<Project Name>

Test report

Version <

<

VERSION HISTORY

[Provide information on how the development and distribution of the Test Report was controlled and tracked. Use the table below to provide the version number, the author implementing the version, the date of the version, the name of the person approving the version, the date that particular version was approved, and a brief description of the reason for creating the revised version.]

|  |  |  |
| --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** |
| 1.0 | *Group 3* | *05/04/16* |
|  |  |  |
|  |  |  |
|  |  |  |

***Note to the Author***

[This document is a template of a **Test Report** document for a project. The template includes instructions to the author, example text, and fields that should be replaced with the values specific to the project.

* Blue italicized text enclosed in square brackets ([text]) provides instructions to the document author, or describes the intent, assumptions and context for content included in this document.
* Blue italicized text enclosed in angle brackets (<text>) indicates a field that should be replaced with information specific to a particular project.
* Text and tables in black are provided as examples of wording and formats that may be used or modified as appropriate to a specific project. These are offered only as suggestions to assist in developing project documents; they are not mandatory formats.

When using this template for your project document, it is recommended that you follow these steps:

1. Replace all text enclosed in angle brackets (e.g., <Project Name>) with the correct field values. These angle brackets appear in both the body of the document and in headers and footers.
2. Modify example text as appropriate to the specific project.
3. To add any new sections to the document, ensure that the appropriate header and body text styles are maintained. Styles used for the Section Headings are Heading 1, Heading 2 and Heading 3. Style used for boilerplate text is Body Text.
4. To update the Table of Contents, right-click and select “Update field” and choose the option- “Update entire table”
5. Before submission of the first draft of this document, delete this “Notes to the Author” page and all instructions to the author, which appear throughout the document as blue italicized text enclosed in square brackets.]

**Table of Contents**

[1.0 Introduction 5](#_Toc445795901)

[1.1 Purpose 5](#_Toc445795902)

[2.0 Test PLAN 5](#_Toc445795903)

[3.0 Test Assessment 5](#_Toc445795904)

[4.0 Test Results 5](#_Toc445795905)

[4.1 Unit/Module/System Testing 5](#_Toc445795906)

[4.2 System Testing 6](#_Toc445795907)

[4.3 User Acceptance Testing 6](#_Toc445795908)

[4.4 Regression Testing 7](#_Toc445795909)

[4.5 Performance Testing 7](#_Toc445795910)

[4.6 <Type of Test> 8](#_Toc445795911)

[5.0 Variances 8](#_Toc445795912)

[6.0 Test Instances 8](#_Toc445795913)

[6.1 Resolved Test Incidents 8](#_Toc445795914)

[6.2 Unresolved Test Incidents 8](#_Toc445795915)

[7.0 Recommendations 9](#_Toc445795916)

[APPENDIX A: REFERENCES 10](#_Toc445795917)

[APPENDIX B: KEY TERMS 11](#_Toc445795918)

# 1.0 Introduction

## 1.1 Purpose

This *Stores and Purchase Management* Test Report provides a summary of the results of test performed as outlined within this document.

# 2.0 Test PLAN

[Include basic information about what was tested and what happened. How the test data was chosen? And other relevant information.]

# 3.0 Test Assessment

[Enter a comprehensive assessment of your interpretation of how adequate the test was in light of how thorough the test plan said it should be? What wasn't tested well enough?]

# 4.0 Test Results

# The major issues observed during testing were the non-functioning of the ‘Check Status’, ‘Admin Request Approval’ use cases and Notification utility. These modules are not implemented/working to the fully desired extent.

# When testing with different inputs to expect desired outputs in the testing phase, errors for invalid input were never reported except login. For example, when admin is making a purchase request, entering a negative integer in any of the fields like cost or quantity is accepted by the system. Also, if we enter a string in any of these fields, no error is shown and the system seems to have hanged.

# It was observed during UI testing that consistency in the position of Notification utility wasn’t present. Sometimes it is on right side in navigation bar while sometimes it is o left side. The logout button is also redundant. There are two buttons to perform logout on a single page.

# And above all, the Notification utility doesn’t show notifications when it appears on left side. Instead it shows a dropdown containing ‘Link 2’ as an option. In case of admin this utility disappears on ‘purchase request’ page. The dashboard always disappears while making a request, as a result of which no means of navigation remains.

# Overall, the results of testing are not at all satisfying with reference to SRS. The software developers must take immediate action to resolve all the issues. The software doesn’t meet the requirements specified in the SRS.

## Unit/Module/System Testing

Unit, module, and system integration testing activities were performed during the development of the system build or release.

## 4.2 System Testing

The table below summarizes the results of system testing:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Tester | Pass/Fail | Severity of Defect [Low/ Medium/ High] | Summary of Defect | Closed prior to Production Release?  *[This column will be filled by the dev team.]* | Comments |
| 1. | 29/3/16 | Daval | Pass |  | No Defect | <Yes> or <No> | Login use case |
| 2. | 29/3/16 | Ashish | Pass |  | No Defect |  | Student- Making Request use case |
| 3. | 29/3/16 | Ajeet | Fail | High | Quantity field in Student Request Object also accepts negative Integers | Yes | It should accept only positive integers |
| 4. | 30/3/16 | Abhimanyu | Fail | Medium | Student – ‘Check Status’ use case not implemented | Yes | Use case should be implemented as mentioned in SRS |
| 5. | 30/3/16 | Ajeet | Pass |  | No Defect |  | Faculty– ‘Making Request’ |
| 6. | 30/3/16 | Abhimanyu | Pass |  | No Defect |  | Faculty– ‘Student Request Approval’ |
| 7. | 30/3/16 | Ashish | Fail | High | Quantity field in Faculty Request Object also accepts negative Integers | Yes | It should accept only positive integers |
| 8. | 31/3/16 | Daval | Fail | Medium | In Admin – ‘Admin Request Approval’ use case, no option to disapprove a request. | Yes | Use case should be implemented as mentioned in SRS |
| 9. | 31/3/16 | Abhimanyu | Fail | High | In Admin – ‘Admin Request Approval’ use case, on approving a request, all currently available resources of that type are assigned. | Yes | The mechanism of Approving a request by admin should be corrected to assign only requested number of resources. |
| 10. | 1/4/16 | Ashish | Fail | Low | Requests are not categorized a/c to their status as mentioned SRS | Yes | They should be categorized as mentioned in ‘Admin Request Approval’ use case |
| 11. | 1/4/16 | Ashish | Pass |  | No Defect |  | Admin– ‘Purchase request’ |
| 12. | 1/4/16 | Ajeet | Fail | High | All user input fields in Admin Purchase Request also accept negative Integers | Yes | They should accept only positive integers |
| 13. | 1/4/16 | Daval | Fail | High | If resources are available and admin approves one request, then all other requests in notification list get approved. | Yes | Approval of request should be correctly implemented. |
| 14. | 2/4/16 | Abhimanyu | Fail | High | All pending requests from different users are not shown simultaneously to admin | Yes | The implementation of displaying notifications to admin should be corrected |

*[If the test case failed, list the corresponding Test Incident ID in the Comments column.]*

## User Acceptance Testing

The table below summarizes the test cases employed for user acceptance testing and the test results obtained for each test case:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Tester | Pass/Fail | Severity of Defect [Low/ Medium/ High] | Summary of Defect | Closed prior to Production Release? *[This column will be filled by the dev team.]* | Comments |
| 1. | 1/4/16 | Ajeet | Pass |  | No Defect | <Yes> or <No> | Login (Functional correctness and completeness) |
| 2. | 2/4/16 | Daval | Pass |  | No Defect |  | Has Notification Functionality |
| 3. | 2/4/16 | Daval | Fail | Low | When faculty approves a request, popup shows improper message | Yes | Either popup should show proper message or popup should not be shown at all |
| 4. | 2/4/16 | Abhimanyu | Fail | High | In admin, user details also show object details.  Faculties are not shown in user details. | Yes | User details should show only user details, including faculties. Object details should be under a new option in dashboard. |
| 5. | 3/4/16 | Abhimanyu | Fail | Medium | For student and faculty, notification shifts from right to left in Navigation bar, and for admin it disappears, on clicking 2nd link in dashboard | Yes | Notification option should have a fixed position. Preferably right.  It should not disappear for admin. |
| 6. | 3/4/16 | Ajeet | Fail | Low | For zero notification, on clicking it shows a half hidden blank dropdown | No | It should show a proper message like – ‘No requests’ instead of blank, for zero notification |
| 7. | 3/4/16 | Ashish | Fail | Medium | Dashboard disappears on clicking 2nd link in it for all, i.e., Student, Faculty and Admin | Yes | It should not disappear so as to provide some means of navigation. |
| 8. | 3/4/16 | Ashish | Fail | High | No errors are shown anywhere for wrong input, except login | Yes | Proper error messages should be shown. Preferably in RED color. |
| 9. | 3/4/16 | Ajeet | Fail | Low | Logout option is redundant | No | It should be removed from dashboard |
| 10. | 4/4/16 | Abhimanyu | Pass |  | No defect |  | Data is shown in proper table format |
| 11. | 4/4/16 | Ashish | Pass |  | No defect |  | IRC channel given for help & support |
| 12. | 4/4/16 | Daval | Fail | High | In Admin – Purchase Request, Due Date to Expire is ambiguous | Yes | Title given is due date while placeholder is days. |

*[If the test case failed, list the corresponding Test Incident ID in the Comments column.]*

## Regression Testing

The table below summarizes the test cases employed for regression testing and the test results obtained for each test case:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Tester | Pass/Fail | Severity of Defect [Low/ Medium/ High] | Summary of Defect | Closed prior to Production Release? *[This column will be filled by the dev team.]* | Comments |
|  |  |  |  |  |  | <Yes> or <No> |  |

*[If the test case failed, list the corresponding Test Incident ID in the Comments column.]*

## Performance Testing

The table below summarizes the test cases employed for performance testing and the test results obtained for each test case:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Tester | Pass/Fail | Severity of Defect [Low/ Medium/ High] | Summary of Defect | Closed prior to Production Release? *[This will be filled by the dev team.]* | Comments |
| 1. | 2/4/16 | Ashish | Pass |  | Load | <Yes> or <No> | Can handle large database |
| 2. | 2/4/16 | Daval | Pass |  | Stress |  | Can handle large number of users |
| 3. | 3/4/16 | Abhimanyu | Pass |  | Scalability |  | Can be used by and scaled to large number of users |
| 4. | 4/4/16 | Ajeet | Pass |  | Volume |  | Can handle sufficiently large amount of data |

*[If the test case failed, list the corresponding Test Incident ID in the Comments column.]*

## <Type of Test>

The table below summarizes the test cases employed for <type of test (e.g., unit/ module/ interface testing)> and the test results obtained for each test case:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Tester | Pass/Fail | Severity of Defect [Low/ Medium/ High] | Summary of Defect | Closed prior to Production Release? *[This will be filled by the dev team.]* | Comments |
| 1. | 27/03/16 | Ajeet | Fail | High | The system shows errors with python 3+ | Yes | It should be mentioned in Readme and SRS both that it works only with python 2+ |
| 2. | 27/3/16 | Abhimanyu | Fail | High | System doesn’t start without package ‘requests’ | Yes | This package should be mentioned in requirements.txt |
| 3. | 27/3/16 | Ajeet | Fail | Medium | Source Code is not properly commented | Yes | Comments help readers to understand code |
| 4. | 28/3/16 | Abhimanyu | Fail | Low | User already logged in still login option is shown | Yes | If user is logged in and tries to access home page, redirect him to dashboard |
| 5. | 28/3/16 | Daval | Fail | Low | Login use case not mentioned in SRS. | No | Mention it in SRS |
| 6. | 28/3/16 | Ashish | Fail | Low | Name ProjectYasha has no significance with project | No | Actual name of project should be displayed |
| 7. | 28/3/16 | Daval | Fail | Low | On home page, options like- Source code, framework and tools used have no relevance for user |  | Instead a section for developers would be better |

*[If the test case failed, list the corresponding Test Incident ID in the Comments column.]*

# 5.0 Variances

The test cases were manually created as compared to the scripts and software being used in the industries but were sufficient enough to test all the boundary cases as well as major drawbacks in the system. Following are the main points:

1. Only Limited amount of data was considered for Load and Stress.
2. Only limited users were handled simultaneously through single machine.

# 6.0 Test Instances

[Provide a brief description of the unexpected results, problems, or defects that occurred during the testing.]

## Resolved Test Incidents

*[Identify all resolved test incidents and summarize their resolutions. Reference may be made to Test Incident Reports that describe in detail the unexpected results, problems, or defects reported during testing, along with their documented resolutions, which may be included as an appendix to this document.]*

*[This will be filled by the dev team.]*

## Unresolved Test Incidents

[Identify all unresolved test incidents and provide a plan of action for their resolution. Reference may be made to Test Incident Reports that describe in detail the unexpected results, problems, or defects reported during testing, which may be included as an appendix to this document.]

*[This will be filled by the dev team.]*

# Recommendations

The User Interface and System both need improvements. At the same time code also requires some attention.

Following are our suggestions:

1. The notification utility should be fully corrected as mentioned above in the system testing report.
2. Notification to admin should have a disapprove option.
3. Notification should have a fixed positon on every page. It should not disappear in case of admin.
4. Dashboard should also not disappear from all the pages of making a request, so as to provide a means of navigation.
5. Logout button should be removed from Dashboard, as it is redundant.
6. ‘Check Status’ use case should be implemented as mentioned in SRS.
7. The code isn’t commented, please provide comments for understandability.
8. ‘Admin Request Approval’ use case is not written correctly in SRS. There is a mix-up between faculty and admin. Correct it.
9. Categorization of requests according to their status is also not done. Please do it.
10. Check the user input for validation before storing it in database.
11. Add package ‘requests’ in requirements.txt.
12. Mention in readme that system works only with python 2+ not with python 3+.

APPENDIX A: REFERENCES

The following table summarizes the documents referenced in this document.

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Version** | **Description** |
| *SRS Document (SE\_UG201213022.docx)* | *1.0* | *It is the Software Requirements Specification document of the system ‘Stores and Purchase Management’ as provided by developer team.* |

APPENDIX B: KEY TERMS

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

|  |  |
| --- | --- |
| **Term** | **Definition** |
| *Load* | *To understand the behavior of the application under a specific expected load.* |
| *Stress* | *To determine the robustness of software by testing beyond the limits of normal operation.* |
| *Scalability* | *The capability of a system, network, or process to handle a growing amount of work.* |
| *Volume* | *It refers to the amount of data being transported between various processes/pages of the system.* |