MovieUniverse

Software Test Plan

Version 1.0

08.03.2023

Roaya Heib

QA Tester

Prepared for

Samana Project

Winter 2023

**Table of Contents**

[1 INTRODUCTION 3](#_Toc21693)

[2 SCOPE 3](#_Toc21694)

[3 QUALITY OBJECTIVES 4](#_Toc21695)

[3.1 Primary Objectives 4](#_Toc21696)

[3.2 Secondary Objectives 4](#_Toc21697)

[4 TEST APPROACH 4](#_Toc21698)

[5 ROLES AND RESPONSIBILITIES 5](#_Toc21700)

[6 ENTRY AND EXIT CRITERIA 5](#_Toc21701)

[6.1 Entry Criteria 5](#_Toc21702)

[6.2 Exit Criteria 5](#_Toc21703)

[7 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS 5](#_Toc21704)

[7.1 Suspension criteria 5](#_Toc21705)

[7.2 Resumption criteria 5](#_Toc21706)

[8 TEST STRATEGY 6](#_Toc21707)

[8.1 QA role in test process 6](#_Toc21708)

[8.2 Bug life cycle: 7](#_Toc21709)

[8.3 Testing types 7](#_Toc21710)

[8.4 Bug Severity and Priority Definition 8](#_Toc21711)

[Severity List 8](#_Toc21712)

[Priority List 9](#_Toc21713)

[9 RESOURCE AND ENVIRONMENT NEEDS 9](#_Toc21714)

[9.1 Testing Tools 9](#_Toc21715)

[9.3 Test Environment 9](#_Toc21717)

[TERMS/ACRONYMS 10](#_Toc21720)

**Project “MovieUniverse”**

Document Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Version | Description | Author | Reviewer | Approver |
| 08.03.2023 | A test | Test plan was created | Roaya Heib |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# INTRODUCTION

The Movie Management System is an application that enables users to manage a database of movies and reviews. The system is built using:

* Flask web framework,
* SQL Alchemy database toolkit.
* Flask-Migrate extension for database migrations.

This document describes approaches and methodologies that will apply to the unit, integration and system testing of the website "<http://127.0.0.1:5000/>”. It includes the objectives, test responsibilities, entry and exit criteria, scope, schedule major milestones, entry and exit criteria, and approach. This document identifies what the test deliverables will be, and what is deemed in and out of scope.

# SCOPE

The document mainly validates the efficiency of the website, by testing its functionality.

**2.1 Functions to be tested**

* Add new movie.
* Add review.
* Modify movie data.
* Delete movie from the database.
* Add a movie to the watchlist.
* The file extension when adding an image file.
* Movies covers are displayed on the home page.
* Relevant movie covers are displayed on the watchlist page.
* The search bar.

# QUALITY OBJECTIVES

## Primary Objectives

**3.1.1 General requirements**

* Core i5 or later.
* Flask web framework,
* SQLite database.
* SQL Alchemy database toolkit.
* Flask-Migrate extension for database migrations.
* A computer or server capable of running Python 3.6 or later.
* Sufficient storage to host the database and image files.

**3.1.2 functional requirements**

The system does not require user registration, and there is no login functionality. Any user can access the system and perform the following functions:

* Add movie.
* Delete movie.
* View movie data.
* Add review.

**3.1.3 Non-functional requirements**

* Performance.
* Reliability.
* Security.

## 3.2 Secondary Objectives

The secondary objectives of testing will be to: identify and expose all issues and associated risks, and ensure that all issues are addressed in an appropriate matter before release. As an objective, this requires careful and methodical testing of the application to first ensure all areas of the system are scrutinized and, consequently, all issues (bugs) found are dealt with appropriately.

# TEST APPROACH

The approach that is used is Analytical, therefore, by the requirements-based strategy, Test cases will be created during exploratory testing. All test types are determined in Test Strategy.

Also, experience-based testing and error guessing are used.

# ROLES AND RESPONSIBILITIES

|  |  |  |
| --- | --- | --- |
| Role | Staff Member | Responsibilities |
| QA tester |  | 1. Understand requirements.  2. Writing and executing Test cases.  3. Reviewing Test cases.  4. Defect reporting and tracking  5. Retesting and regression testing. |

# ENTRY AND EXIT CRITERIA

## Entry Criteria

* All test hardware platforms must have been successfully installed, configured, and functioning properly.
* All the necessary documentation, design, and requirements information should be available that will allow testers to operate the system and judge the correct behavior.
* All the standard software tools including the testing tools must have been successfully installed and functioning properly.
* Proper test data is available.
* The test environment such as hardware, software, and system administration support should be ready.

## Exit Criteria

* A certain level of requirements coverage has been achieved.
* No high-priority or severe bugs are left outstanding.
* All high-risk areas have been fully tested, with only minor residual risks left outstanding.
* The schedule has been achieved.

**7. SUSPENSION CRITERIA AND RESUMPTION**

## 7.1 Suspension criteria

## The build contains many serious defects which seriously limit testing progress.

## Software/Hardware problems.

## 7.2 Resumption

Resumption will only occur when the problem(s) that caused the suspension have been resolved.

**8. TEST STRATEGY**

**8.1 Test approach**

The tests mainly target GUI testing; testing the functions of the application that are visible to the user and validating data Requirements Specification.

Understanding Requirements:

* + Requirement specifications.
  + Understanding of requirements will be done by the QA tester.

Preparing Test Cases:

* QA tester will be preparing test cases based on exploratory testing. This will cover all scenarios for requirements.
* Any comments or suggestions on test cases and test coverage will be provided by the reviewer’s respective Author of the Test Case.
* Suggestions or improvements will be reworked by the author and will be sent for approval.
* Re-worked improvements will be reviewed and approved by the reviewer.

Executing Test Cases:

* Test cases will be executed by respective QA tester on the test application based on designed scenarios, test cases, and Test data.
* Test result (Actual Result, Pass/Fail) will be updated in the test case document Defect Logging and Reporting.
* QA will be logging the defect/bugs in a Word document, found during the execution of test cases. After this, QA will inform the respective developer about the defects/bugs.

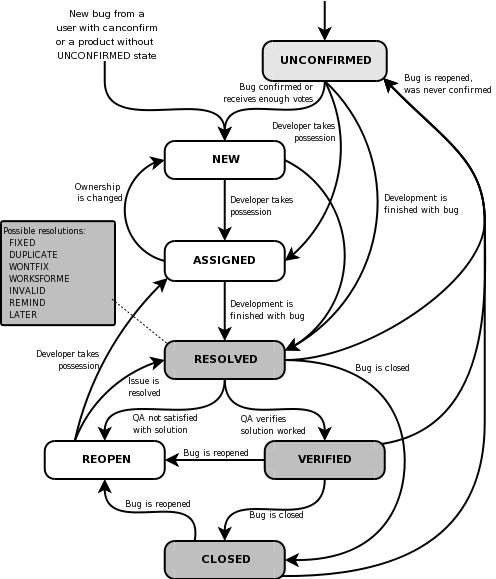
Retesting and Regression Testing:

Retesting for fixed bugs will be done by the respective QA once it is resolved by the respective developer and bug/defect status will be updated accordingly. In certain cases, regression testing will be done if required.

## 8.2 Bug life cycle

All the issues found while testing will be logged into a Word document.

The bug life cycle for this project is as follows:



## 8.3 Testing types

GUI Testing:

Graphic User Interface Testing (GUI) testing is the process of ensuring the proper functionality of the graphical user interface (GUI) for a specific application. This involves making sure it behaves by its requirements and works as expected across the range of supported platforms and devices.

Smoke testing:

Smoke testing, also known as "Build Verification Testing" or "Build Acceptance Testing", is a type of software testing that is used to quickly verify that a new build or release of a system is performing correctly and that the major functionalities are working as expected. It is a type of superficial testing that focuses on the crucial function of a system and is usually done immediately after the build is ready. The goal of smoke testing is to quickly identify if the build is stable enough to proceed with further testing or if it needs to be fixed before extensive testing is done. Smoke testing is also sometimes referred to as "Sanity Testing" as it indicates the sanity of the build.

Security testing:

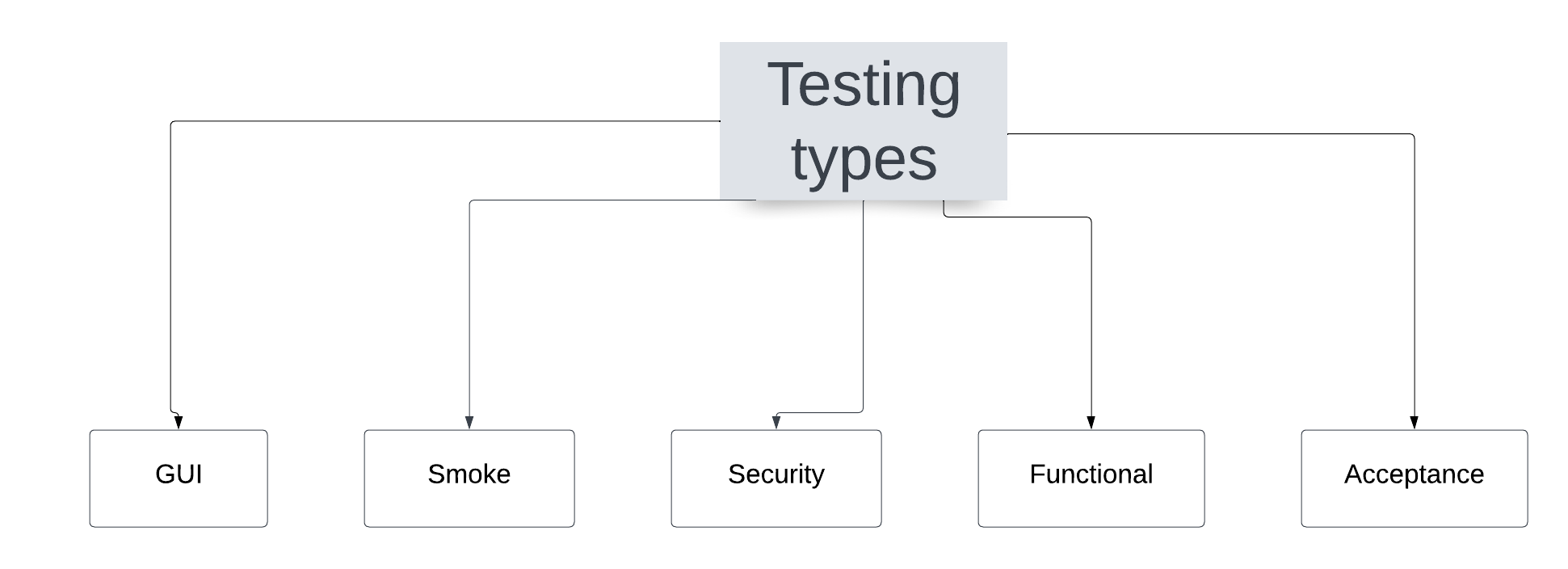
The main goal of **Security Testing** is to identify the threats in the system and measure its potential vulnerabilities so that the threats can be encountered, and the system does not stop functioning or can not be exploited. It also helps in detecting all possible security risks in the system and helps developers to fix the problems through coding.

Functional testing:

Functional testing is a type of testing that seeks to establish whether each application feature works as per the software requirements. Each function is compared to the corresponding requirement to ascertain whether its output is consistent with the end user's expectations.

Acceptance testing:

Acceptance testing is a quality assurance (QA) process that determines how an application meets end users' approval. Depending on the organization, acceptance testing might take the form of beta testing, application testing, field testing, or end-user testing.



## Bug Severity and Priority Definition

Bug Severity and Priority fields are both very important for categorizing bugs and prioritizing if, and when the bugs will be fixed. The bug Severity and Priority levels will be defined as outlined in the following tables below. Testing will assign a severity level to all bugs.

## 8.4.1 Severity List

|  |  |  |
| --- | --- | --- |
| **Severity ID** | **Severity** | **Severity Description** |
| 1 | Critical | The bug crashes the system. |
| 2 | High | It affects the website’s functionality.  The failed function is unusable but there exists an acceptable alternative method to achieve the required results. |
| 3 | Medium | It prevents other parts of the website from being tested, yet other areas can independently be tested. |
| 4 | Minor | Error messages. |

## 8.4.2 Priority List

|  |  |  |
| --- | --- | --- |
| **Priority** | **Priority Level** | **Priority Description** |
| 1 | High Priority | This bug must be fixed immediately. |
| 2 | Medium Priority | These are important problems that should be fixed as soon as possible. |
| 3 | Low Priority | The defect is an irritant that should be repaired, but a repair can be deferred until after more serious defects have been fixed. |

# 9. RESOURCE AND ENVIRONMENTAL NEEDS

# 9.1 Testing Tools

|  |  |  |
| --- | --- | --- |
| Process |  | Tool |
| Test case creation | Microsoft Excel |  |
| Test case tracking | Microsoft Excel |  |
| Test case execution | Selenium |  |
| Database toolkit | SQL Alchemy |  |
| Web Framework | Flask |  |
| Database | SQLite |  |
| Project structure | PyCharm |  |

## 9.2 Test Environment

* Browsers:
* Edge (latest)
* Chrome (latest)
* Devices:
* Computer Core i5 or later, RAM 8GB.
* Support levels:
* Windows 11, Version 22H2.

# 10. TERMS/ACRONYMS

|  |  |
| --- | --- |
| **TERM/ACRONYM** | **DEFINITION** |
| SQL | Structure Query Language. |
| GUI | Graphical user interface |
| QA | Quality Assurance |