*Movie Rental Application*

Test Summary Report

Version *<1.0>*

*12/03/2018*

*Authors:*

*Joseph Antonacci*

*Lu Lin*

*Tracy Zeigler*

VERSION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 1.0 | *Joseph Antonacci*  *Lu Lin*  *Tracy Zeigler* | *11/03/2018* |  | *11/03/2018* | *Initial completion* |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**UP Template Version:** 12/03/2018

TABLE OF CONTENTS

[1 Introduction 4](#_Toc216837569)

[1.1 Purpose 4](#_Toc216837570)

[1.2 Introduction of Application 4](#_Toc216837570)

[2 Proposal 4](#_Toc216837571)

[2.1 *Objective*  4](#_Toc216837572)

[2.2 *Tasks*  4](#_Toc216837573)

[3 Testing hardware/software requirements 4](#_Toc216837575)

[3.1 *Black Box Testing PC* 4](#_Toc216837572)

[3.2 *White Box Testing PC* 4](#_Toc216837573)

[4 Test Summary 5](#_Toc216837571)

[4.1 *Unit Testing (White Box Testing)* 5](#_Toc216837572)

[4.2 *Intergration (Black Box Testing)* 5](#_Toc216837573)

[4.3 *Volume Testing (Non-Functional Testing)* 6](#_Toc216837574)

[4.4 *Boundry and Partition Testing (Non-Functional Testing)* 6](#_Toc216837574)

[5 Test Assessment 6](#_Toc216837576)

[6 Test Results 7](#_Toc216837576)

[7 Conclusion 8](#_Toc216837577)

[Appendix A: Test Summary Report Approval 9](#_Toc216837578)

[APPENDIX B: ATTACHMENTS 10](#_Toc216837579)

[APPENDIX C: REFERENCES 10](#_Toc216837579)

# 

# Introduction

## Purpose

This Movie Rental Application Test Report provides a summary of the results of test performed as outlined within this document.

## INTRODUCTION OF APPLICATION

The name of application we are testing is Movie Rental, which is developed using C#, with a built-in Visual Studio software environment and MySQL, which is basically organized in terms of functions like customer, movie, and employee. Movie Rental Application offers the name of movie and its rental status to customers; the customer data are available in customer ID, first name, last name, address and e-mail; the administrator is able to add, delete, and update information related to movies and customers.

# Proposal

## Objective

The objective of the tests is to verify the functionality of Movie Rental works as per the client’s requirements. The following will be provided in the report:

* Test Summary Report
* A description of each test case conducted
* The test case results
* Test Incident Report

## tasks

The following will be included in the project:

* Unit Testing (White Box Testing)
* Integration Testing (Black Box Testing)
* Volume Testing (Non-Functional Testing)
* Boundary and Partition Test cases (Non-Functional Testing)

# testing Hardware/software Requirements

## White Box Testing PC

* PC running Windows 7 or higher
* Intel Core i3 or better
* Minimum 2 GB of Ram
* .Net Framework 4.5.2 or greater
* Visual Studio 2017
* SQL Expressions 2017 or better

## Black box testing PC

* PC running Windows 7 or higher
* Intel Core i3 or better
* Minimum 2 GB of Ram
* .Net Framework 4.5.2 or greater
* SQL Server Express 2017

# Test Summary

For this project Unit, Integration, and Volume testing was completed. It tests the functionality of application to ensure they meet the clients requirements.

**Project Name**:Movie Rental Project

**System Name**: Windows OS 7/10

**Version Number**: 1.0

## Unit Testing (White box testing)

For unit testing we tested the functionality of the objects that comprised of the application GUI. This will be back-end testing and black box testing the functionality of the GUI objects. We just made sure the buttons functioned, the text fields were able to accept text, and the databases updated appropriately.

**Test Owner**: Tracy Zeigler

**Test Date**: 10/13/2018

**Test Results**: From the testing completed, everything worked as expected. I did not encounter any issues with the functionality of the application. Buttons worked, text fields accepted the appropriate text, and the databases updated as expected.

**Additional Comments**: There could be some additional cleanup on the text fields, such as having a zip-code check more accurately for valid zip-codes, have date fields do an auto format of the date entered.

## Intergration Testing (Black box testing)

For Integration testing we first generated an excel file with test cases to be blackbox tested. Once this was complete, the program was actually installed on a testing machine and the test cases run.

**Test Owner**: Joseph Antonacci

**Test Date**: 11/10/2018 – 11/30/2018

**Test Results:** Each test in the test plan was executed without access to the code, as if by a naive user. The majority of tests passed, but some failed. Failure tended to result from items that required certain types of input validation. As mentioned, buttons, navigation, and major program functions worked properly.

**Additional Comments**: Given the general types of failed tests, Equivalence Partitions, and Boundary Conditions were created. Though we did not run these tests themselves, they were generated with the idea that they could be used in the event that further changes to certain functions were made. I.e. the next round of testing might include tests more specific to these (for example, zip code checks).

## Volume Testing (non-Functional)

We tested 3 files, one for customer data, one for movie data, and one for rental data. The movie and rental data each had 50,000 entries, and the customer had 150,000 entries.

**Test Owner**: Lu Lin

**Test Date**: 11/10/2018 – 11/30/2018

**Test Results**: All test cases passed, but it did take some time for the files to load.

**Additional Comments**: The files loaded, but the application was not responsive while the data was being uploaded to the database.

## Boundary and Equivalence partition Testing (Non-Functional Testing)

**Test Owner**: Joseph Antonacci

**Test Date**: 11/10/2018 – 11/30/2018

**Test Results**: This is just the test cases we believed would be appropriate.

**Additional Comments**: Please note: Some categories for validity were chosen based on known valid sets. Rather than list these, it seems in the interest of time, effort, and expense to utilize already existing tools, where possible, to handle checking certain types of partition conditions. For example, valid email addresses. These are each marked with a \*

# Test Assessment

The following is a summary of the test case results obtained for the testing efforts.

| Summary Assessment | Total Number of Test Cases | % of Total Planned | Comments |
| --- | --- | --- | --- |
| Unit Testing | 23 | 100% | No issues found |
| Unit Test Cases Run | 23 | 100% | No issues found |
| Unit Test Cases Reviewed | 23 | 100% | No issues found |
| Unit Test Cases Passed | 23 | 100% | No issues found |
| Test Cases Failed | 0 | 0% | No issues found |

| Summary Assessment | Total Number of Test Cases | % of Total Planned | Comments |
| --- | --- | --- | --- |
| Volume Testing | 3 | 100% | No issues found |
| Volume Test Cases Run | 3 | 100% | Loading was a little slow |
| Volume Test Cases Reviewed | 3 | 100% | No issues found |
| Volume Test Cases Passed | 3 | 100% | No issues found |
| Volume Cases Failed | 0 | 0% | No issues found |

| Summary Assessment | Total Number of Test Cases | % of Total Planned | Comments |
| --- | --- | --- | --- |
| Integration Testing | 58 | 100% | No issues found |
| Integration Test Cases Run | 58 | 100% | No issues found |
| Integration Test Cases Reviewed | 58 | 100% | No issues found |
| Integration Test Cases Passed | 43 | 74% | No issues found |
| Integration Cases Failed | 15 | 26% | Found a few issues with application crashing, error messages issues, and unable to process some functions |

# Test Results

The testing overall was a success. There are a total of 74 test cases completed, with only 15 of them failing. We look into the issues and fix them and set up a retest of the test cases. In addition, we discovered we need to put a more restraints on some fields such as verifying zip-codes are valid.

During Unit testing all test cases passed, without any issues. Databases updated properly, buttons functioned as expected. Everything the requirements requested is addressed and is coded for.

Volume testing, we were mainly focused to make sure if a large amount of data was being uploaded in a file, it would load without issues. After our testing, we did not find any major issues. It was slow to upload, but it did upload to the correct database without any loss of data or errors.

Integration testing, we focused on the functionality of the GUI. We tested to ensure warning messages popped up, the buttons functioned outside a development environment, data loaded appropriately, and the usability of the app. We had issues initially installing the app, but we were able to work through it and get it installed. There were 15 test cases that failed have been submitted on the incident report and back to the developer to look into the issues.

# Conclusion

The purpose of this project was to test the Movie Rental Application program to test the functionality of the application. We did white box testing, where we tested functionality while viewing the code. It was mainly unit testing where we tested to make sure the objects in the application function properly.

We then did some black box testing. Joseph, tested the application after installing it on a Windows PC. He tested a test plan written by Lu Lin. The test plan tested all the various fields on the application, making sure all buttons performed the appropriate tasks, loading files to the various database, and all fields took in the appropriate data. The majority of the test cases passed, but there were some test cases that were entered into the Incident Report Log. These will be looked at by Lu Lin, and if need be sent back to the development team to correct.

Lu Lin, did a non-functional volume test. She loaded large amounts of data to each of the databases to see if there would be any errors. No errors occurred and all data was loaded without any loss of data.

A boundary and equivalence partition was created to demonstrate the minimum and maximum values that would need to be tested to prevent from having an abundant number of test cases.

Testing overall was a great experience and we were able to use a lot of what we learned throughout the class. It was good there were some hiccups here and there because that made us think about how to handle the situations.

Appendix A: Test Summary Report Approval

The undersigned acknowledge they have reviewed the *Movie Rental Application* **Test Summary Report** and agree with the approach it presents. Changes to this **Test Summary Report** will be coordinated with and approved by the undersigned or their designated representatives.

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: | 12/03/2018 |
| Print Name: | Joseph Antonacci |  |  |
| Title: | Sr QA |  |  |
| Role: | Tester |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: | 12/03/2018 |
| Print Name: | Lu Lin |  |  |
| Title: | Project Manager |  |  |
| Role: | PM |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: | 12/03/2018 |
| Print Name: | Tracy Zeigler |  |  |
| Title: | Lead Developer |  |  |
| Role: | Developer/Tester |  |  |

APPENDIX B: ATTACHMENTS

The following table summarizes the documents referenced in this document.

|  |  |  |
| --- | --- | --- |
| **Document Name and Version** | **Description** | **Location** |
| *v 1.0* | *Unit Test Plan/Cases* | *Attached* |
| *v 1.0* | *Volume Test Plan/Cases* | *Attached* |
| *v 1.0* | *Integration Test Plan/ Cases* | *Attached* |
| *v 1.0* | *Test Incident Cases* | *Attached* |
| *v 1.0* | *Boundary and Equivalence Partition Case* | *Attached* |

**APPENDIX C: REFERENCES**

**1. Srinivasan Desikan and Gopalaswamy Ramesh, *Software Testing Principles and Practices*, Pearson 2013 edition.**

**2. Software Testing Tips: *101 Expert Tips, Tricks and Strategies for Better, Faster Testing and Leveraging Results for Success*, <https://stackify.com/software-testing-tips/>**

**3. System testing, <https://en.wikipedia.org/wiki/System_testing>**

**4.What is Integration Testing (Tutorial with Integration Testing Example) <https://www.softwaretestinghelp.com/what-is-integration-testing/>**

**5.Familiarization to Advantages of Ad Hoc Testing.** [**http://testorigen.com/familiarization-to-advantages-of-ad-hoc-testing/**](http://testorigen.com/familiarization-to-advantages-of-ad-hoc-testing/)

**6. MySQL Tutorial. Accessed 29.5.2014 <http://dev.mysql.com/doc/refman/5.0/en/tutorial.html>**

**7. Smita Joshi, Movie Rental Application, A Technical Overview, January 31, 2017. <http://www.softwaretree.com/v1/white-paper-pdf/JFlixProjectReport.pdf>**