Master Documentation

Sprint Three

Team F

Table of Contents

[Source Control 2](#_Toc43290971)

[Optimization Report 3](#_Toc43290972)

[Introduction 3](#_Toc43290973)

[Legibility/Beatify 3](#_Toc43290974)

[Security 4](#_Toc43290975)

[Ajax Optimization 5](#_Toc43290976)

[Search Optimization 5](#_Toc43290977)

[Summary 6](#_Toc43290978)

[Project Management Plan 7](#_Toc43290979)

[Software Testing Plan 8](#_Toc43290980)

[Scope 8](#_Toc43290981)

[In Scope 8](#_Toc43290982)

[Functional(foundamental): 8](#_Toc43290983)

[Non-functional: 8](#_Toc43290984)

[Quality Objective 8](#_Toc43290985)

[Roles and Responsibilities 8](#_Toc43290986)

[Methodology 9](#_Toc43290987)

[Agile model and RAD model 9](#_Toc43290988)

[Test method 9](#_Toc43290989)

[Test Completeness 10](#_Toc43290990)

[Resource & Environment Needs 10](#_Toc43290991)

[Testing Tools 10](#_Toc43290992)

[Test Environment 10](#_Toc43290993)

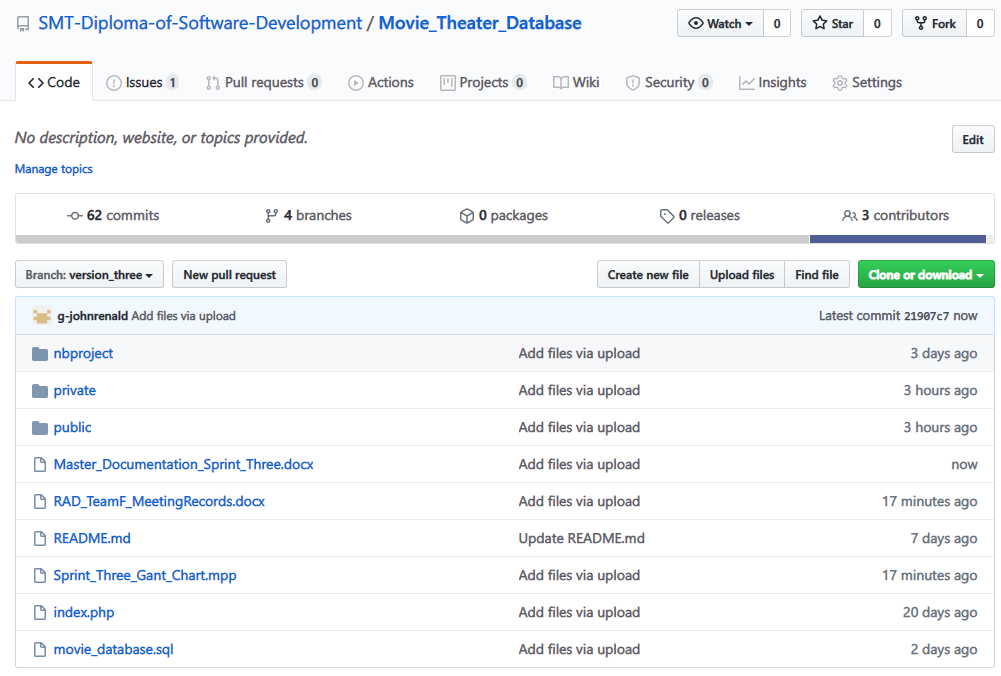
[Test Deliverables 10](#_Toc43290994)

[Test case: 10](#_Toc43290995)

[Report 12](#_Toc43290996)

# Source Control

The snapshot below is the version three of the Movie Theatre Database. The new branch named version three is created, master documentation and meeting records are also included.



<https://github.com/SMT-Diploma-of-Software-Development/Movie_Theater_Database>

# Optimization Report

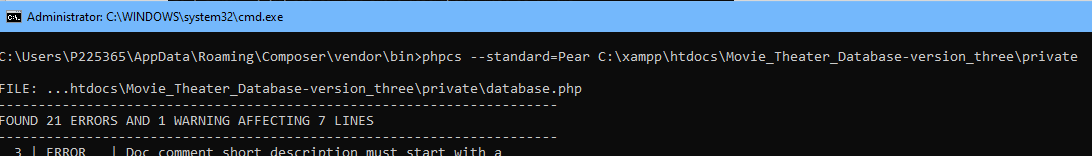
## Introduction

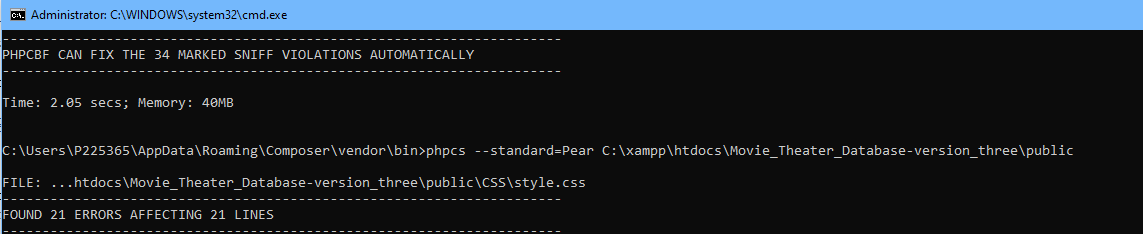
This report will cover and outline the changes made to implementation in the pursuit of improving and optimizing the solution for greater security, stability, legibility and full proofing additionally this report also covers the testing plan’s effectiveness and optimizes it where deemed appropriate.

## Legibility/Beatify

Code reviewed manually for efficiency, checking for redundant lines, followed by using the optimization tool ‘PHP Beautify’ to the ‘Pear’ conventions/standards.

Below are screenshots of PHP Code Sniffer being applying the Pear standard to the private a public (in that order) php pages of the project.

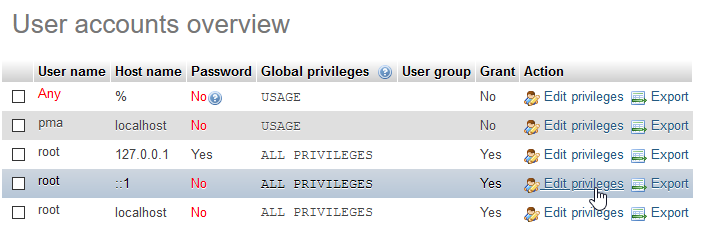




This process involves automatic uniform spacing in lines and manual changes that cannot be done automatically such as character line limits, the changes are visual only and they do not provide any functional optimization, they are solely for the purpose of legibility.

## Security

Setting a password to the server database’s ‘root’ login and editing the Xampp config file on the designated server so that it provides the correct password on prompt.



Setting a password to all 3 ‘root’ logins to avoid any other devices on the LAN from obtaining database information, especially passwords as they do not include any form of encryption.

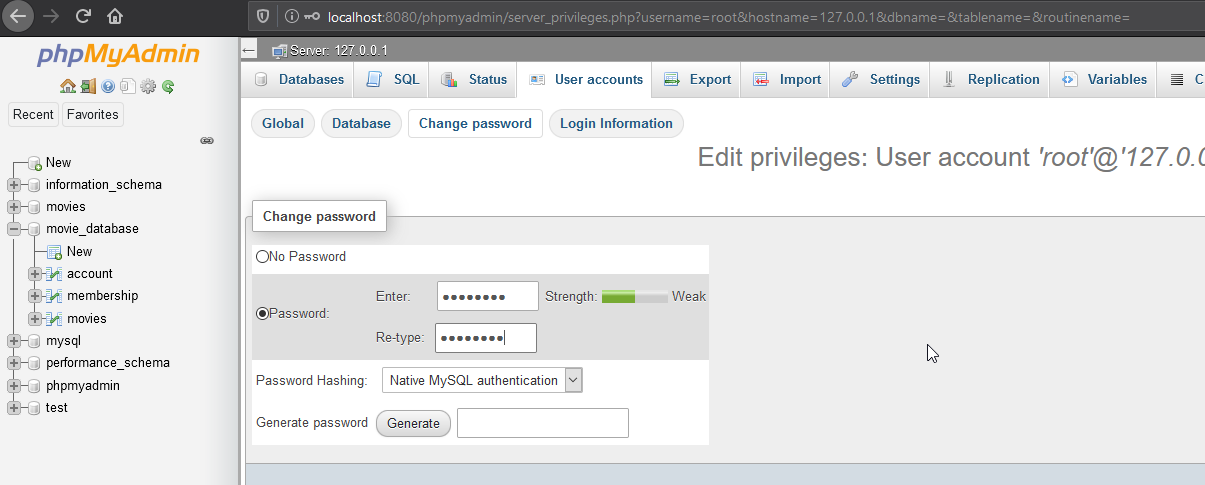
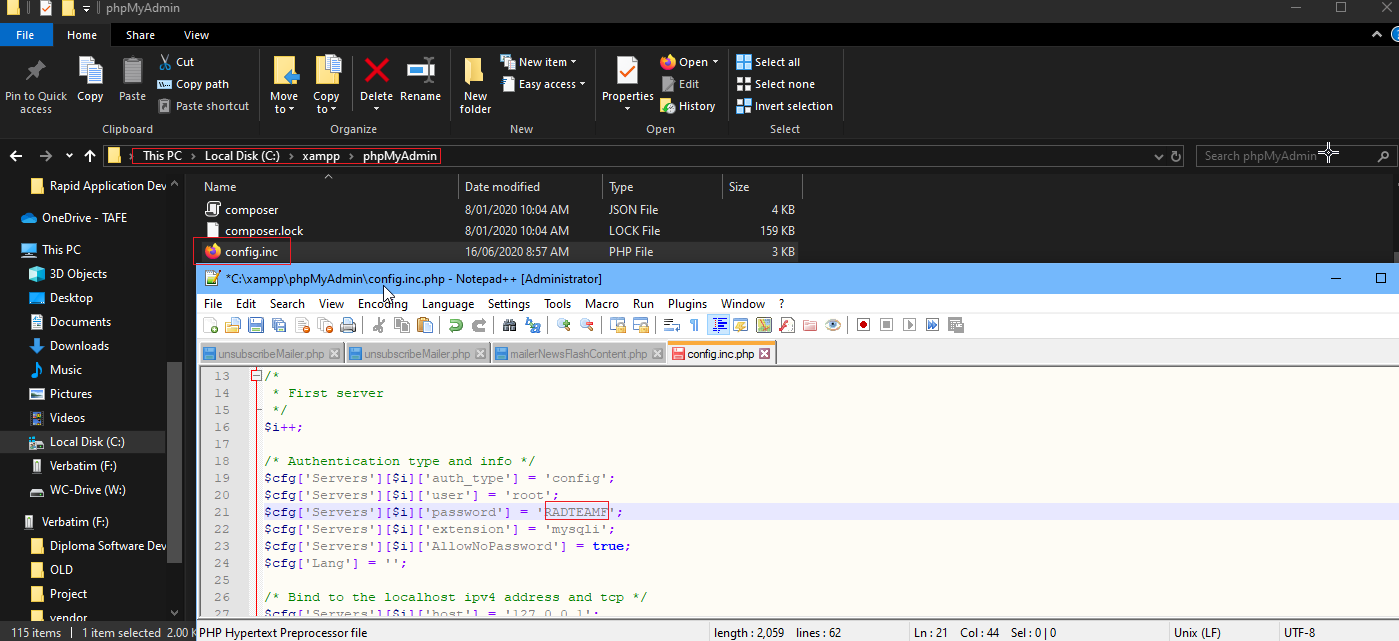
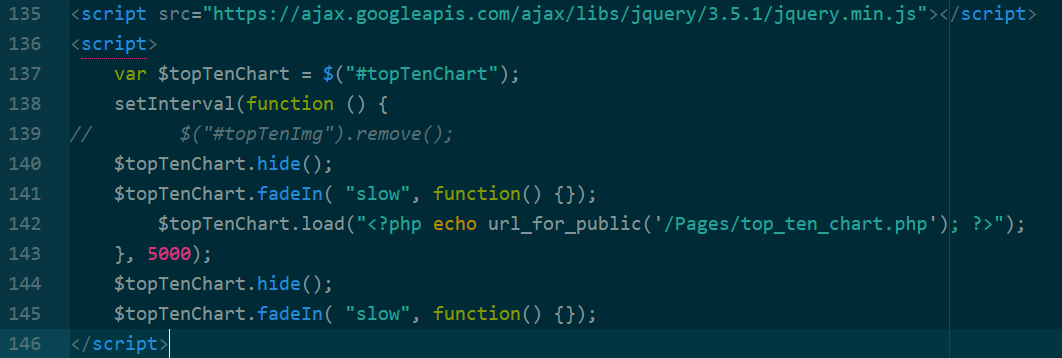


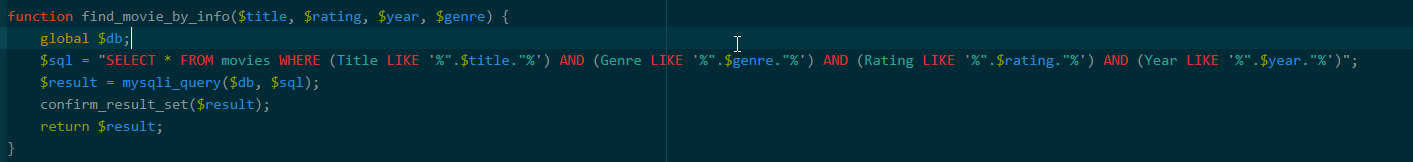
Image below shows the config file edited with a password ‘RADTEAMF’ which is the same password that should be set in the phpMyAdmin GUI.

## Ajax Optimization

Currently the Ajax implementation works correctly but its refreshing is unpleasant to the eyes as the elements appear jumbled and flashy for a split second before refresh, the rate could be reduced as well.  
A transition effect ‘fadein()’ and ‘hide()’ should be applied with jQuery to avoid the Ajax script pronouncing itself as flashy page refreshing to the viewer.

  
  
Using ‘hide()’ and ‘fadein()’ before and after rather than just after allows the page to hide its old version before refreshing, this prevents elements from becoming jumbled on the page for that split-second.

Search Optimization

The php page ‘query\_functions\_movie.php’ in private folder is modified now to give a more comprehensive search. Using the SQL ‘LIKE’ function instead of selecting exactly by variable and concatenating inside ‘if’ statements. 

This approach allows users to find the movie they are after even if they do not know the precise name of the title, genre, rating or year it was produced in, this means there is now a lot more search results on every query.

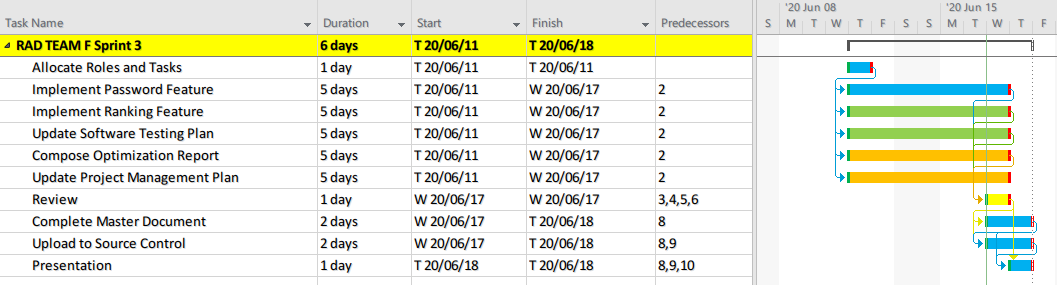
## Summary

For the purpose of optimizing the prototype - the search method, the ajax script and our database security have been modified, these modifications should improve the user experience as well appease the client’s security concerns. The whole project has been put through PHP Code Sniffer and formatted to the ‘PEAR’ coding conventions, for legibility and code optimization.

For further optimizations we can continue by including encryption and Including a change password feature for staff and admin, as these are substantial modifications - we will leave that for the client to consider first.

# Project Management Plan

As trial plan for Monday is expired, we moved to using Microsoft Project. The actual MS project file is available on GitHub.



# Software Testing Plan

## Scope

### In Scope

Defines the features, functional or non-functional requirements of the software that **will be** tested

### Functional(foundamental):

* Calculate top 10 popular movies when page open.
* User add rating to movie
* Top 10 movies appended to streaming table and display to front end

### Non-functional:

* Appropriate GUI
* Able to be used in Multi-Platform
* Top ten table and chart should be refresh when user evaluate movies or every XX seconds.

### Quality Objective

* Ensure the Application Under Test conforms to functional and non-functional requirements
* Product quality verification and validation to ensure that it complies with clients’ business needs and expectations
* Bugs/issues are identified and fixed before release
* GUI provide user a good experience

### Roles and Responsibilities

* CITE Managed Services QA department: QA testing, configuration manage, risk manage
* Test Manager
* Developers and tester

## Methodology

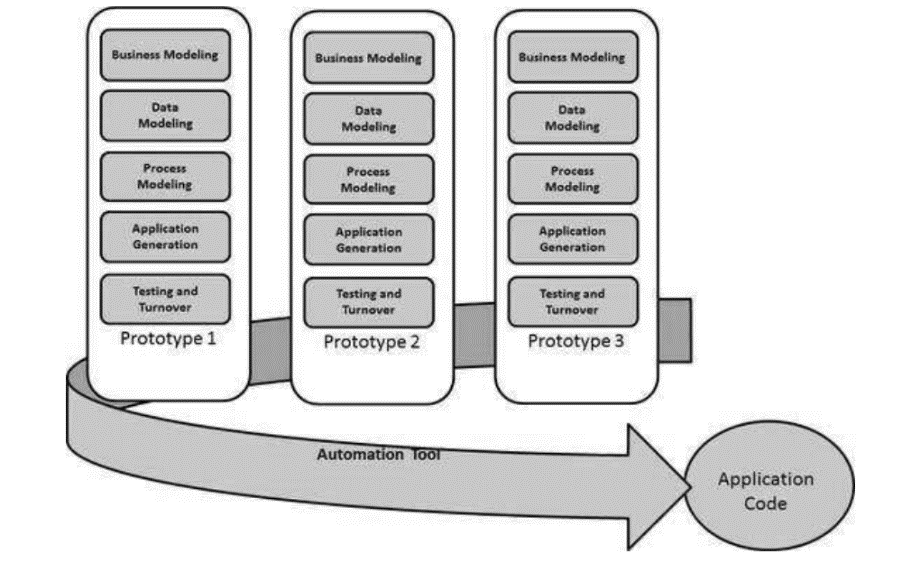
This project will adopt Agile model and RAD model, testing phase will be in the end of each iteration.

### Agile model and RAD model

Agile model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration (Requirement->Design->Development->**Testing**->Evaluation) typically lasts from about one to three weeks. **At the end of the iteration, a working product is displayed to the customer and important stakeholders**. (tutorialspoint, 2019)

Hence testing phase will execute in each interation, it ensure bug can be found early.

Rapid Application Development (formerly known as Rapid Application Development, abbreviated as RAD) is a methodology for software development. RAD methodology is quite different with traditional SDLC, traditional SDLC emphasize requirement analysis before code start. Changing requirement may not so feasible. However**, RAD emphasize the iterative and incremental delivery of working models to the customer rapidly and allow customer give feedback**.



### Test method

#### White box testing:

White box testing or structural testing is tested by who understand the structural or implementation of application. Tester understand how the application works. Usually, white box testing tests application’s internal structure or working. Testers will choose inputs to exercise path through the code and determine the expected outputs.

### Test Completeness

Criterias that deem testing complete.

* 100% test coverage (all functionalities)
* All Test cases executed manually
* Software are tested under a various of platform.
* All testing result will be documented.

## Resource & Environment Needs

### Testing Tools

* XAMPP or server environment
* Multiplatform simulation environment (Google Developer tools)
* Ajax, JavaScript
* Testing document
* Computer

### Test Environment

* OS: Windows 10 64bit
* Processor: Intel Core i7
* Memory: 16 GB RAM
* Graphics: NVIDIA GeForce GTX 1650
* Storage: 100 GB available space

## Test Deliverables

Here mentions all the Test Artifacts that will be delivered during different phases of the testing lifecycle.

### Test case:

**Rating formula:**

searchTimeRating=round(($movie['search\_times']-$minSearchtimes)/(($maxSearchtimes-$minSearchtimes)/5));

Client rating =round(($movie['clients\_rating'] \* $movie['evaluated\_times'] + $\_POST['clients\_rating']) / ($movie['evaluated\_times'] + 1),1)

rating=round($clientRating\*0.6+$searchTimesRating\*0.4);

Testing video<https://www.youtube.com/watch?v=w2YvQ36_8uA>

**Functional test:**

**Variable definition: ST(Search time) MaST(max search time) MiST(min search time) CR(client rate) CRT(client rate times) FCR (final client rate)**

**Notice : final client rating will not store in database**

|  |  |  |  |
| --- | --- | --- | --- |
| Test case (In laptop 1440 X 900) | Data inputted | Expect Result | Result |
| Using searched table to evaluate movie | 0 star (ST=11,MaST=11,MiST=0 CRT=0,CR=0) | Database changed (CR=0, FCR=2) and direct to homepage | Correct result show in testing video |
| Using searched table to evaluate movie | 5 star  (ST=11,MaST=11,MiST=0 CRT=0,CR=0) | Database changed (CR=5, FCR=5) and direct to homepage | Correct result show in testing video |
| Using top ten list table to evaluate movie | 0 stars  (ST=10,MaST=10,MiST=0 CRT=0,CR=0) | Database changed (CR=0, FCR=2) and direct to homepage | Correct result show in testing video |
| Using top ten list table to evaluate movie | 5 star  (ST=10,MaST=10,MiST=0 CRT=0,CR=0) | Database changed (CR=5, FCR=5) and direct to homepage | Correct result show in testing video |
| Update movie rate (edit database) | Title 10  CR=3 | Top ten chart changed FCR=3.8 | Correct result show in testing video |
| Update movie rate (edit database) | Title:10  CR=0  Title: 12 Monkeys (DTS)  CR=5 | Top ten list changed | Correct result show in testing video |

|  |  |  |  |
| --- | --- | --- | --- |
| Test case (In mobile 360X640) | Data inputted | Expect Result | Result |
| Using searched table to evaluate movie | 0 star (ST=11,MaST=11,MiST=0 CRT=0,CR=0) | Database changed (CR=0, FCR=2) and direct to homepage | Correct result show in testing video |
| Using searched table to evaluate movie | 5 star  (ST=11,MaST=11,MiST=0 CRT=0,CR=0) | Database changed (CR=5, FCR=5) and direct to homepage | Correct result show in testing video |
| Using top ten list table to evaluate movie | 0 stars  (ST=10,MaST=10,MiST=0 CRT=0,CR=0) | Database changed (CR=0, FCR=2) and direct to homepage | Correct result show in testing video |
| Using top ten list table to evaluate movie | 5 star  (ST=10,MaST=10,MiST=0 CRT=0,CR=0) | Database changed (CR=5, FCR=5) and direct to homepage | Correct result show in testing video |
| Update movie rate (edit database) | Title 10  CR=3 | Top ten chart changed FCR=3.8 | Correct result show in testing video |
| Update movie rate (edit database) | Title:10  CR=0  Title: 12 Monkeys (DTS)  CR=5 | Top ten list changed | Correct result show in testing video |

|  |  |  |  |
| --- | --- | --- | --- |
| Test case (In ipad 1024 X 768) | Data inputted | Expect Result | Result |
| Using searched table to evaluate movie | 0 star (ST=11,MaST=11,MiST=0 CRT=0,CR=0) | Database changed (CR=0, FCR=2) and direct to homepage | Correct result show in testing video |
| Using searched table to evaluate movie | 5 star  (ST=11,MaST=11,MiST=0 CRT=0,CR=0) | Database changed (CR=5, FCR=5) and direct to homepage | Correct result show in testing video |
| Using top ten list table to evaluate movie | 0 stars  (ST=10,MaST=10,MiST=0 CRT=0,CR=0) | Database changed (CR=0, FCR=2) and direct to homepage | Correct result show in testing video |
| Using top ten list table to evaluate movie | 5 star  (ST=10,MaST=10,MiST=0 CRT=0,CR=0) | Database changed (CR=5, FCR=5) and direct to homepage | Correct result show in testing video |
| Update movie rate (edit database) | Title 10  CR=3 | Top ten chart changed FCR=3.8 | Correct result show in testing video |
| Update movie rate (edit database) | Title:10  CR=0  Title: 12 Monkeys (DTS)  CR=5 | Top ten list changed | Correct result show in testing video |
| Login as Staff | Username = staff, Password = staff | Logged in as staff | Result as expected. Evidence can be found on testing video |
| Login as Admin | Username = admin, Password = admin | Logged in as admin | Result as expected. Evidence can be found on testing video |
| Login as wrong username and password | Username = test, Password = test | Show alert message | Result as expected. Evidence can be found on testing video |

### Report

The test met client’s requirements:

* Every logic path are tested, all output are as expected.
* In the applictaoin, there are no significant bug.
* Every path had noticed users when the operating of users is wrong.
* The layout is responsive

# Master Documents Tree/Links

**Folder:**

Sprint One Master Document – [folder-link](Master_Document_Sprint_One.docx)

Sprint Two Master Document – [folder-link](Master_Document_Sprint_Two.docx)

Handover Master Document – [folder-link](Master_Documentation_Handover.docx)

**Github:**

Sprint One Master Document – [hyperlink](https://github.com/SMT-Diploma-of-Software-Development/Movie_Theater_Database/blob/final_version/Master_Document_Sprint_One.docx)

Sprint Two Master Document – [hyperlink](https://github.com/SMT-Diploma-of-Software-Development/Movie_Theater_Database/blob/final_version/Master_Document_Sprint_Two.docx)

Handover Master Document – [hyperlink](https://github.com/SMT-Diploma-of-Software-Development/Movie_Theater_Database/blob/final_version/Master_Documentation_Handover.docx)