Retail Inventory Management Application

TEST PLAN

Version <1.0>
11/08/2022

[Insert appropriate Disclaimer(s)]

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Ronald Hand	11/03/22	Svyatoslav Mudryy	11/08/22	Test Plan draft

UP Template Version: 12/31/07

TABLE OF CONTENTS

1 INTRODUCTION

The goal of the Retail Inventory Management Application(RIMA) is to provide a functional, user-friendly, low-maintenance management solution for a small store or retail business. The application will assist management with product updates, storage, inventory control, and report production. There will be multiple features of this product that will require component testing to verify product operability and functionality. Interoperability of system components are to be tested throughout the development lifecycle utilizing both automated and manual testing procedures.

2 OBJECTIVE

The objective of this test plan is to determine what testing strategies would be most applicable to be utilized throughout the software development cycle of the RIMA.

3 SCOPE

3.1 ITEMS TO BE TESTED

Item to Test	Test Description	Test Date	Responsibility
GUI	Test GUI for operation/functionality	11/16/22	Ronald Hand
Login/Logout	Junit test login/logout function	11/16/22	Ronald Hand
Logging	Test proper logging procedure	11/22/22	Ronald Hand
Database function	Test Database	11/22/22	Ronald Hand

3.2 ITEMS TO NOT TEST

Item to not Test	Test Description	Test Date	Responsibility
N/A at this time			

4 REFERENCES

All testing is to be derived from Open Web Application Security Project(OWASP) Testing Guide 4.0

5 TESTING OVERVIEW

5.1 TEST PROCESS

Testing should occur at all phases of the software development cycle to test for correct functionality, as well as, testing for possible security related issues that may occur. The testing director will maintain constant feedback cycles with developers and the project manager. The testing process allows the detection of errors in the application. It is the primary quality measure applied during the software's development. During testing, the system application is executed with specific test cases, and the software's output is assessed to establish if it is performing as to its expectations. In this inventory management software, unit testing will be applied on single modules as they are completed and become implemented. The modules will be tested using the following strategies:

Black Box Testing

This strategy will allow the test cases to be created as input conditions that fully implement all the functional requirements of the system program. This testing will detect errors in the following groups

- Missing or incorrect functions.
- Interface errors.
- Permanent errors.
- Errors in external database access or data structure.
- Initialization and termination errors.

The method of testing will allow only the input to be checked for accuracy if the logical flow of data is not checked.

White Box Testing

The test cases will be created based on the logic of each module by drawing flow graphs of the module and logical decisions that are tested on all the cases. This strategy is used to generate test cases in the following cases:

- Execute internal data structure to guarantee their validity
- Execute all loops at their boundaries within their operational bounds
- Ensure that all independent paths have been executed
- Execute all logical decisions on their false and true sides.

Supporting documents

The support documents consist of user documentation and technical documentation whereby the system personnel will maintain the application throughout its productive life(technical) and the people that will be using the systems as part of their daily lives (users). The user document will involve training by applying particular, general for the off-the-shelf software and the operating system. User documentation will also involve educational materials and problem solving assistance to the information system.

5.2 DATA CREATION FOR TESTING

Data creation for testing will be conducted using test functions through the NetBeans IDE test function as well as possible outside toolkits to test for program functionality and security protocols as appropriate.

5.3 BUG LIFE CYCLE

System bugs and vulnerabilities will be tracked using a Google document located in the team 4 Google drive area. Bugs assigned to the tracker will be corrected in a timely manner by an assigned team member and corrected according to bug severity with major issues being corrected immediately.

6 TEST STRATEGY

6.1 TESTING TYPES

- Manual testing of various features within the program to ensure complete operability and intuitiveness.
- Automatic testing of appropriate units to increase efficiency of testing during development phases.

6.2 TOOLS

Tool Name	Vendor	Version Date	Version
NetBeans	Apache	08/31/22	15
Junit	Open Source	N/A	5
AWS	Amazon	N/A	N/A

6.3 UNITS TESTED

- GUI testing will be conducted using both automatic and manual methods to ensure unit operability.
- Login/logout operability will mostly be tested using automatic methods, but manual testing also occurs during the troubleshooting of other components.
- Logging tests will mostly be manual, but automatic testing may be applied if appropriate.
- Database function testing will be primarily manual, but automatic testing may be applied if appropriate.

7 TEST ENVIRONMENT

Environment	Version
Java Development Kit	8.x
DynamoDB	2019.11.21

8 TEST SCHEDULE

8.1 PLANNING

Testing requirements determined at this phase

Activity	Start	End	Responsible
Prepare Test Plan	7th Nov 2022	8th Nov 2022	Ronald Hand
Review Test plan	8th Nov 2022	8th Nov 2022	Svyatoslav Mudryy
Prepare Test scripts, Checklists	9th Nov 2022	15th Nov 2022	Jerome Reed Arnaud Tako
Review Test Scripts	15th Nov 2022	15th Nov 2022	Jamie McCarthy
PrepareTest Environment	16th Nov 2022	18th Nov 2022	Jerome Reed
System Testing	19th Nov 2022	23rd Nov 2022	Ronald Hand
General Functionality Testing	24th Nov 2022	1st Dec 2022	Arnaud Tako
Performance Testing	5th Dec 2022	10th Dec 2022	Svyatoslav Mudryy

8.2 DESIGN PHASE

Team collaboration activities aimed at determining project type. This phase will compromise the development of the team's project plan, test plan, and initial user guide. The team will also determine project requirements such as systems requirements and tool utilization.

8.3 CODE COMPLETE

Testing will be performed throughout the code development process. Complete system testing will be performed at this stage, and all issues found are logged and assigned for correction or documented if appropriate.

8.4 FEATURE COMPLETE

Any unresolved issues will be logged and documented for correction at further program version itineration.

8.5 REGRESSION TEST

A final systems regression test will be executed to ensure system operability and performance after final coding changes have occurred. Any major issues detected will be immediately corrected, and regression testing will occur again.

9 CONTROL PROCEDURE

9.1 REVIEWS

Reviews will be conducted as appropriate for coding or requirement changes that may impact unit testing

9.2 BUG REVIEW

Bug reviews will be conducted as appropriate. A bug review will be conducted after major code, design, and requirements changes.

9.3 CHANGES

Changes will be approved by team majority consensus, and final approval determined by the project manager. Test cases will be changed as appropriate after preliminary changes have been implemented.

9.4 DEFECTS

Defects found will be logged in the team 4 bug tracker document located in the team 4 Google drive.

10 ROLES

Role	Name	Responsibility
Project Manager(PM)	Svyatoslav Mudryy	Overall project oversight
Requirement Manager/Technical Writer(RM/TE)	Jamie McCarthy	Develops project system requirements, and software documentation
Software Designer(SW)	Jerome Reed	Develops coding necessary for project requirements
Software Designer(SW)	Tako Arnaud	Develops coding necessary for project requirements
Test Director(TD)	Ronald Hand	Develops unit and overall testing of software

11 DELIVERABLES

Deliverable	Responsibility
Test Plan	Overview of test plans during design phase
Unit test reports	Final unit testing results and regression testing

12 ENTRY CRITERIA

Unit and system tests have been completed and determined to be of satisfactory quality. Likewise, unit and system test cases should be completely documented for the team's final report. Regression testing should also be documented for final reporting procedural purposes.

13 SUSPENSION CRITERIA

Significant issues surrounding the operability/usability of the RIMA will cause a stand down of testing until such a time those issues are corrected.

14 RESUMPTION CRITERIA

Once detected significant issues/defects/bugs are corrected, and then testing will be resumed at the discretion of the project manager and test director.

15 EXIT CRITERIA

No significant issues were detected that might affect system operability. All test cases have been executed without major fault, and minor issues have been properly logged and documented.

16 RISKS

Major issues or changes requiring a delay in development may also delay or degrade system/unit test functions, causing missed project milestones and concatenating possible systematic failures.

Appendix A: References

[Insert the name, version number, description, and physical location of any documents referenced in this document. Add rows to the table as necessary.]

The following table summarizes the documents referenced in this document.

Document Name and Version	**Description**	**Location**
<Document Name and Version Number>	*[Provide description of the document]*	*<URL or Network path where document is located>*

Appendix B: Key Terms

[Insert terms and definitions used in this document. Add rows to the table as necessary. Follow the link below to for definitions of project management terms and acronyms used in this and other documents.

http://www2.cdc.gov/cdcup/library/other/help.htm

The following table provides definitions for terms relevant to this document.

Term	**Definition**
[Insert Term]	*[Provide definition of the term used in this document.]*
[Insert Term]	*[Provide definition of the term used in this document.]*
[Insert Term]	*[Provide definition of the term used in this document.]*