**Table of Contents**

[1 INTRODUCTION 3](#_Toc140751159)

[2 SCOPE 3](#_Toc140751160)

[2.1 Functions to be tested. 3](#_Toc140751161)

[2.2 Functions not to be tested. 3](#_Toc140751162)

[3 QUALITY OBJECTIVES 4](#_Toc140751163)

[3.1 Primary Objectives 4](#_Toc140751164)

[3.2 Secondary Objectives 4](#_Toc140751165)

[4 TEST APPROACH 4](#_Toc140751166)

[4.1 Test Automation 4](#_Toc140751167)

[5 ROLES AND RESPONSIBILITIES 5](#_Toc140751168)

[6 ENTRY AND EXIT CRITERIA 5](#_Toc140751169)

[6.1 Entry Criteria 5](#_Toc140751170)

[6.2 Exit Criteria 6](#_Toc140751171)

[7 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS 6](#_Toc140751172)

[7.1 Suspension criteria 6](#_Toc140751173)

[7.2 Resumption criteria 6](#_Toc140751174)

[8 TEST STRATEGY 6](#_Toc140751175)

[8.1 QA role in test process 6](#_Toc140751176)

[8.2 Bug life cycle: 8](#_Toc140751177)

[8.3 Testing types Black 9](#_Toc140751178)

[8.4 Bug Severity and Priority Definition 10](#_Toc140751179)

[Severity List 10](#_Toc140751180)

[Priority List 11](#_Toc140751181)

[9 RESOURCE AND ENVIRONMENT NEEDS 11](#_Toc140751182)

[9.1 Testing Tools 11](#_Toc140751183)

[9.2 Configuration Management 12](#_Toc140751184)

[9.3 Test Environment 12](#_Toc140751185)

[10 TEST SCHEDULE 12](#_Toc140751186)

[APPROVALS: 13](#_Toc140751187)

[TERMS/ACRONYMS 13](#_Toc140751188)

**Project “OrangeHRM”**

Document Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Version | Description | Author | Reviewer | Approver |
| 23.07 | 5.5 | Test plan was created | Muhammad Mukhlis |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 1 **INTRODUCTION**

Dokumen ini di buat untuk menyelesaikan tugas akhir mata kuliah Pengujian perangkat Lunak yang diberikan oleh Septian Cahyadi, S.Kom., M.Kom., untuk melakukan testing pada website opensource milik OrangeGRM dengan tautan <https://opensource-demo.orangehrmlive.com/>. Semoga dengan laporan ini dapat memberikan sedikit penjelasan mengenai testing.

# 2 **SCOPE**

Dokumen testing ini secara garis besar menargetkan UI testing dan laporan output dari validasi data sesuai dengan keinginan masing-masing.

## **2.1 Functions to be tested**.

* Some GUI
* Some Search and Filters Logic

## 2.2 Functions not to be tested.

* Semua yang tidak disebutkan pada 2.1

# 3 QUALITY OBJECTIVES

## 3.1 Primary Objectives

Objek Utama pada testing ini adalah untuk mengetahui bahwa sistem dapat berfungsi dengan baik sesuai dengan dokumentasi yang ada, termasuk kualitas kebutuhan dan hal-hal lain.

Perubahan apa saja, berupa penambahan maupun penghapusan pada dokumen yang diperlukan, spesifikasi fungsi, spesifikasi desain, akan dimasukkan kedalam dokumentasi dan akan di uji oleh saya sebagai penguji.

## 3.2 Secondary Objectives

Objek Sekunder testing ini adalah untuk mengidentifikasi dan melihat masalah-masalah dan resiko yang ada, memberikan informasi kepada team ppengembang jika ada isu terkait, dan hal lain yang saya uji secara sederhana.

# 4 TEST APPROACH

Pendekatan yang digunakan dalam pengujian ini menggunakan analisis secara manual, dan akan di rekam secaraotomatis menggunakan Selenium IDE. Lalu data yang ada akan di input secara manual kedalam dokumen test case menggunakan web aplikasi <https://qase.io>.

## 4.1 Test Automation

Otomatisasi tes juka dilakukan secarasederhana menggunakan Selenium IDE dan browser Chrome.

# 5 ROLES AND RESPONSIBILITIES

|  |  |  |
| --- | --- | --- |
| Role | Staff Member | Responsibilities |
| Project Manager |  | 1. Acts as a primary contact for development and QA team. 2. Responsible for Project schedule and the overall success of the project. |
| QA Lead |  | 1. Participation in the project plan creation/update process.  2.Planning and organization of test process for the release. 3.Coordinate with QA analysts/engineers on any issues/problems encountered during testing. 4.Report progress on work assignments to the PM |
| QA |  | 1. Understand requirements 2. Writing and executing Test cases 3. Preparing RTM 4. Reviewing Test cases, RTM 5. Defect reporting and tracking 6. Retesting and regression testing 7. Bug Review meeting 8. Preparation of Test Data 9. Coordinate with QA Lead for any issues or problems encountered during test preparation/execution/defect handling. |

# 6 ENTRY AND EXIT CRITERIA

## 6.1 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, lab, hardware, software, and system administration support should be ready.
* QA resources have completely understood the requirements
* QA resources have sound knowledge of functionality
* Reviewed test scenarios, test cases and RTM

## 6.2 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.
* Cost – when the budget has been spent.
* The schedule has been achieved

# 7 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

## 7.1 Suspension criteria

* The build contains many serious defects which seriously or limit testing progress.
* Significant change in requirements suggested by client
* Software/Hardware problems
* Assigned resources are not available when needed by test team.

## 7.2 Resumption criteria

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

# 8 TEST STRATEGY

## 8.1 QA role in test process

* Understanding Requirements:
  + Requirement specifications will be sent by client.
  + Understanding of requirements will be done by QA
* Preparing Test Cases:

QA will be preparing test cases based on the exploratory testing. This will cover all scenarios for requirements.

* Preparing Test Matrix:

QA will be preparing test matrix which maps test cases to respective requirement. This will ensure the coverage for requirements.

* Reviewing test cases and matrix:
* Peer review will be conducted for test cases and test matrix by QA Lead
* Any comments or suggestions on test cases and test coverage will be provided by reviewer respective Author of Test Case and Test Matrix
* Suggestions or improvements will be re-worked by author and will be send for approval
* Re-worked improvements will be reviewed and approved by reviewer
* Creating Test Data:

Test data will be created by respective QA on client's developments/test site based on scenarios and Test cases.

* Executing Test Cases:
* Test cases will be executed by respective QA on client's development/test site based on designed scenarios, test cases and Test data.
* Test result (Actual Result, Pass/Fail) will updated in test case document Defect Logging and Reporting:

QA will be logging the defect/bugs in Word document, found during execution of test cases. After this, QA will inform respective developer about the defect/bugs.

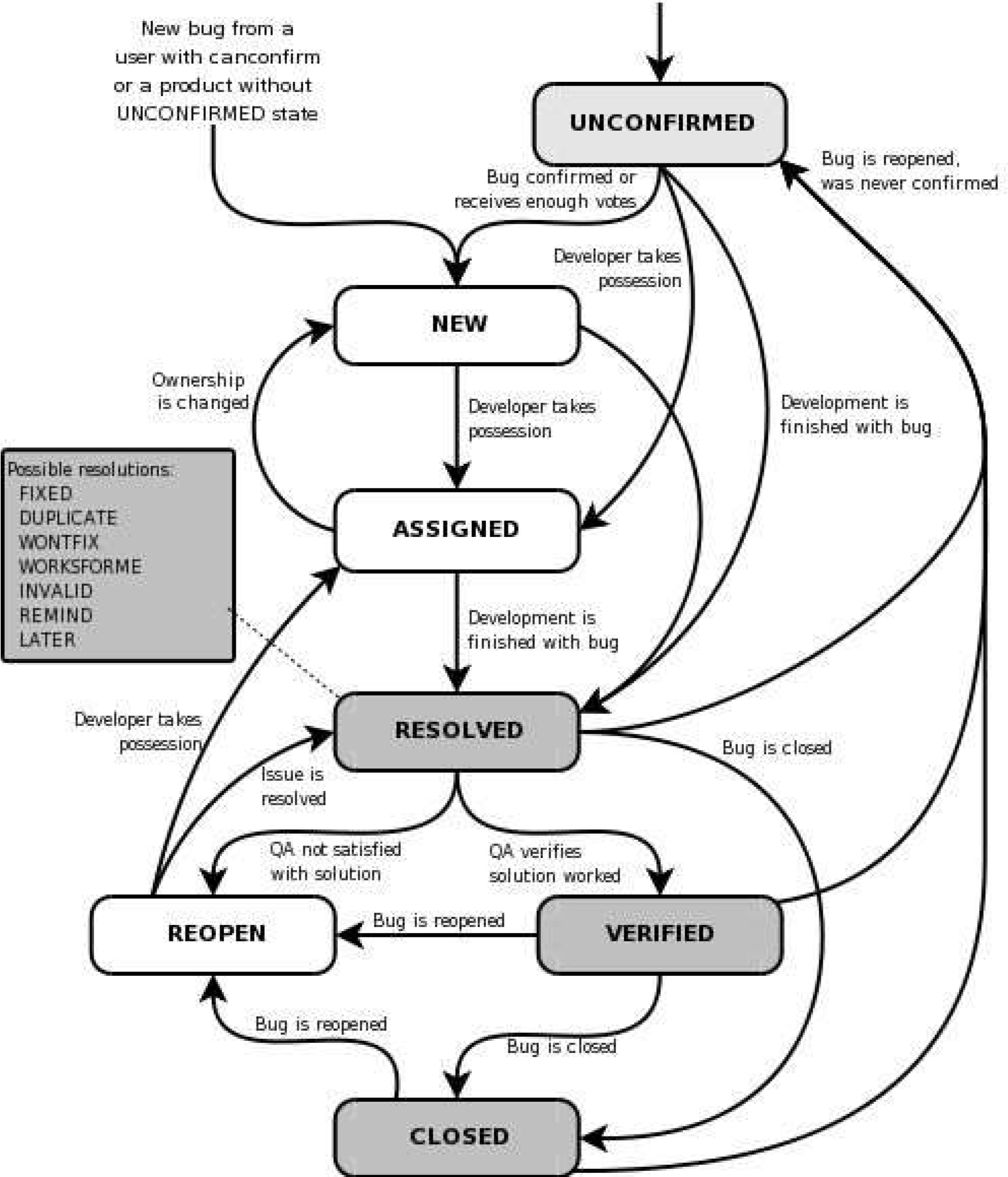
* Retesting and Regression Testing:

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

* Deployment/Delivery:
* Once all bugs/defect reported after complete testing is fixed and no other bugs are found, report will be deployed to client’s test site by PM.
* Once round of testing will be done by QA on client’s test site if required Report will be delivered along with sample output by email to respective lead and Report group.
* QA will be submitting the filled hard copy of delivery slip to respective developer.
* Once lead gets the hard copy of delivery slip filled by QA and developer, he will send the report delivery email to client.

## 8.2 Bug life cycle:

All the issues found while testing will be logged into Word document. Bug life cycle for this project is as follows:



8.3 Testing types

GUI Testing:

GUI testing will includes testing the UI part of report. It covers users Report format, look and feel, error messages, spelling mistakes, GUI guideline violations.

Functional Testing:

Functional testing is carried out in order to find out unexpected behavior of the report. The characteristic of functional testing are to provide correctness, reliability, testability and accuracy of the report output/data.

System Testing:

System testing of software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

## 8.4 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. The Test Lead will be responsible to see that a correct severity level is assigned to each bug.

The QA Lead, Development Lead and Project Manager will participate in bug review meetings to assign the priority of all currently active bugs. This meeting will be known as

“Bug Triage Meetings”. The QA Lead is responsible for setting up these meetings on a routine basis to address the current set of new and existing but unresolved bugs.

# Severity List

The tester entering a bug into GForge is also responsible for entering the bug Severity.

|  |  |  |
| --- | --- | --- |
| **Severity ID** | **Severity** | **Severity Description** |
| 1 | Critical | The module/product crashes or the bug causes nonrecoverable conditions. System crashes, GP Faults, or database or file corruption, or potential data loss, program hangs requiring reboot are all examples of a Sev. 1 bug. |
| 2 | High | Major system component unus able due to failure or incorrect functionality. Sev. 2 bugs cause serious problems such as a lack of functionality, or insufficient or unclear error messages that can have a major impact to the user, prevents other areas of the app from being tested, etc. Sev. 2 bugs can have a work around, but the work around is inconvenient or difficult. |
| 3 | Medium | Incorrect functionality of component or process. There is a simple work around for the bug if it is Sev. 3. |
| 4 | Minor | Documentation errors or signed off severity 3 bugs. |

# Priority List

|  |  |  |
| --- | --- | --- |
| **Priority** | **Priority Level** | **Priority Description** |
| 1 | Must Fix | This bug must be fixed immediately; the product cannot ship with this bug. |
| 2 | Should Fix | These are important problems that should be fixed as soon as possible. It would be an embarrassment to the company if this bug shipped. |
| 3 | Fix When Have  Time | The problem should be fixed within the time available. If the bug does not delay shipping date, then fix it. |
| 4 | Low Priority | It is not important (at this time) that these bugs be addressed. Fix these bugs after all other bugs have been fixed. Enhancements/ Good to have features incorporatedjust are out of the current scope. |

# 9 RESOURCE AND ENVIRONMENT NEEDS

## 9.1 Testing Tools

|  |  |  |
| --- | --- | --- |
| Process |  | Tool |
| Test case creation | Qase.io |  |
| Test case tracking | Qase.io |  |
| Test case execution | Manual, Selenium |  |
| Test case management | Qase.io |  |
| Defect management | Microsoft Word |  |
| Test reporting | PDF |  |
| Check list creating | Qase.io |  |
| Project structure | … |  |

## 9.2 Configuration Management

* Documents CM: SVN
* Code CM: Git

## 9.3 Test Environment

* Support level 1 (browsers):
* Windows 10: Chrome (latest)
* Support level 3:
* anything else

# 10 TEST SCHEDULE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Effort** | **Comments** |
| Test Planning | 08.00 | 13.00 |  | Support! |
| Review Requirements  documents | 13.00 | - |  |  |
| Create test basis | - | - |  |  |
| Staff and train new test resources | - | - |  |  |
| First deploy to QA test |  |  |  |  |
| environment | - |  |  |  |
| Functional testing –  Iteration 1 | - | - |  |  |
| Iteration 2 deploy to QA test environment | - | - |  |  |
| Functional testing –  Iteration 2 | - | - |  |  |
| System testing |  |  |  |  |
| Regression testing |  |  |  |  |
| UAT |  |  |  |  |
| Resolution of final defects and final build testing |  |  |  |  |
| Deploy to Staging  environment |  |  |  |  |
| Performance testing |  |  |  |  |
| Release to Production |  |  |  |  |

# APPROVALS:

|  |  |  |
| --- | --- | --- |
|  | **Project Manager** | **QA Lead** |
| **Name** |  |  |
| **Signature** |  |  |

# TERMS/ACRONYMS

The below terms are used as examples, please add/remove any terms relevant to the document.

|  |  |
| --- | --- |
| **TERM/ACRONYM** | **DEFINITION** |
| API | Application Program Interface |
| GUI | Graphical user interface |
| PM | Project manager |
| UAT | User acceptance testing |
| CM | Configuration Management |
| QA | Quality Assurance |
| RTM | Requirements Traceability Matrix |