**Table of Contents**

[1 INTRODUCTION 2](#_Toc21693)

[2 SCOPE 2](#_Toc21694)

[3 QUALITY OBJECTIVES 3](#_Toc21695)

[3.1 Primary Objectives 3](#_Toc21696)

[3.2 Secondary Objectives 3](#_Toc21697)

[4 TEST APPROACH 3](#_Toc21698)

[4.1 Test Automation 4](#_Toc21699)

[5 ROLES AND RESPONSIBILITIES 4](#_Toc21700)

[6 ENTRY AND EXIT CRITERIA 5](#_Toc21701)

[6.1 Entry Criteria 5](#_Toc21702)

[6.2 Exit Criteria 5](#_Toc21703)

[7 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS 5](#_Toc21704)

[7.1 Suspension criteria 5](#_Toc21705)

[7.2 Resumption criteria 6](#_Toc21706)

[8 TEST STRATEGY 6](#_Toc21707)

[8.1 QA role in test process 6](#_Toc21708)

[8.2 Bug life cycle: 7](#_Toc21709)

[8.3 Testing types 8](#_Toc21710)

[8.4 Bug Severity and Priority Definition 9](#_Toc21711)

[Severity List 10](#_Toc21712)

[Priority List 10](#_Toc21713)

[9 RESOURCE AND ENVIRONMENT NEEDS 11](#_Toc21714)

[9.1 Testing Tools 11](#_Toc21715)

[9.2 Configuration Management 11](#_Toc21716)

[9.3 Test Environment 11](#_Toc21717)

[10 TEST SCHEDULE 12](#_Toc21718)

[APPROVALS 13](#_Toc21719)

[TERMS/ACRONYMS 13](#_Toc21720)

**Test Plan**

**Project “Student Management”**

Document Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Version | Description | Author | Reviewer | Approver |
| 07.10 | 0.1 | Test plan was created | Huong |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# INTRODUCTION

The Test Plan has been created to facilitate communication within the team members. This document describes approaches and methodologies that will apply to the unit, integration and system testing of the “Student Management System” application that can be used for managing students.

It includes the objectives, test responsibilities, entry and exit criteria, scope, schedule major milestones, entry and exit criteria and approach. This document has clearly identified what the test deliverables will be, and what is deemed in and out of scope.

# SCOPE

The document mainly targets the GUI testing and validating data in report output as per Requirements Specifications provided by Client.

2.1 Functions to be tested.

- Functional testing:

* Unit Testing
* Application functions: view, search, insert, update, delete.

2.2 Functions not to be tested.

* Usability (friendly, attractive interface): The Simple Console App does not contain an application interface.

# QUALITY OBJECTIVES

## Primary Objectives

* Ensure the Application Under Test conforms to functional requirements
* Ensure the AUT meets the quality specifications defined by the client at the end of the project development life cycle.

## Secondary Objectives

* Bugs/issues are identified and fixed before going live.

# TEST APPROACH

* All types of testing and level of testing are determined in Test Strategy.
* Test case will be created during exploratory testing
* Team also used experience-based testing and error guessing

## Test Automation

Automated unit tests are part of the development process, and UI smoke-tests from CHL01 must be also automated during which performance data must be captured

# 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. |
| Team leader |  | 1. Participation in the project plan and work division for team members.  2.Planning and organization of the test process for the release.  3.Communication with all team members to ensure all members work themselves properly. |
| Tester |  | 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. |

# ENTRY AND EXIT CRITERIA

## Entry Criteria

* All the necessary documentation, design, and requirements information should be available that will allow testers to operate the system and judge the correct behavior.
* Proper test data is available.
* The test environment such as, lab, hardware, software, and mobile Operating System 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

## Exit Criteria

* A certain level of requirements coverage has been achieved.
* No high priority or severe bugs are left outstanding.
* 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

# SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

## Suspension criteria (when testing stop)

* Significant change in requirements suggested by client
* Software/Hardware problems
* Assigned resources are not available when needed by the test team.

## Resumption criteria

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

# TEST STRATEGY

## 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 a test matrix which maps test cases to respective requirements. 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 the client's development/test site based on designed scenarios, test cases and Test data.
* Test result (Actual Result, Pass/Fail) will be updated in test case document Defect Logging and Reporting:

QA will be logging the defect/bugs in Word documents, found during execution of test cases. After this, QA will inform the 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 the 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/defects reported after complete testing are fixed and no other bugs are found, the report will be deployed to the client's test site by PM.

•  Once a round of testing will be done by QA on the client's test site if required Report will be delivered along with sample output by email to the respective lead and Report group.

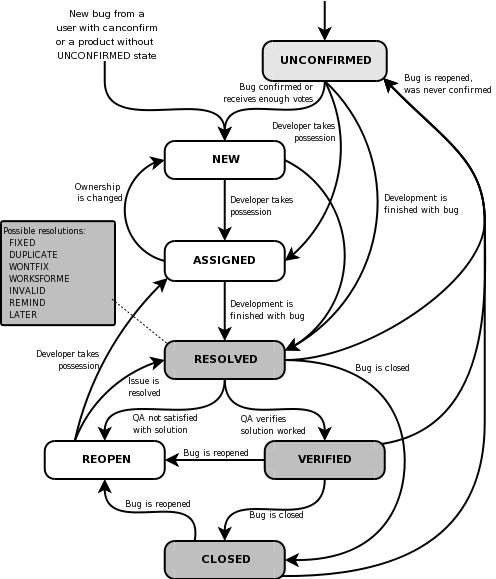
•  QA will be submitting the filled hard copy of the delivery slip to the respective developer.

•  Once the lead gets the hard copy of the delivery slip filled by QA and developer, he will send the report delivery email to the client.

## Bug life cycle:

All the issues found while testing will be logged into Word documents.

Bug life cycle for this project is as follows:



## Testing types

Functional Testing:

* All functions should be to provide correctness, reliability and accuracy with data and report output.
* Find out the unexpected behavior of the report.

System Testing:

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

* Testing system of booking cabs: select origin, select destination, select trip type, select pick-up date, select duration, enter email and search for cabs.

## 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 components unusable 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, prevent 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 the 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 incorporated just are out of the current scope. |

# RESOURCE AND ENVIRONMENT NEEDS

## Testing Tools

|  |  |  |
| --- | --- | --- |
| Process |  | Tool |
| Test case creation | Microsoft Excel |  |
| Test case tracking | Microsoft Excel |  |
| Test case execution | Manual, Selenium |  |
| Test case management | Microsoft Excel |  |
| Defect management | Microsoft Word |  |
| Test reporting | PDF |  |
| Check list creating | Microsoft Excel |  |
| Project structure | Mind Map (.xmind) |  |

## Configuration Management

* Documents CM: SVN
* Code CM: Git

## Test Environment 0020

# Support level 1 (browsers): - Windows : Edge, Chrome (latest), Firefox (latest), Safari (latest) -Mac OS X: Chrome (latest), Firefox (latest), Safari (lates)

# TEST SCHEDULE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Effort** | **Comments** |
| Test Planning | 06/10 | 07/10 |  |  |
| Review Requirements  documents | 07/10 | 08/10 |  |  |
| Create test basis | 08/10 | 12/10 |  |  |
| Staff and train new test resources | - | - |  |  |
| First deploy to QA test environment |  |  |  |  |
| Functional testing –  Iteration 1 | 10/10 | 13/10 |  |  |
| Iteration 2 deploy to QA test environment | 14/10 | 15/10 |  |  |
| Functional testing –  Iteration 2 | 16/10 | 17/10 |  |  |
| 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 |