
* Integration testing – as the website is connected to a database we will also need to ensure that data would be displayed onto the webpage while customers visit the page.
* System testing – it refers to checking the overall interaction of the components built in the webage to check on the loading, performance, reliability and security of the webpage.
* Acceptance testing – is to test the conduct to find if the requirements of a specification or contract are met as per its delivery. I will be tested by the clients to ensure the level of satisfactory have been met.

2.3 Bug Triage

The purpose of bug triage is to define the resolution type of each bug detected during testing and prioritize fixes based on bug severity and determine a schedule to apply bug fixes.

2.4 Test Completeness

Once testing is complete and bug fixes have been applied, the acceptance test will be completed to ensure no new bugs are introduced and all functionality meets the client’s requirements.

3 Test Deliverables

At the end of each testing phase a Testing Document will be produced. This document will contain a test table which will describe the tests that have been performed as well as screenshots of the test results.

4 Resources & Environment

4.1 Testing Tools

The IDE debugger will be used to detect and assist in resolving existing bugs and errors in the application and also to ensure that the code can be compiled, making sure that all the functions and requirements are working as per the client requirements.

4.2 Test Environment

The following software is required in addition to client-specific software.

* Devices: PC, tablet, smart phone
* Browsers: Google Chrome, Mozilla Firefox, Safari, Microsoft Edge
* Webserver with PHP and MySQL
* Internet connection

## 3.5 Analysis Report

CITE Business Rules for Software Development

CITE Managed Services (CITE MS) develops software for a range of clients. CITE MS has employees but they utilise external contractors when necessary and all of them must adhere to the company policy.

CITE MS uses Software Development Life Cycle (SDLC) as a process to design, develop, test, and maintain high quality applications. To ensure that any software that will be developed will be adequately documented and tested before it is used for sensitive client information, the SDLC should at least address the following areas:

* Preliminary analysis or feasibility study
* Risk identification and mitigation
* Systems analysis
* Design specification
* Development
* Quality assurance
* Implementation
* Post-implementation maintenance and review

Coding standards must be followed when developing applications for or at CITE MS. To ensure security is rigorously maintained for the production system while the development and test environments can maximise productivity with fewer security restrictions, there must be a separation between the production system, development and test environments. Also, development and test staff must not be permitted to have access to production systems.

CITE Managed Services Quality Assurance

CITE MS has established processes that evaluate project performance, assure that quality standards are being followed, and that the deliverables comply with customer requirements. A Quality Management System (QMS) has been implemented which comprises a complex set of engineering and managerial activities that ensure quality of delivered software throughout the entire workflow. QMS Tasks and Objectives are the following:

* 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.

Acme Entertainment Pty Ltd Development Requirements

Acme Entertainment have commissioned the movie database project and CITE MS have presented a prototype. However, its user interface is designed primarily for desktop computers and Acme requires it to function correctly on major platforms (desktop computer, tablet, and cell phone). To accomplish this task, the interface must utilise either adaptive or responsive design. The project can be hosted on the cloud or a suitable local server.

## Multi-platform Report

1 Introduction

1.1 Purpose of the Multi-platform Report

The purpose of the Multi-platform Report is to outline the differences between the two design options, Adaptive and Responsive Design, and to come to a conclusion on which design is more suitable for the Movie Database Application.

This document aims to clearly describe each design option and provide advantages and disadvantages of each, as well as describe the reason for our selected design approach.

2 Adaptive Design

2.1 Definition of Adaptive Design

Adaptive Design is the design of a website in which there are several fixed layout sizes. When the website is loaded it detects the amount of space that is available and select an appropriate layout size which is then displayed on screen. Adaptive Design is more suitable for designing websites that target a specific device.

2.2 Advantages and disadvantages of Adaptive Design

Benefits of Adaptive Design include allowing designers to build amazing websites tailored to a specific device and being able to optimize advertisements based on user data from smart devices. Developers are able to tailor a website to suite specific devices to produce excellent user experiences.

The downside to Adaptive Design is that it is labour-intensive and inefficient for websites designed for several devices. Websites must include layout designs for multiple screen sizes which can be time consuming. Websites are most commonly designed for desktop or smart phone devices, which can cause problems when viewing those sites from a tablet device.

3 Responsive Design

3.1 Definition of Responsive Design

Responsive Design is the design of websites in which the position of each element is determined by the available browser space. Elements on the site will automatically adjust their position as the browser size is resized. For mobile devices and tablets this happens automatically as the browsers are full screen. This form of design allows developers to design a single, “one size fits all” website that can be viewed on multiple platforms.

3.2 Advantages and disadvantages of Responsive Design

Advantages of a Responsive Design include being highly efficient due to the ability to design a single website that is able to be successfully viewed on multiple devices, reducing the amount of development time as there is no need to worry about multiple layout sizes. There are many templates that can be used to streamline this design approach.

Some of the downsides to Responsive Design are that elements can move around on the screen as the browser is resized and will sometimes be pushed into strange positions. Advertisements can also be pushed outside screen. This increases the need for testing on different screen sizes. Websites can also take slightly longer to load on mobile devices.

4 Conclusion

Our client has requested a Movie Database Web Application that is able to be accessed across all major digital platforms. For this reason we have decided to use the Responsive Design approach to produce a website that is visually appealing and responds smoothly to changes in web browser size.

This decision will allow our developers to operate more efficiently and focus on a single site that is accessible on all devices.

## Test Document

Test Data

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Description** | **Expected Result** | **Actual Result** |
| Case 1 | List all movies | All movies are loaded from the database and are shown in the table | As expected. Ref SS1. |
| Case 2 | Search by title | A movie is searched by its title | As expected. Ref SS2. |
| Case 3 | Search by genre | Movies are listed by its genre | As expected. Ref SS3. |
| Case 4 | Search by rating | Movies are listed by its rating | As expected. Ref SS4. |
| Case 5 | Search by year | Movies are listed by its year | As expected. Ref SS5. |
| Case 6 | Advanced search | Movie/s is/are searched using multiple input (title, genre, rating & year) | As expected. Ref SS6. |
| Case 7 | Top 10 most frequently searched | A graph is generated showing 10 movies and their search frequency | As expected. Ref SS7. |
| Case 8 | No movie found | The user is prompted a message | As expected. Ref SS8. |
| Case 9 | Open the application using different browsers | The application works using different browsers | As expected. Ref SS9.  Microsoft Edge shown. |
| Case 10 | Use the application in three major platforms | The application works properly and the UI adjusts smoothly | As expected. Ref SS10 & 11.  Smart phone and tablet shown. |

|  |  |
| --- | --- |
| SS1 |  |
| SS2 |  |
| SS3 |  |
| SS4 |  |
| SS5 |  |
| SS6 |  |
| SS7 |  |
| SS8 |  |
| SS9 |  |
| SS10 |  |
| SS11 |  |

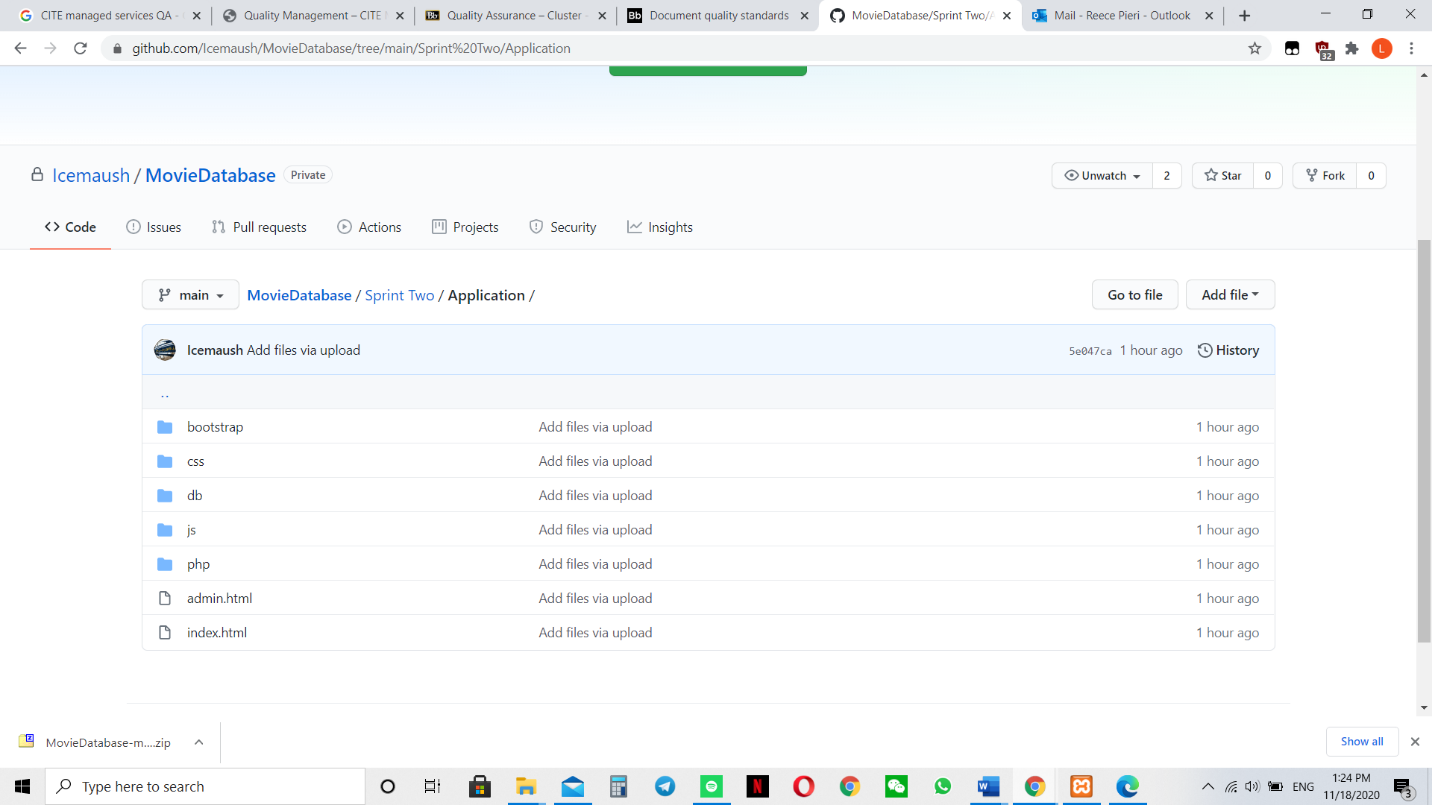
# 4 Sprint Two

## 4.1 Team

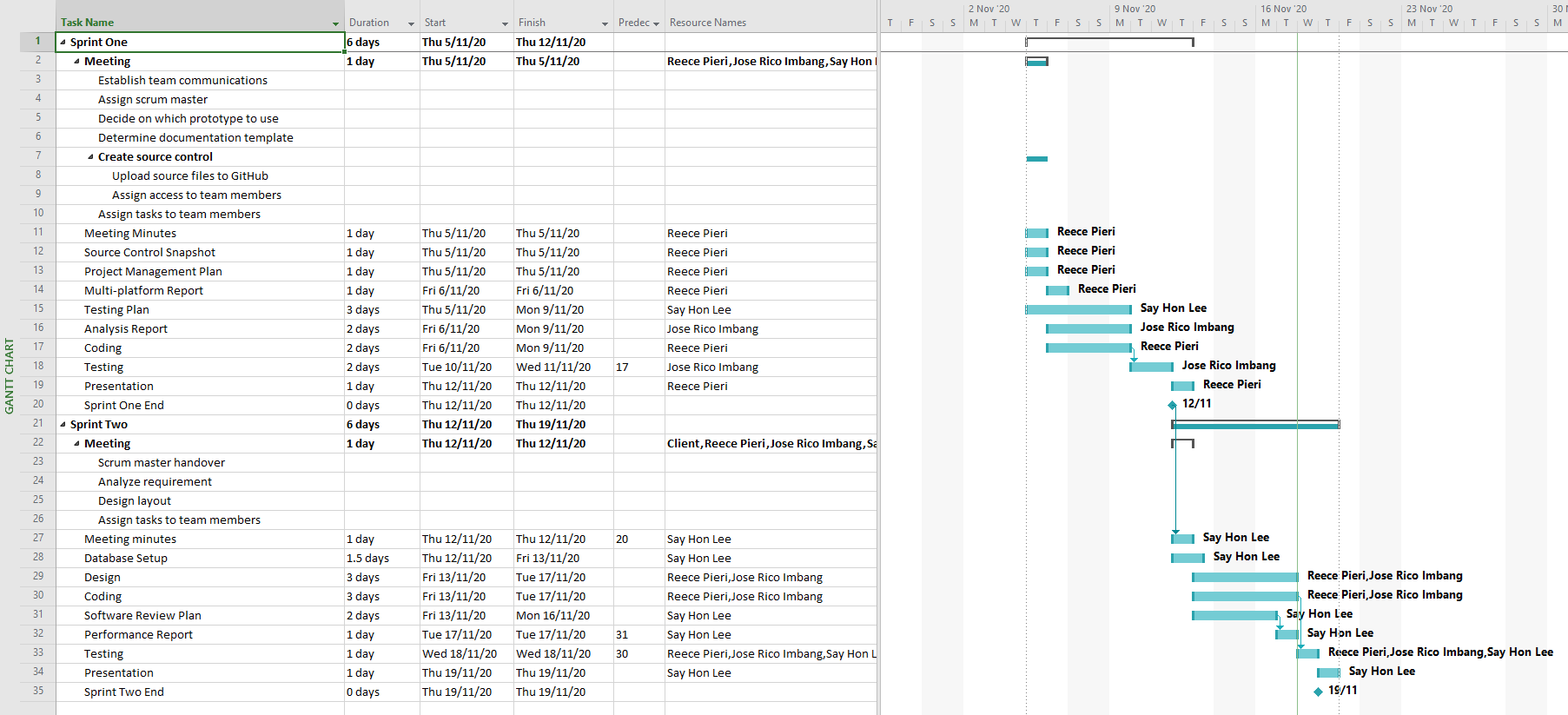
Scrum Master: Say Hon Lee

Members: Jose Rico Imbang, Reece Pieri

## 4.2 Source Code Snapshot



## 4.3 Project Management Plan



## 4.4 Software Review Plan

1 Introduction

The software review plan is a very important part of the software development life cycle (SDLC), it helps the software engineers to validate the quality, functions, and other vital requirements of the software and to examine the software (Professional QA, 2020). The software would be examined by the project personnel, managers, client, and users of the software for approval and comments for the room for improvement of the software.

There is also a minimum acceptable requirement guideline to follow according to the IEEE standard for software reviews and audits. The standard which does not meet the “systematic” minimum requirement attributes such as team participation, documented result of the review and documented procedures for reviews will be considered to be a “non-systematic” reviews. (IEEE, 2008)

2 Type Of Software Review

2.1 Software peer review

Peer review is the process of evaluating the technical content and quality of the product. It will be conducted by the developer and the scrum master. The purpose of peer review is to provide a discipline engineering practice for detecting and correcting errors of the software. (Professional QA, 2020)

Types of peer review

Code review - to fix mistakes and remove the vulnerabilities from the software product in order to further improve the quality and security of the product.

Pair programming – to review the code by all the programmers which works on a single workstation to develop a code together.

Informal – referring to a informal review of the code by the team to review the program for suggestion and also to agree that the software is of satisfactory.

Walkthrough – is where the designer or the lead of the software development team go through the software product and ask question regarding the software, make necessary comments regarding the various defects and errors of the software.

Technical review – is where a team of qualified personnel review the software and examine the software functionality and define its intended use as well as to identify various discrepancies.

Inspection – is where experienced and qualified individual examine the software products for bugs and defects using a defined process to give the team their opinion and improvement on the software if exist.

2.2 Software Audit Review

Software audit review is a type of external review where one or more auditors which is not part of the development team conduct an independent examination of the software to assess the compliance with stated specification, standards, and other important criteria. This stage is to be done by the managerial level personnel. (Professional QA, 2020)

3 Formal Review VS Informal Review

3.1 Formal review

Formal review is a type of peer review, it follows a formal process which has a specific formal agenda while running a review on the software. It has a well structured and regulated process to it. It is usually implemented at the end of each life cycle. In this process a formal review panel or board considers the necessary steps for the next life cycle. (Professional QA, 2020)

Features of formal review includes:

* Evaluation conformance to specification and standards.
* Conduct by a group of 3 or more.
* The review team petitions the management of technical leadership to act on the suggested recommendation.
* The leader have to verify that the action documents are verified and incorporated into external process.
* The six important steps of formal review consist of planning, kick-off, preparation, review meeting, rework and follow up.

3.2 Informal review

Informal review refers to the times during the development stage of the software development process of the software. The difference between formal and informal review is that the formal review is to follow a formal agenda of the software, where informal review is to be conducted anytime that is needed by any of the team. It is a more time saving process and it does not require anything to be documented neither does it requires a group of people. (Professional QA, 2020)

Features of informal review includes:

* It could be conducted with 2 or more people which includes the designer or any interested party.
* The team identifies the errors and issues and examine alternative.
* It allows the team members to learn.
* Changes will be made by the software designer.
* Changes will be verified by other project controls
* The purpose if to keep the author informed regarding the issue and to improve the quality of the software.

1. Review User

To ensure that we were to release a satisfied software to the client the programmers and the manager will first try out the software to make sure that we as a user of the software is satisfied with the software before allowing our client to try out the program. They will then give us a feedback regarding the software if they do have a suggestion for any room of improvement regarding the software. Once everything is accepted, we would also leave the contact email on the software so that the users of the software could also send us a feedback, bugs, or errors that they faced while using the software. (Professional QA, 2020)

1. Software Review Plan to be implemented

All steps of the review plan will be implemented as it is a very important process for the development of the software and also to ensure that the best of the software could be implemented and presented to the client and users without any defects in the software which we produce.

1. Software Review Checklist

**Software Review Checklist**

|  |  |  |
| --- | --- | --- |
| S/N | Requirement | Yes No |
| **UI DESIGN** | | |
| 1 | Is user interface clear and easy to use |  |
| 2 | Is the performance of the software acceptable |  |
| 3 | Is all the functions of the software (If no, please indicate below) |  |
| 4 | Is the admin page easy and clear to use |  |
| 5 | Is the software responsiveness working on multiple devices |  |
|  | **Security** |  |
| 6 | Is the admin page safely secured |  |
| 7 | Does the admin password meet the minimum complexity requirements |  |
|  | **Functions** |  |
| 8 | Are you able to search movie with just the name |  |
|  | Are you able to search for movie with just the genre |  |
| 9 | Are you able to search the movie with just the rating |  |
| 10 | Are you able to display the top 10 searched movie graph |  |
| 11 | Are you able to subscribe and unsubscribe from newsletter and flashnews |  |
| 12 | Is the admin able to receive the email when a subscriber request to unsubscribe |  |
| 13 | Can the admin search for a existing user in the admin page |  |

## 4.5 Performance Report

1 Introduction

The performance report is to collect information regarding the performance of the software, analyzing it, creating report documentation to present it to the stakeholders which is involved in the project. Performance report is part of the communication management plan.

2 Code Optimization

2.1 About code optimization

Code optimization is a programming technique which tries to help improve the code by helping it to reduce the consumption of the resources such as CPU and memory and also to allow the software to be delivered in a higher speed.

2.2 Advantages of code optimization

The advantages of the code optimization (Monus, 2020):

* Gives you cleaner code base
* Increase the consistency of the software
* Allow the software to run faster
* Increase code readability to allow the team to improve on better workflow
* Increase refactoring efficiency
* Easier code maintenance and allowing quicker feature development in future

2.3 Code optimization tool

For coding of the software, we could install and use the PHP CodeSniffer to implement in every file of the software as it is a application. PHP CodeSniffer is a application which could be installed in visual studio code which helps to do a code optimization.

3 Software Requirement

The movie website should have the following function

* Home

The home page of the website supposed to display all the movie from the database.

* Search

The search functions allow you to search for a particular movie using by Indicating the title of the movie or genre or rating or year of the movie.

* Membership subscription

Users should be able to subscribe for monthly newsletter and breaking newsflash.

* Admin login

The administrator should be able to login on the admin page to view the members which subscribed and members which request to unsubscribe for the newsletter or breaking newsflash.

4 Performance Testing

4.1 What is performance testing

Performance testing is a software testing process. The purpose of it is to test the speed, response, time, stability, reliability, scalability, and resource usage of the software under a particular workload. The main purpose of this test is also actually to identify and eliminate the performance bottleneck in the software application. It is a subset of performance engineering which is known as “Perf Testing”.

Speed testing – To determine the response of the application

Scalability testing – To find out the maximum user load available for the software to handle.

Stability testing – To determine if the application is stable under varying load.

4.2 Type of performance testing

* Load testing – The purpose of this test is to check the software ability while performing under anticipated user loads. The objective is to identify performance bottleneck before the software could goes live in future.
* Stress testing – The purpose of this application involve testing the software under extreme workload to see how the software handless with high volume of traffic or while the data is processing.
* Endurance testing – The purpose of this test is to ensure that the software is able to handle the expected load over a long period of time.
* Spike testing – The purpose of this test if to check if the software could react to a sudden large spike in the load that is generated by the users of the software
* Volume testing – The purpose of this test is to test large number of data is populated in a database and the overall software systems behaviour Is monitored. The objective is to check the software application performance under varying database volumes.
* Scalability testing – the purpose for this testing is the check the software application effectiveness while “scaling up” to support the increase in user load. It helps to plan the capacity addition to the software system.

4.3 Common performance issue

Below is a list of the common performance issue with a web application which could cause the company to lose potential users.

* Excessive load time – The load time is usually the time taken for the application to start up. It should be kept to the minimum. While some application are impossible to make load in under a minute, load time should be kept under a few seconds if it is possible.
* Poor response time – The response time is the time the application take when a data is input into the application to the time where the application outputs a response to the input. Generally, it should be done as soon as possible, if it takes too long, users might lose interest as well.
* Poor scalability – Poor scalability affect the software product if it is unable to handle the expected number of users or when it is not able to accommodate a wide enough range of users. Load testing should be done to be certain the application have the ability to handle the anticipated number of users.
* Bottlenecking – Bottlenecking is a obstruction in a system which degrade the overall system performance. Bottlenecking is when either coding errors or hardware issues cause a decrease of throughput under certain loads. Bottlenecking is often caused by one faulty section of code. In order to fix a bottlenecking issue is to find the section of code that causes the slow process and fix it. It is generally fixed by either fixing the poor running processes or adding additional hardware. The common performance bottlenecks are:
  + CPU utilization
  + Memory utilization
  + Operating system limitation
  + Disk usage

4.4 Performance Testing Process

There is a wide variable for performance testing, but the objective for performance test remains the same. It helps to demonstrate that your software system is able to meet certain pre-defined performance criteria. Or it can help compare the performance of two different software system. It also could help identify part of the software system which degrade the performance.

The steps to performance testing process is to:

* Identify the testing environment – knowing what’s your physical test environment, production environment and what testing tools are available. Understand the details of the hardware, software and network configuration used during testing before you begin the testing process. It helps to identify possible challenges that users may encounter during the performance testing procedure.
* Identify the performance acceptance criteria – this includes goals and constraints for the throughput, response time and resource allocation. It is necessary to identify the project criteria outside of the goals and constraints. Testers should be empowered o set performance criteria and goals as the project specification might not includes a wide enough variety of performance benchmarks. It is also a good way to find a similar application to compare which helps to set a performance goal.
* Plan and design performance test – this determine the usage amongst the end user and identify key scenarios to test for all possible use cases. It is necessary t simulate a variety of end user, plan performance test data and outline what metric will be gathered.
* Configure test environment – This is to prepare the testing environment before executing the software. Also, to arrange tools and other resources.
* Implement test design – Create the performance test according to the test designs.
* Run the test – execute the software and monitor the test.
* Analyse, tune and retest – consolidate, analyse and share the test result. If needed, fine tune the software and test it again to see if there is and improvement or decrease in performance of the software. Improvements generally grow smaller with each retest, stop when bottlenecking is caused by the CPU and consider the option of increasing the CPU power.

4.5 Performance Testing Metrics

This refers to the basic parameter that will be monitored during the performance testing.

* Processor usage
* Memory use
* Disk time
* Bandwidth
* Private bytes
* Committed memory
* Memory pages/ second
* Page faults/second
* CPU interrupts per second
* Disk queue length
* Network output queue length
* Network byte total per second
* Responds time
* Throughput
* Amount of connection pooling
* Maximum active sessions
* Hit ratios
* Hits per second
* Rollback segment
* Database locks
* Top waits
* Thread counts
* Garbage collection

## 4.6 Test table

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Description** | **Expected Result** | **Actual Result** |
| Case 1 | Members sign up | Members sign up successfully | As expected. Ref SS1. |
| Case 2 | Members selection to subscribe | Members able to choose what they would like to sign up to | As expected. Ref SS2. |
| Case 3 | Members unsubscribe | Members unsubscribe successfully | As expected. Ref SS3. |
| Case 4 | Admin Login | Admin login successfully | As expected. Ref SS4. |
| Case 5 | Admin display all members | All subscribed members displayed | As expected. Ref SS5. |
| Case 6 | Admin display members unsubscribe request | Unsubscribe request displayed | As expected. Ref SS6. |
| Case 7 | Admin able to search for members to unsubscribe | Searching of members found | As expected. Ref SS7. |
| Case 8 | Admin to receive a email when a member request to unsubscribe | Email received | As expected. Ref SS8. |
| Case 9 | Use the application in three major platforms | The application works properly and the UI adjusts smoothly | As expected. Ref SS9, Ref SS10, Ref SS11, Ref SS12. |

|  |  |
| --- | --- |
| SS1 |  |
| SS2 |  |
| SS3 |  |
| SS4 |  |
| SS5 |  |
| SS6 |  |
| SS7 |  |
| SS8 |  |
| SS9 |  |
| SS10 |  |
| SS11 |  |
| SS12 |  |

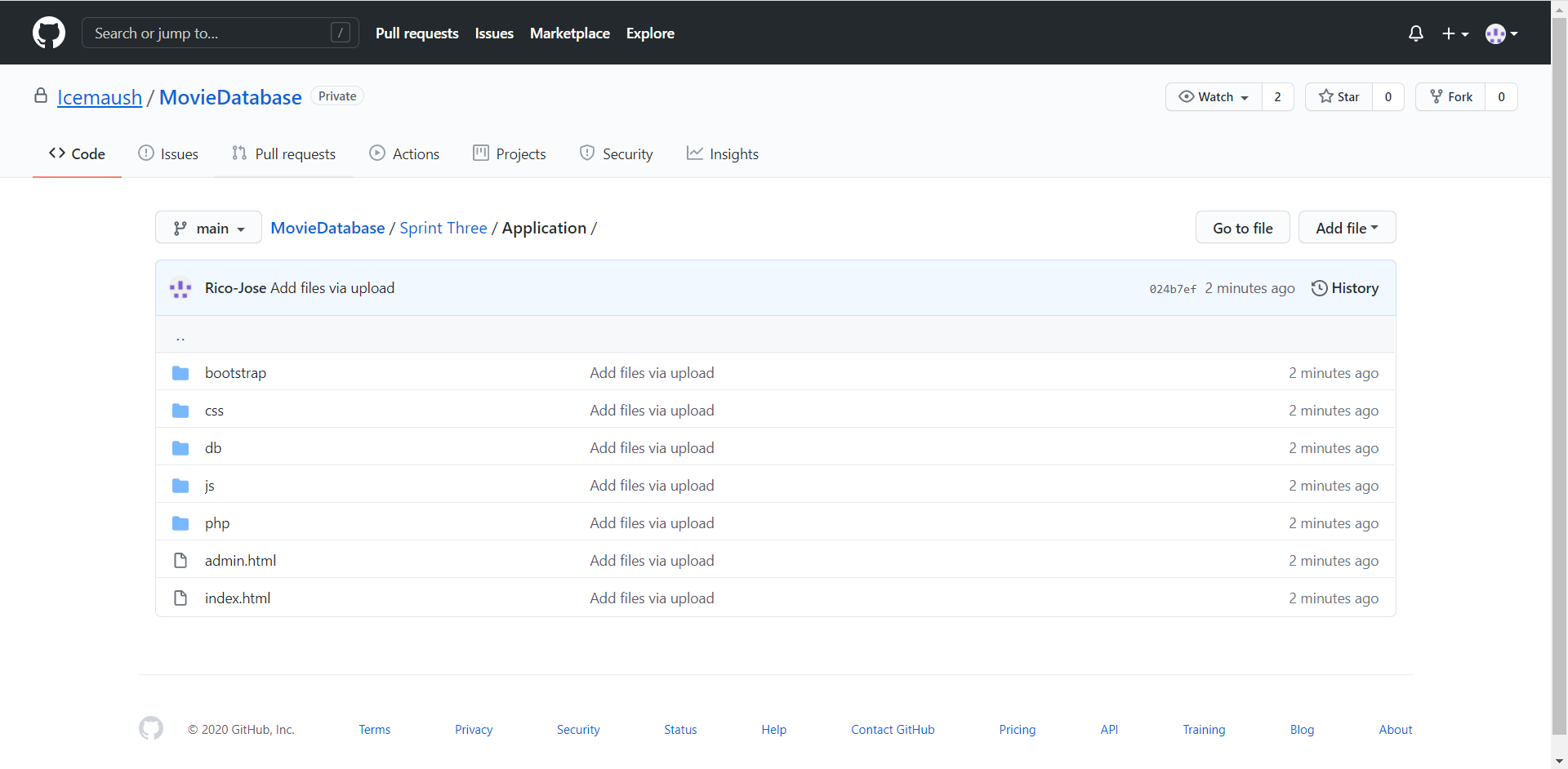
# 5 Sprint Three

## 5.1 Team

Scrum Master: Jose Rico Imbang

Members: Reece Pieri, Say Hon Lee

## 5.2 Source Code Snapshot



## 5.3 Project Management Plan

Graphical user interface, application

Description automatically generated

## 5.4 Optimisation Report

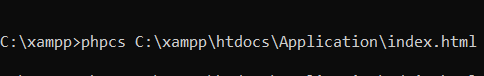
1 Introduction

This optisation report shows the errors of the HTML and PHP code which have been fixed in the software.

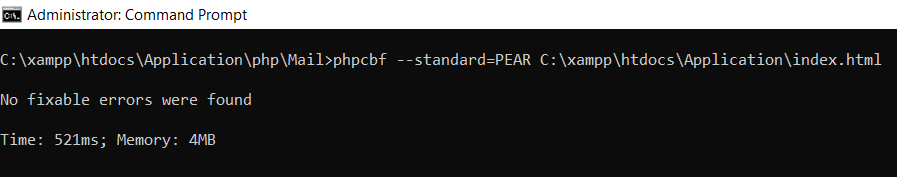
We have chosen to use PHP CodeSniffer to do our code optimisation. We will include the numbers of errors remaining as well as the number of errors remaining in the code which could not be fixed by using the code optimiser.

2 Code Optimisation Done for HTML

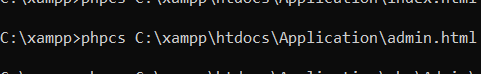
2.1 Index



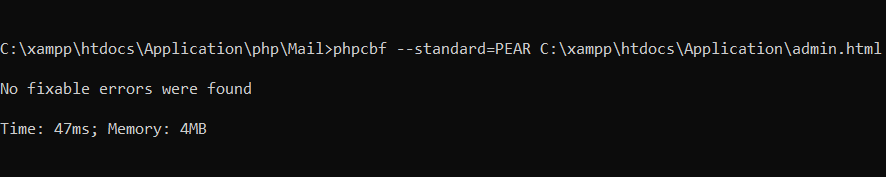
There is no error in the file, nothing have been changed.



2.2 Admin



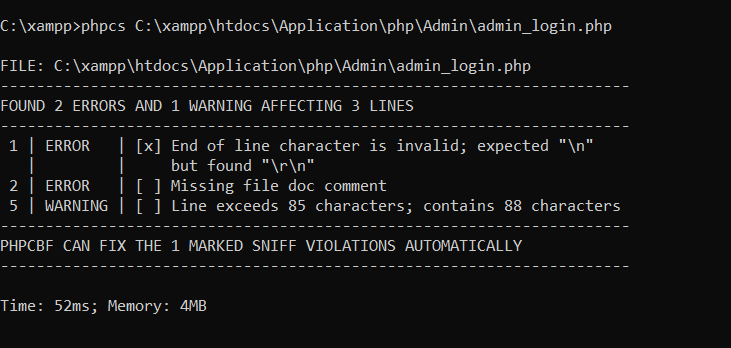
There is no error in the file, nothing have been changed.



3 Code Optimisation Done for PHP

3.1 Admin

This is the number of issue resolved by using PHP Code Sniffer for the whole of admin php file. The changes have been made and displayed in the end of the admin page screenshots.



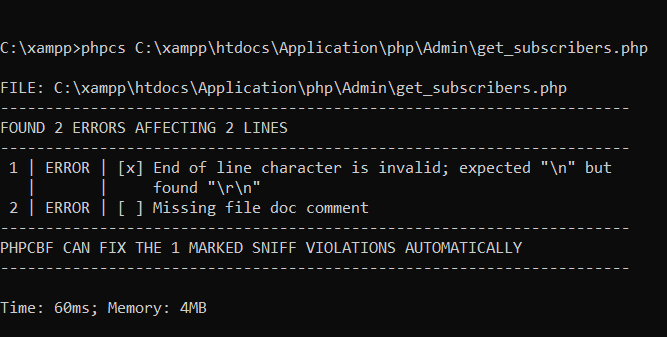
Initial amount of errors in admin\_login.php



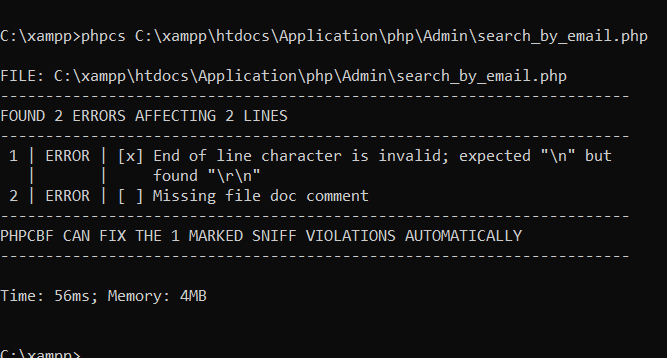
Initial amount of errors in create\_ admin\_account.php



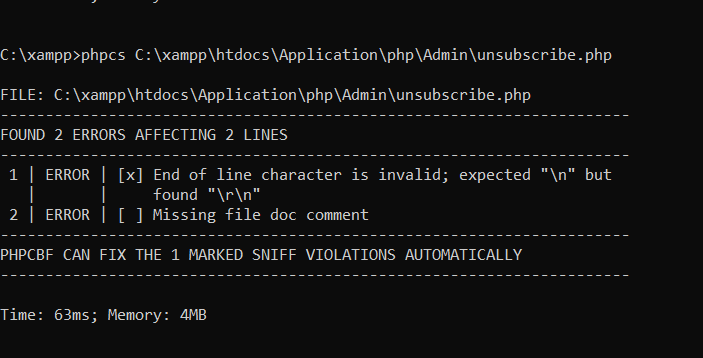
Initial amount of errors in get\_pending\_removals.php



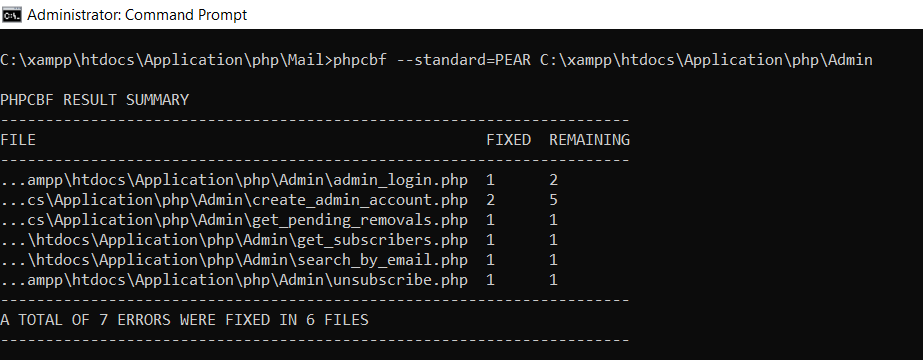
Initial amount of errors in get\_subscribers.php



Initial amount of errors in search\_by\_email.php



Initial amount of errors in unsubscribe.php



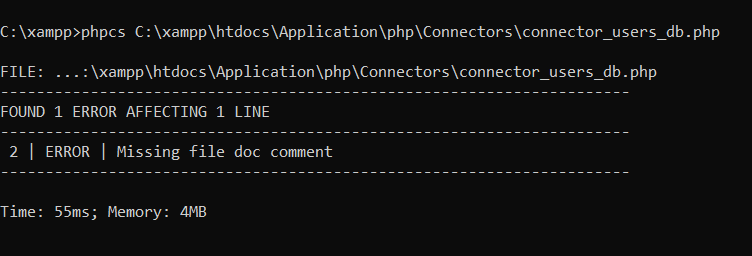
This shows the number of errors which have been fixed in the Admin file by using PHP CodeSniffer as well as the remaining amount of errors.

3.2 Connectors

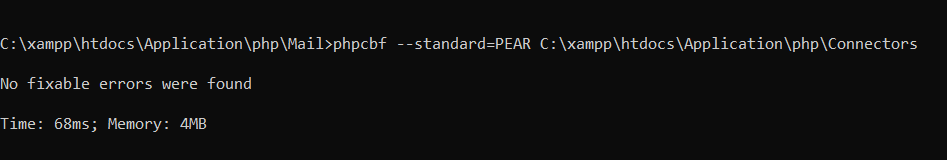
This is the number of issue resolved by using PHP Code Sniffer for the whole of connector php file.



Initial amount of errors in connectors\_movies\_db.php



Initial amount of errors in connectors\_users\_db.php



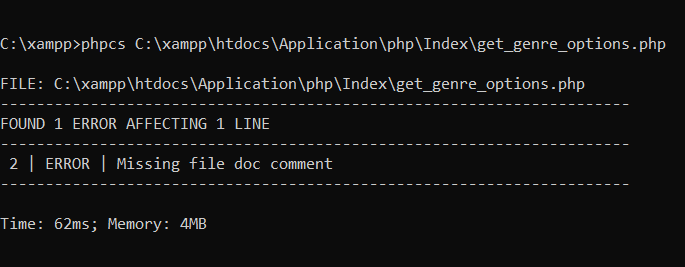
The error which occurs in the connector file could not be resolved by PHP CodeSniffer.

3.3 Index

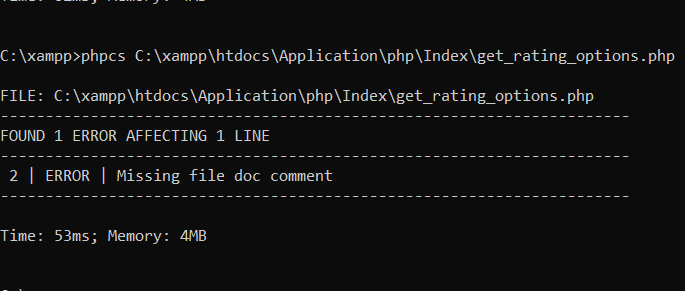
This is the number of issue resolved by using PHP Code Sniffer for the whole of Index php file.



Initial amount of errors in find\_member.php



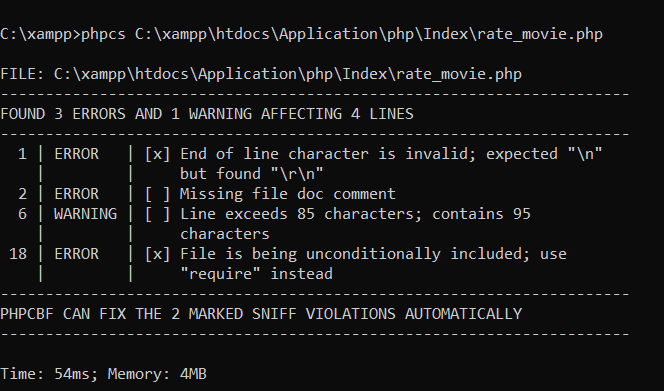
Initial amount of errors in get\_genre\_options.php



Initial amount of errors in get\_rating\_options.php



Initial amount of errors in get\_top\_ten\_graph.php



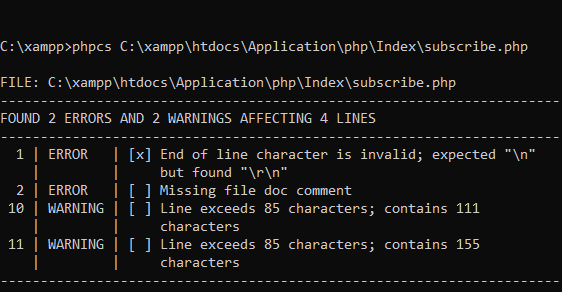
Initial amount of errors in rate\_movie.php



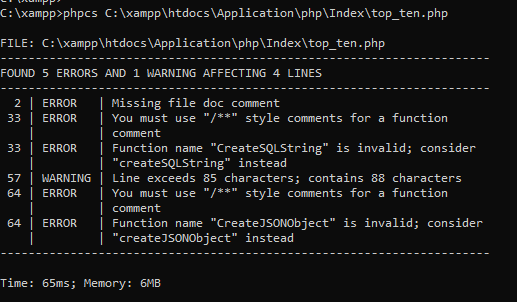
Initial amount of errors in get\_year\_options.php



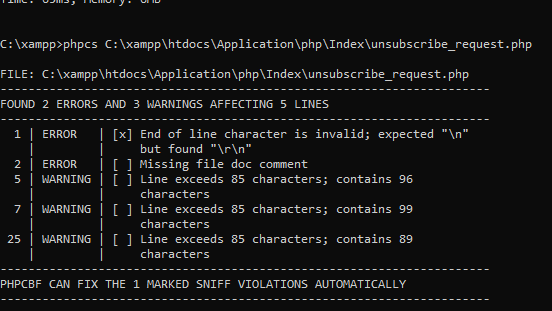
Initial amount of errors in search\_db.php



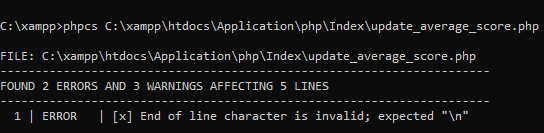
Initial amount of errors in subscribe.php



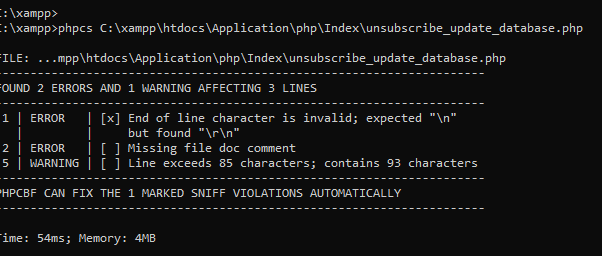
Initial amount of errors in top\_ten.php



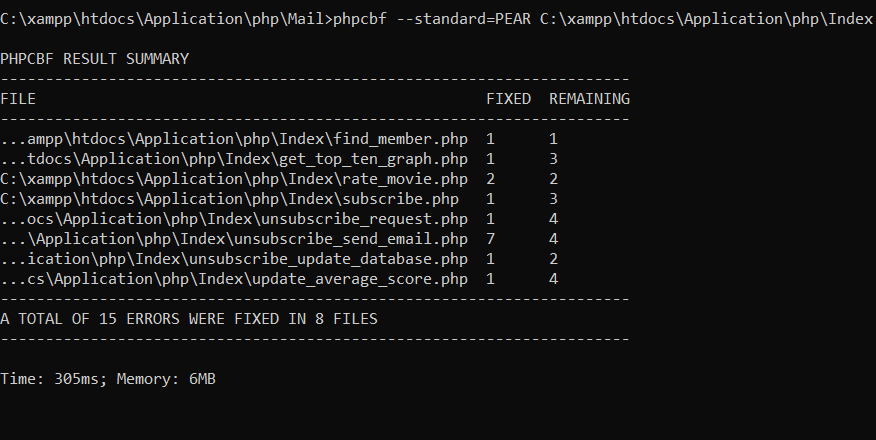
Initial amount of errors in unsubscribe\_request.php



Initial amount of errors in update\_average\_score.php



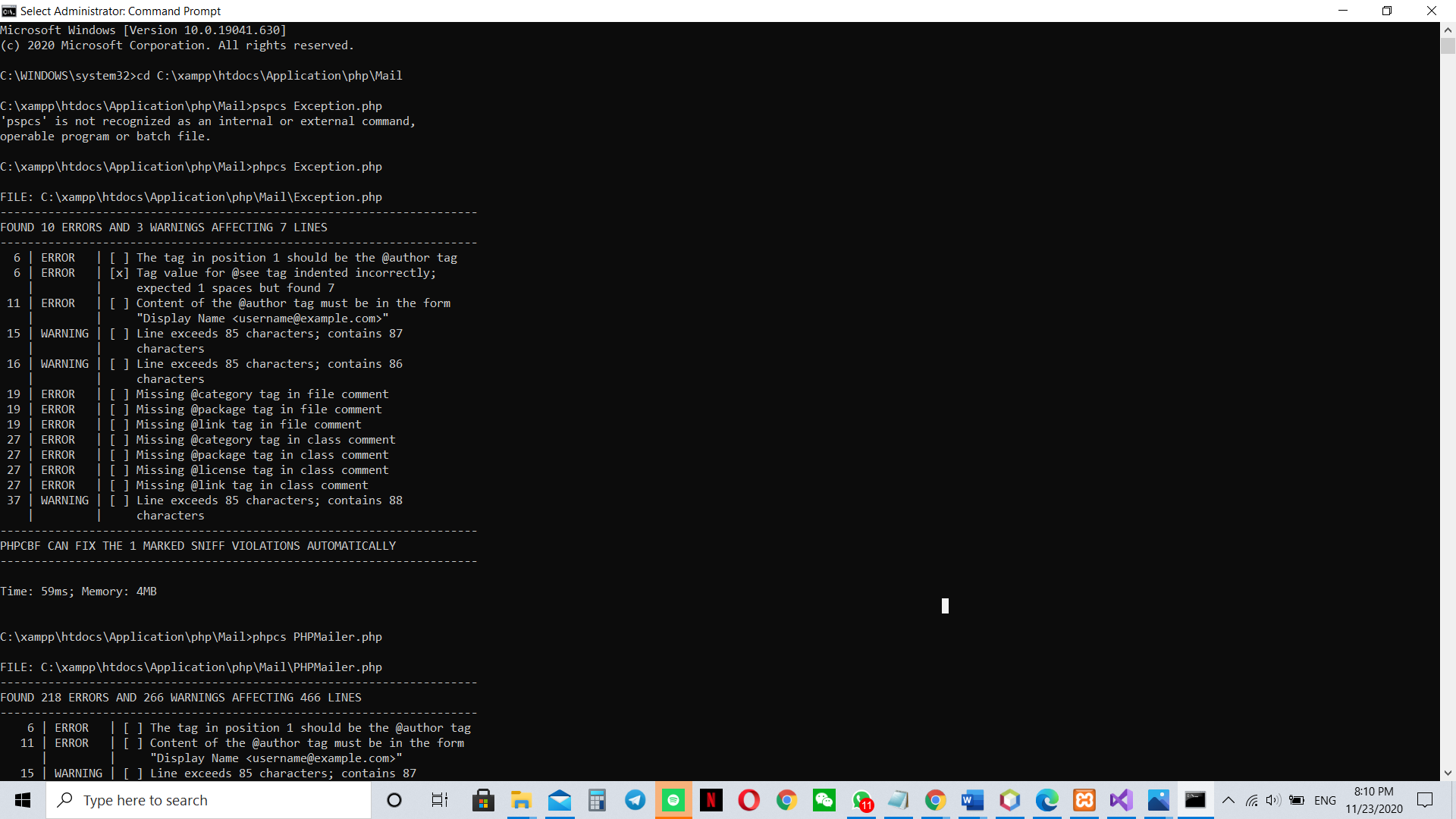
Initial amount of errors in unsubscribe\_update\_database.php



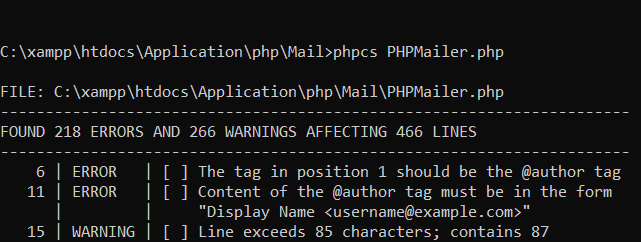
This shows the number of errors which have been fixed in the index file by using PHP CodeSniffer as well as the remaining amount of errors.

3.4 Mail

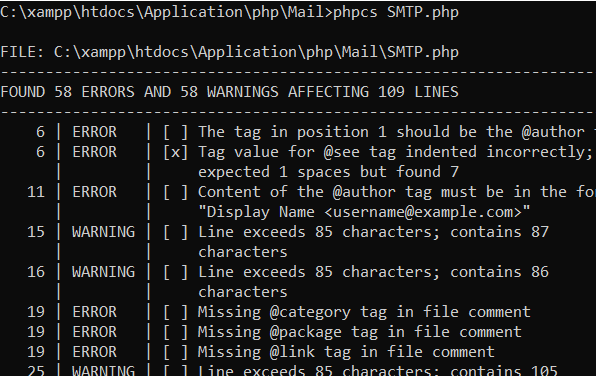
This is the number of issue resolved by using PHP Code Sniffer for the whole of Mailer php file.



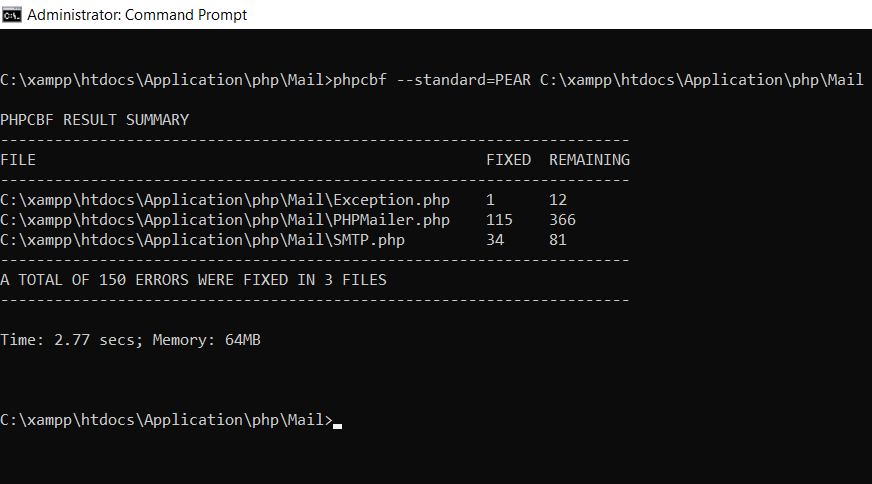
Initial amount of errors in Exception.php



Initial amount of errors in PHPMailer.php



Initial amount of errors in SMTP.php



This shows the number of errors which have been fixed in the Mail file by using PHP CodeSniffer as well as the remaining amount of errors.

## 5.5 Software Testing Plan

1 Introduction

1.1 Purpose of the Software Development Testing Plan

The purpose of the Software Development Testing Plan is to outline the scope and strategies that will be applied to the testing of the Movie Database application.

1.2 Scope

The functional requirements of the Movie Database application include:

* Acme personnel can log in (security interface)
* Users can rate movies
* A table shows the Top 10 highest rated movies

The non-functional requirements of the Movie Database application include:

* Password must pass a minimum complexity
* The Top 10 highest rated movies table refreshes automatically to show the updated information

1.3 Quality Objective

* Ensure the Application Under Test’s functional and non-functional requirements meet the client expectations.
* Bugs and issues are identified and fixed before presenting the product to the client.

1.4 Roles and Responsibilities

* Scrum Master – Jose Rico
* Developer – Reece
* Tester – Reece and Jose Rico
* Technical Documentation – Say Hon and Jose Rico

2 Test Methodology

2.1 Overview

Due to the client’s requirement not being clearly outlined at the beginning of the project, the testing methodology best suited for this project is the Agile methodology due to its flexibility and the need to present a working prototype to the client frequently. This allows the improvement of the product quality in a progressive manner.

2.2 Test Levels

Four stages of testing will be applied to the Movie Database application. These include:

* Unit testing – it is a smallest testable portion of the systems ensuring it could be compiled, loaded and executed.
* Integration testing – as the website is connected to a database we will also need to ensure that data would be displayed onto the webpage.
* System testing – it refers to checking the overall interaction of the components built in the webage to check on the loading, performance, reliability and security of the webpage.
* Acceptance testing – is to test the conduct to find if the requirements of a specification or contract are met as per its delivery. It will be tested by the clients to ensure the level of satisfactory has been met.

3 Test Deliverables

3.1 Test Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Description** | **Expected Result** | **Actual Result** |
| Case 1 | Security interface | Acme personnel can log in | As expected. Ref SS1. |
| Case 2 | Movie rating | Users can rate the movies | As expected. Ref SS2. |
| Case 3 | Analytics | A table shows the Top 10 highest rated movies | As expected. Ref SS3. |
| Case 4 | Update analytics | Top 10 highest rated movies table refreshes automatically to show the updated information | As expected. Ref SS4. |
| Case 5 | Multi-platform | The application works properly and the UI adjusts smoothly in an iPad Pro | As expected. Ref SS5. |
| Case 6 | Multi-platform | The application works properly and the UI adjusts smoothly in an iPhone X | As expected. Ref SS6. |

|  |  |
| --- | --- |
| SS1 |  |
| SS2 |  |
| SS3 |  |
| SS4 |  |
| SS5 |  |
| SS6 |  |

4 Resources & Environment

4.1 Testing Tools

Every modern web browser includes a powerful suite of developer tools. These tools do a range of things, from inspecting currently-loaded HTML, CSS and JavaScript to showing which assets the page has requested.

We used the developer tool to see what the HTML on the page looks like at runtime, as well as what CSS is applied to each element on the page. The developer tool allowed us to instantly modify the HTML and CSS and see the results of the changes reflected live in the browser viewport. We also used the developer tool to see how the application is going to look like in different platforms like tablet and cellphone.

4.2 Test Environment

The application is tested using a PC running a Windows 10 operating system. The PC also has a XAMPP installed to be able to use MySQL and so the PC can create a local server. The browser that is mainly used is Google Chrome.