Software Review Report

South Metropolitan TAFE | RAD

Carl Haricombe

HandOver

2020

Table of Contents

[How the team ensured the development quality of the software during the three sprints 1](#_Toc57897595)

[What processes were utilised to ensure adequate control of the quality of the development process. 2](#_Toc57897596)

[Describe the code testing process. 3](#_Toc57897597)

[Software Verification: 3](#_Toc57897598)

[Software Validity: 3](#_Toc57897599)

[How can the software be supported for future modification and refinements (comments, code documentation, development plans). 4](#_Toc57897600)

[Provide a mapping of the user requirement to the final software product (did you meet the clients expectations and requirements) 5](#_Toc57897601)

# How the team ensured the development quality of the software during the three sprints

We compared our updated site with the expectations and requirements of the client and the previous website. The methods we used to go about this were pretty simple, when each team member completed their end of the assignment, we would upload it to Github and notify the others to check that all the needs of the client had been for example the client wanted a multiplatform website and so we went on a venture to find the best and quickest solution which we believed to have been a responsive website.

We ensured that we followed the QA cite ISO 9001 and 90003 with regards to the code testing and processes of the website. Making use of dynamic verification method where we are able to monitor the working behavior of the software while executing the prototype. And blackbox validation method to test the front end and assure the client that the backend is working accordingly to the requirements they had given us for the 3 sprints.

# What processes were utilised to ensure adequate control of the quality of the development process.

|  |  |
| --- | --- |
| **GUI Design** | |
| Easily visible for people with Colourblind disabilities | Passed |
| Text is clear to see | Passed |
| **Functionality** | |
| Search a movie by title etc | Passed |
| Display graph of most searched movies | Passed |
| Rate a movie | Passed |
| Display graph of movie ratings | Passed |
| Create a new user | Passed |
| User can delete account | Passed |
| Admin can remove a user | Passed |
| **Security** | |
| User login through name and gmail address | Passed |
| Admin login through usercode, gmail and password | Passed |

# Describe the code testing process.

### Software Verification:

#### Dynamic verification

This focuses on the working behavior of the software and is done during the execution of the system.

### Software Validity:

#### Black Box testing

This is where the internal design of the prototype that is being tested, isn’t directly known to the tester. In other words, the team tested only the front end of the final product at the end of each sprint.

We also made use a performance report which was used to test the responsiveness and speed of the website.

We also used an Optimization report where we used code sniffer to detect any bugs or glitches in our site or where there was any syntax or logic errors.

# How can the software be supported for future modification and refinements (comments, code documentation, development plans).

## Comments:

Whoever may need to carry the website on further can just follow the comments to understand what certain algorithms were used for and how we went about it. Our comments clearly define what the purpose was shows the additional code to the new website as opposed to the previous.

## Code Documentation:

Our documentation such as our test documentation and master documentation clearly define the different areas that needed to be added such rating a movie and adding a graph as well as making the website responsive. So by using our code documentation you can take it further and add on to where the next changes may be made.

## Development plans:

After every sprint there was a development plan to display who did certain tasks in that sprint and who checked that person’s work.

# Provide a mapping of the user requirement to the final software product (did you meet the clients expectations and requirements)

Yes, I believe the team has met the client’s requirements. In saying so we had tested one another’s work after every sprint to ensure each requirement was met no matter how small the task. We were able to do so through github when we’d upload our end and communicate with one another to find out what could be done better or what was missing.

Furthermore, we presented the final product for that sprint at the end of each sprint to ensure the client is pleased with the result before deployment.