Software Development Testing Plan

Develop a test plan for the project

CHANGELOG

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Date of change | Change by | Outline |
| 1.0 | 5/11/2020 | Daniel Ewen | Test Plan Creation |
| 1.1 | 16/11/2020 | Lisa McKenna | Update Test Plan for Sprint 2 |
| 1.2 | 23/11/2020 | Swen Lee | Update Test Plan for Sprint 3 |
| 1.3 | 3/12/2020 | Daniel Ewen | Updated plan and added screenshots to testing plans. |

# Introduction

## Scope

### In Scope

Features to be tested:

* Application design is responsive
* User Interface
* Interface responds in real-time
* Database migrates correctly to the new design.

## Out of Scope

Features that won’t be tested:

* Movie database

## Quality Objective

Our team’s objective is to deliver a working prototype to the client with the following qualities:

* AUT[\*](#_Terms_/_Acronyms) must conform to the client’s requirements for each sprint.
* AUT must meet the client’s quality standards.
* Bugs/defects are tested and fixed before the application is deployed for the client.
* The AUT meets coding and commenting standards defined by each coding language standard.

## Roles and Responsibilities

|  |  |
| --- | --- |
| Role | Team Member/s |
| Developers | Research, design, implement and manage software programs |
| Test Manager | Organise and control the testing process to deliver a high-quality software |
| QA analyst | Test software on other computers to ensure they are functioning accurately |
| Business Analyst | Connect IT and business using data analytics and determine client requirements |
| Bug Triage | Evaluate, prioritise and assign resolution defects |

# Test Methodology

## Overview

Our team will utilize the RAD[\*](#_Terms_/_Acronyms) test methodology throughout this project. Each sprint will seek to develop, test, and deliver a fully functioning prototype to the specification of the client.

## Test Levels

Our testing team will implement the following test types for this project:

* Exploratory Testing
* Functional Testing
* Accessibility Testing
* Compatibility Testing
* Integration Testing
* System testing
* User Acceptance Testing (UAT)

## Bug Triage

Our process for bug/defect triage is as follows:

* **Bug discovered:** Bug report is added to the bug list. For this our team will use the issue tracker built into GitHub, detailing the bug.
* **Investigate:** Developers read the report and try to replicate the bug. If replication occurs, the bug can then be prioritized.
* **Bug added to backlog:** Bug is awaiting resolution during sprint.
* **Resolved:** Bug is fixed during development and the issue is closed.

## Suspension Criteria & Resumption Requirements

During testing, should any test return more than a 30% fail the following should occur:

* Testing ceases
* Test case / module being tested investigated for design flaws
* If bugs are discovered, add them to the triage queue.
* Once the bug has resolved in the triage queue, run the test again.

## Test Completeness

Testing will be complete when:

* The AUT has been deemed to have 100% test coverage,
* All designed test cases return with a pass,
* All current bugs/issues have been resolved,
* The client is satisfied with the current condition of the AUT and has signed off on its deployment.

# Test Deliverables

|  |
| --- |
| Deliverables |
| Test Plan |
| Test Cases with validation |
| Requirements |
| Analysis Report |
| Bug Reports |
| Client Sign-off |

# Resource & Environment Needs

## Testing Tools

* **Chromium developer tools:** Exploratory testing and during development.
* **PHP CodeSniffer:** Checks the PHP code for syntax and standards errors.

## Test Environment

### Hardware Environment

A modern development workstation with the following minimum specifications:

|  |
| --- |
| Part |
| Intel CPU 4-cores or more with hyperthreading |
| 16GB DDR4 RAM 2600MHz |
|  |

### Required Software

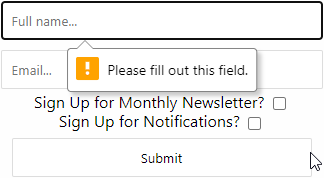
|  |  |
| --- | --- |
| Software | |
| Xampp | Local webserver and database hosting |
| Visual Studio Code | Code and application development |
| GitHub Desktop | Source Control |

# Test Table

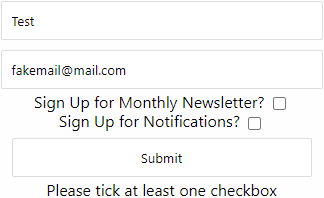
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Expected Results | Actual Results | Pass/Fail |
| Sign-Up Form | | | | |
| Precondition: Enter full name | | | | |
| 1 | User enters Full name with any combination of letters, symbols or numbers | Name validates on form submission as the input is not empty | Name validates on form submission as the input is not empty | Pass |
| 2 | User does not enter anything into full name field. | Name fails to validate and form submission is rejected | Name fails to validate and form submission is rejected | Pass |
| Precondition: Enter email address | | | | |
| 3 | User enters email with correct format | Email validates during form submission | Email validates during form submission | Pass |
| 4 | User enters email with incorrect formatting | Email does not validate at form submission and prints error. | Email does not validate at form submission and prints error. | Pass |
| Precondition: Communication option selection (monthly newsletter and/or breaking news flash notification as they occur) | | | | |
| 5 | One or both checkboxes are checked | If rest of form validates, prints success and preferences will be added to database | If rest of form validates, prints success and preferences will be added to database | Pass |
| 6 | Neither box is checked | Form fails to validate and prints error | Form fails to validate and prints error | Pass |

**Ref.**

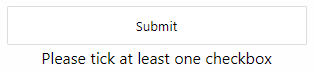
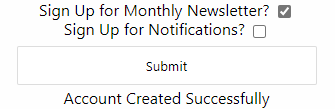
1&2.

3&4.

5&6.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Update Form  Functions identically to the signup form but if it validates and the account exists it will change their preferences. | | | | |
| Precondition: User enters details with new preferences | | | | |
| 7 | User enters valid account details with different preferences to current ones. | Success message printed.  User’s details are updated | Success message printed.  User’s details are updated | Pass |
| 8 | User enters invalid account | Failure message printed | Failure message printed | Pass |

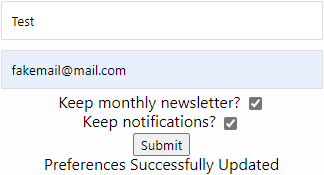
Ref:

7.

Original preferences:

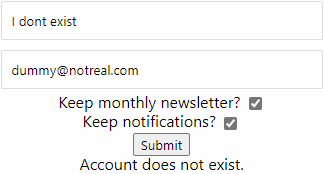


New Preferences:





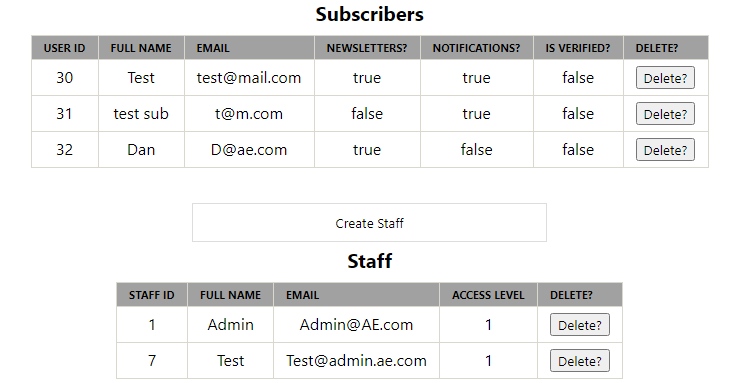
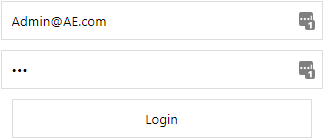
8.



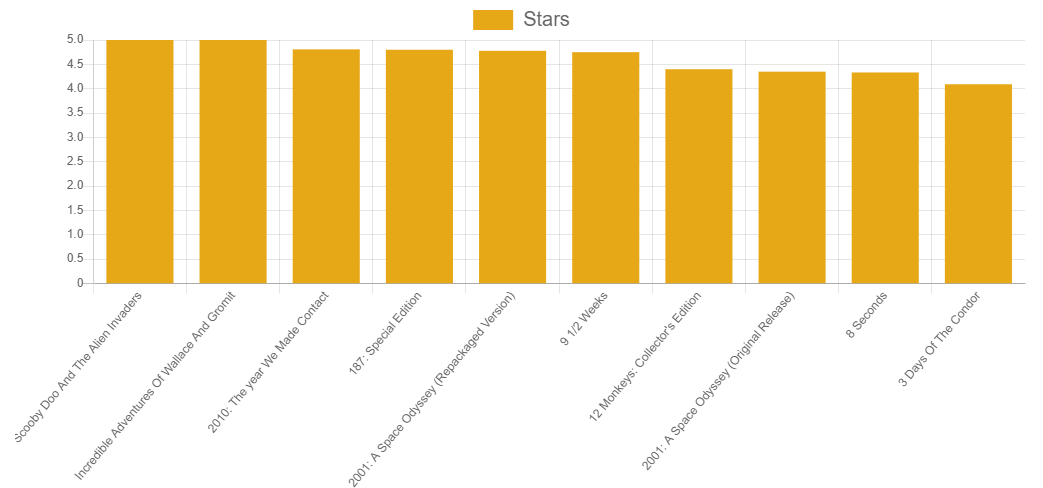
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Expected Results | Actual Results | Pass/Fail |
| Sprint 3 Update | | | | |
| Precondition: Administrator enters username and password | | | | |
| 9 | Login as administrator with correct credentials | Able to access and edit database | Able to access and edit database | Pass |
| Precondition: N/A | | | | |
| 10 | Click on ratings page | Display the top 10 rated movies | Display the top 10 rated movies | Pass |
| 11 | Remain at the top 10 ratings page | Refreshes the page automatically every few minutes | Refreshes the page automatically every few minutes | Pass |
| 12 | Click star to rate movie on movie page. | Rating increases / decreases based on rating direction. | Rating number changes expectedly | Pass |

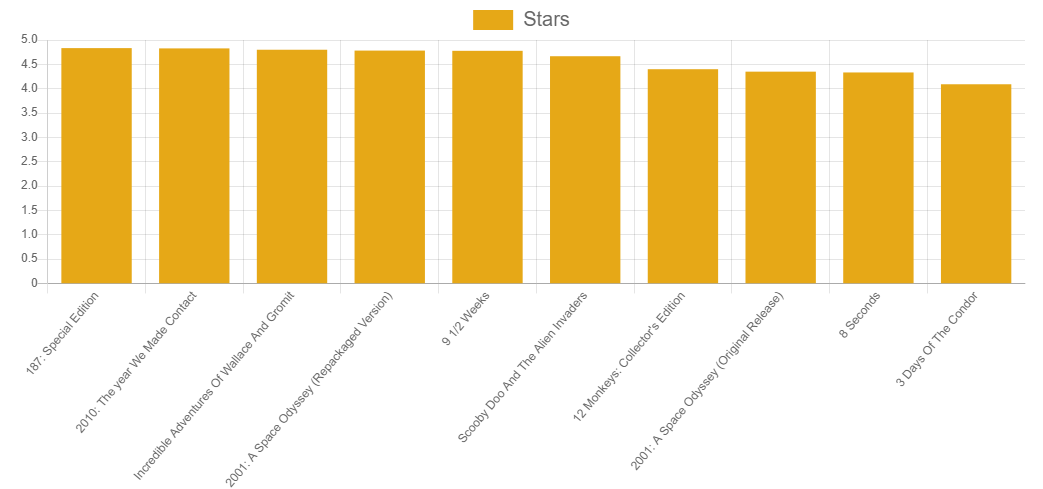
Reference:

9.a/b



10/11.





12.a/b



# Exit Criteria

## Exit Criteria

Exit criteria defines when to stop the testing process, for example, when a set of tests has achieved its objective.

Exit criteria may involve:

* Estimates of defect density or reliability measures
* Cost
* Residual risks (e.g. unfixed bugs)
* Thoroughness measures (e.g. code functionality)

(Mostafa, 2018)

# Conclusion

With Agile development, project development is split into sprints or iterations. At the end of every sprint, the product is tested and client input is obtained to ensure the product meets the requirements. If there are errors or bugs detected in the sprint, that sprint is repeated until it meets the specified requirements. With Agile, client satisfaction is prioritised and therefore the quality of the product is ensured.

# Terms / Acronyms

|  |  |
| --- | --- |
| Term / Acronym | Definition |
| RAD | Rapid Application Development |
| AUT | Application Under Test |