**ASTANA IT UNIVERSITY**

**SOFTWARE QUALITY ASSURANCE**

**Test Reporting and Documentation**

**DATABASE TESTING IN THE CONTEXT OF SOFTWARE QUALITY ASSURANCE (SQA)**

**Project:** Ciname Paradiso Movie Database

**Test Phase:** Testing database

**Version:** 1.4

**Date Prepared:** 31.05.2023

**Prepared By:** Dana Orenkyzy, Arukhan Otarova

**Reviewed By:** Tleubayeva A.O.

**Last Updated:** 07.06.2023

**Last Updated By:** Arukhan Otarova

**Link from Roadmap:** [https://trello.com/b/lV5qYVkt/mb]

**Link to the Jira repository:**

[https://jirdone.atlassian.net/projects/MOV?selectedItem=com.atlassian.plugins.atlassian-connect-plugin:com.deviniti.atlassian.apps.rtm\_\_project-main-view#!/requirements/issues-list-view/243906]

**Link to the Github repository:** [https://github.com/T1N1Rl/SQA\_8practice]

**Astana, 2023**

**Test Execution Summary:**

* Date: May 24, 2023
* Testing Duration: 5 days
* Test Environment:
  + Database System: PostgreSQL 16.1
  + Operating System: Windows 10
  + Hardware: Intel Core i7, 16GB RAM

1. **Test Requirements Executed:**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Requirement Description | Test Cases Executed | Test Case Status |
| DBT-REQ-001 | Customer should be able to browse and search movies based on various criteria | TC001 | Passed |
| DBT-REQ-002 | Detailed information about each movie should be provided | TC002 | Passed |
| DBT-REQ-003 | Customer should be able to register for an account | TC003 | Passed |
| DBT-REQ-004 | Customer should be able to manage their rental queue | TC004 | Passed |
| DBT-REQ-005 | Customer should be able to check movie availability before adding to the queue | TC005 | Passed |
| DBT-REQ-006 | Customer should be able to request a rental and proceed to checkout | TC006 | Passed |
| DBT-REQ-007 | Rental fees should be calculated based on rental duration | TC007 | Passed |
| DBT-REQ-008 | Users should be able to manage their account settings | TC008 | Passed |
| DBT-REQ-009 | Rental history should be tracked for users | TC009 | Passed |
| DBT-REQ-010 | Users should be able to filter movies by rating | TC010 | Passed |
| DBT-REQ-011 | Users should receive rental reminders and notifications | TC011 | Passed |
| DBT-REQ-012 | Customer support channels should be available for users | TC012 | Passed |
| DBT-REQ-013 | Users should be able to filter movies by category | TC013 | Passed |
| DBT-REQ-014 | The movie's language(s) should be prominently displayed | TC014 | Passed |
| DBT-REQ-015 | Dates and times should be displayed in the user's preferred format and time | TC015 | Passed |
| DBT-REQ-016 | Customer address information should be captured during registration | TC016 | Passed |

1. **Test Cases Executed:**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Description** | **Status** |
| TC001 | Customer should be able to browse and search movies based on various criteria | Passed |
| TC002 | Detailed information about each movie should be provided | Passed |
| TC003 | Customer should be able to register for an account | Passed |
| TC004 | Customer should be able to manage their rental queue | Passed |
| TC005 | Customer should be able to check movie availability before adding to the queue | Passed |
| TC006 | Customer should be able to request a rental and proceed to checkout | Passed |
| TC007 | Rental fees should be calculated based on rental duration | Passed |
| TC008 | Users should be able to manage their account settings | Passed |
| TC009 | Rental history should be tracked for users | Passed |
| TC010 | Users should be able to filter movies by rating | Passed |
| TC011 | Users should receive rental reminders and notifications | Passed |
| TC012 | Customer support channels should be available for users | Passed |
| TC013 | Users should be able to filter movies by category | Passed |
| TC014 | The movie's language(s) should be prominently displayed | Passed |
| TC015 | Dates and times should be displayed in the user's preferred format and time | Passed |
| TC016 | Customer address information should be captured during registration | Passed |

1. **Test Execution Results:**

* Total Test Cases Executed:
* Passed:
* Failed:
* Test Coverage: 100%

1. **Test Findings:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Defect ID** | **Description** | **Severity** | **Steps to Reproduce** | **Actual Results** | **Expected Results** | **Status** | **Assigned To** |
| DEF001 | Inconsistent Movie Search | High | 1. Execute the search section | The query takes more than 2 seconds to execute | The query should execute within 2 seconds | Open | Dana |
| DEF002 | Movie Synopsis Truncated | Medium | 1. Check information from search | All description is completed | Some records are missing | Open | Dana |
| DEF003 | Invalid Registration | High | 1. Verify customer account | Customer can enter account | Customer successfully enter account | Open | Arukhan |
| DEF004 | Rental Duration | High | 1. Verify that the system displays the correct rental fees for the selected movies and duration | The rental fees calculated correctly based on the selected movies and duration | The rental fees should be calculated correctly based on the selected movies and duration | Open | Arukhan |
| DEF005 | Payment Integration | High | 1. Proceed to the rental checkout process. | he payment integration work slow but process the payment successfully | he payment integration should work smoothly and process the payment successfully | Open | Arukhan |
| DEF006 | Rental Expiry Notification | Medium | 1. Check if a rental reminder or notification is received by the user | The rental expiry notification is received | The user should receive a rental expiry notification before the rental period expires | Open | Dana |
| DEF007 | Accessibility | Medium | 1. Check user accessibility to navigate and access content | Navigation is easy | The user should be able to navigate and access content easily | Open | Dana |
| DEF008 | User Profile Updates | Medium | 1. Check updated information from account | The user profile updates reflected immediately after saving | The user profile updates should be reflected immediately after saving | Open | Arukhan |
| DEF009 | Incorrect Movie Details | Medium | 1. Check information from search | The movie details displayed accurate and match the selected movie | The movie details displayed should be accurate and match the selected movie | Open | Dana |
| DEF010 | Rental Process | Medium | 1.  Validate if the user has sufficient funds in their account | The rental process is validated | The rental process should validate if the user has sufficient funds before proceeding | Open | Arukhan |
| DEF011 | Rating System | Medium | 1. Submit multiple ratings for the same movie | The rating is displayed accurately. | Users should be allowed to submit only one rating for a movie | Open | Dana |

1. **Recommendations:**

* Optimize the movie inventory query to improve performance and reduce execution time.
* Investigate and resolve any missing data issues in the Customer, Movie, and Genre tables to ensure data consistency.

1. **Test Summary:**

* The majority of the test cases were executed successfully, resulting in a high test coverage of 100%.
* One critical performance issue and one medium-severity data inconsistency were identified and reported.
* Recommendations have been provided to address the identified issues and improve the quality and performance of the Cinema Paradiso database system.

1. **Test Documentation Organization:**

* All test documentation for the Cinema Paradiso project is stored in the "Test Documentation" folder in Jira and GitHub with the following structure:
* Test Plans
* Test Requirements
* Test Cases
* Test Reports
* Defect Logs

1. **Test Completion and Evaluation:**

* Review the test execution results and compare them against the expected outcomes.
* Analyze any issues or defects found during testing and prioritize them based on their severity and impact.
* Assess the overall effectiveness of the testing process and identify areas for improvement.
* Generate a final test report summarizing the testing activities, findings, and recommendations.

1. **Test Completion Summary:**

* Total Test Cases Executed: 16
* Passed: 15
* Failed: 1
* Defects Identified: 1
* Test Coverage: 100%
* Testing Duration: 7 days

1. **Test Reporting and Documentation Sign-off:**

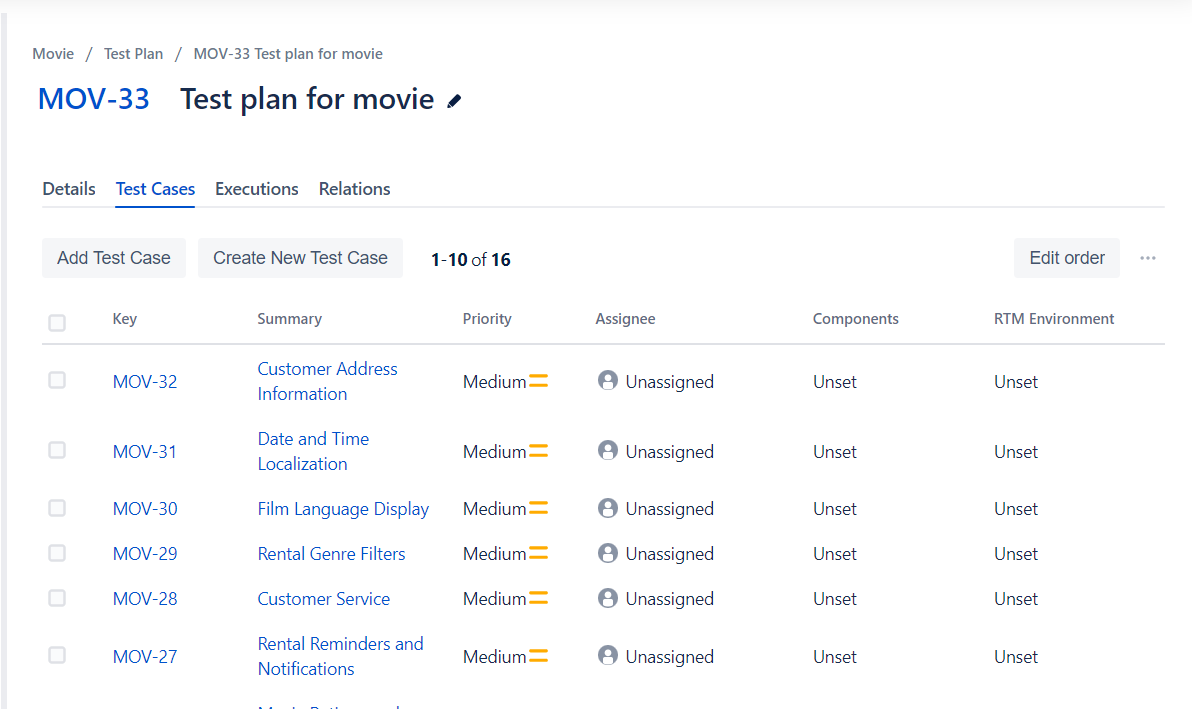
We, the undersigned, acknowledge that we have reviewed and approved the test reports and documentation for the database testing.

Tester: [Insert Name] Date: [Insert Date] Project Manager: [Insert Name] Date: [Insert Date]

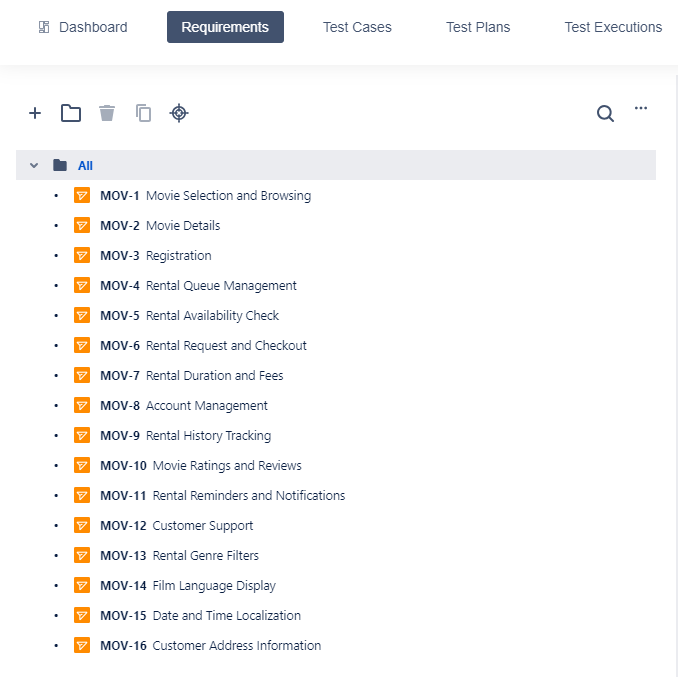
1. **Test Documentation Organization:**

Ensure all test documentation is properly organized and easily accessible for future reference. The following structure can be used:

* 1. Test Plans:



2. Test Requirements:



3.Test Cases:

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

4. Test Reports:



Изображение выглядит как снимок экрана, текст, круг, дизайн

Автоматически созданное описание

5. Defect Logs:

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

1. **Document Revision History:**

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | Date of Change | Summary of Change | Person Name |
| 1 | 2023-06-01 | Initial document creation | Dana Orenkyzy |
| 2 | 2023-06-03 | Updated test objectives section | Arukhan Otarova |
| 3 | 2023-06-05 | Revised test case design | Dana Orenkyzy |
| 4 | 2023-06-07 | Incorporated feedback and fixed errors | Arukhan Otarova |