TEST PLAN

**Version2.0**

**Revision History and Sign-off Sheet**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Content** |
| 1.0 | 05/05/2017 | Qui Ngo | Create Test Plan Version 1.0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Approvers/Reviewers List – To track who has reviewed and sign-off the test plan | | | | |
| **Name** | **Role** | **Approver/Reviewer** | **Approval/Review Date** | **Comment** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| Reference Documents – Clearly mark the document used as an input to create the test plan | | |
| **Version** | **Date** | **Document Name** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Table of Contents**

[**1.** **Introduction** 6](#_Toc482833080)

[**1.1.** **Purpose** 6](#_Toc482833081)

[**1.2.** **Intended Audience** 6](#_Toc482833082)

[**1.3.** **Project Background** 7](#_Toc482833083)

[**1.3.1.** **Objective** 7](#_Toc482833084)

[**1.3.2.** **Outcome** 7](#_Toc482833085)

[**2.** **Test Strategy** 7](#_Toc482833086)

[**2.1.** **Test Objectives** 7](#_Toc482833087)

[**2.2.** **Assumptions and Principles** 7](#_Toc482833088)

[**2.2.1.** **Test Assumptions** 7](#_Toc482833089)

[**2.2.2.** **Test Principles** 8](#_Toc482833090)

[**2.3.** **Data Approach** 8](#_Toc482833091)

[**2.4.** **Scope and Levels of Testing** 8](#_Toc482833092)

[**2.4.1.** **Static Test** 8](#_Toc482833093)

[**2.4.2.** **Exploratory Test** 8](#_Toc482833094)

[**2.4.4.** **Integration Test** 9](#_Toc482833095)

[**2.4.5.** **Quality Attribute Test** 9](#_Toc482833096)

[**2.4.6.** **System Test** 9](#_Toc482833097)

[**2.4.7.** **User Acceptance Test** 9](#_Toc482833098)

[**3.** **Test Environment** 10](#_Toc482833099)

[**3.1.** **Hardware** 10](#_Toc482833100)

[**3.2.** **Software** 10](#_Toc482833101)

[**3.3.** **Defect Tracking and problem reporting** 10](#_Toc482833102)

[**3.3.1.** **Overview** 11](#_Toc482833103)

[**3.3.2.** **Role and Activities** 11](#_Toc482833104)

[**4.** **Execution Strategy** 12](#_Toc482833105)

[**4.1.** **Test strategy** 12](#_Toc482833106)

[**4.1.1.** **Features to be tested** 12](#_Toc482833107)

[**4.1.2.** **Functional testing** 12](#_Toc482833108)

[**4.2.** **Pass/Fail Criteria** 12](#_Toc482833109)

[**4.3.** **Suspend/ Resume Criteria** 13](#_Toc482833110)

[**5.** **Management Approach** 13](#_Toc482833111)

[**5.1.** **Design and Execution Process** 13](#_Toc482833112)

[**5.2.** **Human Resources** 13](#_Toc482833113)

[**5.3.** **Training need** 14](#_Toc482833114)

[**5.4.** **Schedules** 14](#_Toc482833115)

[**5.4.1.** **Test Effort Estimation** 14](#_Toc482833116)

[**5.4.2.** **Test Deliverables** 14](#_Toc482833117)

[**5.5.** **Test Risks and Mitigation Factors** 15](#_Toc482833118)

[**6.** **Glossary** 15](#_Toc482833119)

**List of Tables**

[**Table 1. Intended Audience** 7](#_Toc482832531)

[**Table 2. Hardware** 11](#_Toc482832532)

[**Table 3. Software** 11](#_Toc482832533)

[**Table 5. Defect Tracking Role and Activities** 12](#_Toc482832534)

[**Table 6. Features to be tested** 12](#_Toc482832535)

[**Table 7. Human Resource** 14](#_Toc482832536)

[**Table 8. Training need** 14](#_Toc482832537)

[**Table 9. Test Effort Estimation** 14](#_Toc482832538)

[**Table 10. Test Deliverables** 15](#_Toc482832539)

[**Table 11. Test Risks and Mitigation Factors** 15](#_Toc482832540)

**List of Figures**

[**Figure 2. Defect tracking and problem reporting** 11](#_Toc482832515)

1. **Introduction** 
   1. **Purpose**

This test plan describes the testing approach and overall framework that will drive the testing of ECB system being developed for the Faculty of Information Technology of Van Lang University by TripleXTeam. The document introduces:

* Test Strategy: rules the test will be based on, including the givens of the project, description of the process to set up a valid test.
* Execution Strategy: describes how the test will be performed and process to identify and report defect, and to fix and implement fixes.
* Test Management: process to handle the logistics of the test and all the event that come up during execution.
  1. **Intended Audience**

|  |  |  |
| --- | --- | --- |
| **Role** | **Name** | **Responsibility** |
| Test Leader | Qui Ngo | * Analyze the product * Design test strategy * Plan Test environment * Resource planning * Deliverables * Provide technical direction * Design Test Plan * Prepare summary Report * Create Test Plan |
| Project Manager | Van Phan | * Monitoring & controlling progress |
| Tester | Nhuan Tran, Tan Huynh | * Develop Test Case * Execute Test Case * Log result * Report the defect * Run Test Case |
| Develop Team | Thang Nguyen, Nhuan Tran, Tan Huynh, Qui Ngo | * Fix bugs |
| User | Phung Bui, Manh Nguyen, Thoa Nguyen | * Perform User Acceptance testing |

**Table 1. Intended Audience**

* 1. **Project Background**
     1. **Objective**

Objective of this project is TripleX team will create Website Electronic Contact Book and Android Mobile App to help Faculty of Information Technology build a system to support the management of student training and learning from Faculty to parent

* Faculty information (mark, tuition, learning progress, study plan, Schedule)
* Manage mark information of student and can post mark board by import file or type manual on Website
* Notices mark of student, tuition from the university and faculty to parents by Android mobile app on parent’s mobile
* System attractive

The project will be considered successful if functions as specified in the List function of Website (functions that negotiated with customer) within 30 weeks.

* + 1. **Outcome**

When the project closed, TripleX team will release Website ECB, Android Mobile App and all document in project.

1. **Test Strategy** 
   1. **Test Objectives**

Ours testing objectives are insuring that:

* All required functionalities of ECB system are in place and are working correctly according client’s requirements.
* The test will execute and verify the test scripts, identify, fix and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects for fixing.
* The final product of test is two fold:
* A production-ready software.
* A set of stable test script that can be reused for Functional and UAT test execution.
  1. **Assumptions and Principles**
     1. **Test Assumptions**
* Exploratory Testing would be carried out once the build is ready for testing.
* Test case design activities will be performed by Tester.
* Test environment and preparation activities will be owned by Test Leader
* Customers will review and sign-off all Test cases prepared by Test Team prior to start of Test execution.
* Test Leader will provide Defect fix plans based on the Defect meetings during each cycle to plan. The same will be informed to Test team prior to start of Defect fix cycles.
  + 1. **Test Principles**
* Testing will be focused on meeting the business objectives, cost efficiency, and quality.
* There will be common, consistent procedures for all teams supporting testing activities.
* Testing processes will be well defined, yet flexible, with the ability to change as needed.
* Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
* Testing environment and data will emulate a production environment as much as possible.
* Testing will be a repeatable, quantifiable, and measurable activity.
* Testing will be divided into distinct phases, each with clearly defined objectives and goals.
* There will be entrance and exit criteria.
  1. **Data Approach**

In functional testing, ECB system will contain pre-load test data and which is used for testing activities. All information used test provide by staff.

* 1. **Scope and Levels of Testing** 
     1. **Static Test**
* **Purpose:** This testing is to improve the quality of software products by finding errors in early stages of the development cycle. This testing is also called as Non-execution technique or verification testing.
* **Scope:** During initial phase of testing to catch defect early.
* **Staffs:** Testing Team
* **Method:** Static testing involves manual or automated reviews of the documents.
* **Execution time:** Start Project
  + 1. **Exploratory Test**
* **Purpose:** This test is to make sure critical defect are removed before the next levels of testing can start.
* **Scope:** Use important functions
* **Staffs:** Testing Team
* **Method:** This exploratory testing is carried out in the application without any test script and document.
* **Execution time:** At the beginning of each cycle
  + 1. **Functional Test**
* **Purpose:** Function testing will be performed to check the functions of application. The functional testing is carried out by feeding the input and validates the output from the application.
* **Scope:** All function of front end and back end
* **Staffs:** Testing Team
* **Method:** The test will be performed according Functional script, which are store in SVN Github
* **Execution time:** After Exploratory test is completed.
  + 1. **Integration Test**
* **Purpose:** To test integration between this function with another function
* **Scope:** Front end and Back end
* **Staffs:** Testing Team
* **Method:** The test will be performed according Integration script, which are storage in SVN Github
* **Execution time:** After Functional test is completed
  + 1. **Quality Attribute Test**
* The Quality attributes were described in the SRS must be fully reflected in the System test specification.
* When develop quality attribute test cases & test data should apply Black box testing techniques, such as:
* Performance testing: is a means of quality assurance (QA). It involves testing software applications to ensure they will perform well under their expected workload.
* Usability testing: This testing mainly focuses on the user's-ease to use the application, flexibility in handling controls and ability of the system to meet its objectives.
* Security testing: Identify the threats in the system and measure its potential vulnerabilities. It also helps in detecting all possible security risks in the system and help developers in fixing these problems through coding.
  + 1. **System Test**
* The test case including all the function on product and must be implementing by tester.
* When developing functional test case and test data should apply Black box testing technique.
* Report testing result.
  + 1. **User Acceptance Test**
* **Purpose:** this test focuses on validating of the customer. It allows the end users to complete one final review of the system prior to deployment.
* **Scope:** Requirement of customer
* **Staffs:** The User Acceptance Test is performed by the end users.
* **Method:** Test team write the UAT test cases based on the inputs from End user and customer
* **Test Acceptance Criteria:**
* No critical defects open
* Business process works satisfactorily
* UAT meeting with all stakeholders
* **Execution time:** After Functional test is completed

1. **Test Environment**
   1. **Hardware**

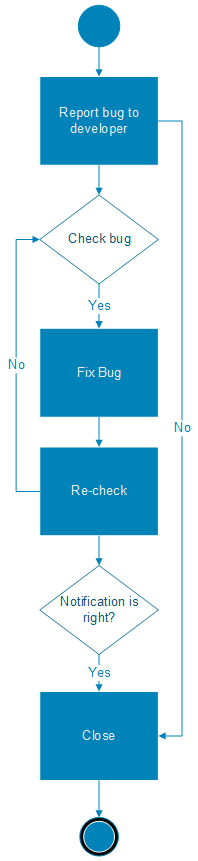
|  |  |
| --- | --- |
| **Hardware** | **Description** |
| Number |  |
| Type |  |
| Size |  |
| Capacity |  |
| Memory |  |

**Table 2. Hardware**

* 1. **Software**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Purpose** | **Type** | | **Note** |
| **Open source/Commercial** | **Supplier** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Table 3. Software**

* 1. **Defect Tracking and problem reporting**
     1. **Overview**

**Figure 2. Defect tracking and problem reporting**

* + 1. **Role and Activities**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Role** | **Step** | **Activities** | **Output** |
| Test Report | * Test Leader * Develop Team | Report Bug to Developer | Tester will send report bugs to develop team. |  |
|  | * Develop Team | Check Bug | Develop Team will check and fix bugs and release build | Build realease with bugs fixed |
|  | * Develop Team | Fix Bug |  |  |
| Build Release | * Tester | Recheck | Tester will rechek and if function righ test case will close. If test case fail tester will re-send to develop team | Test Report update |

**Table 5. Defect Tracking Role and Activities**

1. **Execution Strategy**
   1. **Test strategy**
      1. **Features to be tested**

|  |  |  |
| --- | --- | --- |
| **No.** | **Features** | **Test Description** |
|  |  |  |

**Table 6. Features to be tested**

* + 1. **Functional testing**

Functional testing is a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (not like in white-box testing).

Programs will enter System/Integration test after all critical defects have been corrected. A program may have up to two Major defects as long as they do not impede testing of the program

* + 1. **System testing**

The quality attributes described in Architectural driver specification must be fully reflected in the System test specification.

Programs will enter System/Integration test after all critical defects have been corrected. A program may have up to two Major defects if they do not impede testing of the program.

When developing quality attributes test case and test data should apply black-box testing techniques

* + 1. **Acceptance testing**

A set of test cases are taken from System test cases and executed in the user working environment.

In this type of test, the software will be tested by the user to find out whether it meets with their requirements and expectations.

In this phase, the tester can perform or the customer has their own tester to perform it.

Customer will test functional requirement.

* 1. **Pass/Fail Criteria**

The test process will be completed when the project leader will be satisfied with the result of the test. For this, at least 90% of test cases must pass; all functionalities must be covered in those test cases and most of all, high and medium severity defects must be detected and fixed. Minor defects can be ignored, but with the assurance that it does not lead to severe defect. The project leader will decide whether the detected defects and criticality will cause the release of ECB system to delay.

* 1. **Suspend/ Resume Criteria**
* Testing will be suspended if:
  + A failure of severity critical and high is observed.
* Testing will be resume when defect causing a software failure is repaired or hardware failure have been repaired.

1. **Management Approach** 
   1. **Design and Execution Process**
      * Reference ECB\_TE\_TestProcess\_Ver1.0
   2. **Human Resources**

|  |  |  |
| --- | --- | --- |
| **Role** | **Knowledge and skill** | **Name** |
| Test Team | * Have knowledge Software testing process * Ability to read and understand all requirement docs, design and architect docs * Ability to develop good test plan, test case. * Communication and Presentation Skills * Time & Task Management skills * Problem solving skills * Communication skills | * Qui Ngo * Nhuan Tran * Tan Huynh |
| Develop Team | * Ability to read and understand all requirement docs, design and architect docs * Experience on program language PHP, Android and Database MySQL * Communication and Presentation Skills * Time & Task Management skills * Problem solving skills * Communication skills * Resolves Bugs * Test project document * Developer test case * Run Test Case * Tracking defect | * Qui Ngo * Thang Nguyen * Nhuan Tran * Tan Huynh |
| Project Manager | * Listening skills * Analytical skills * Have a logical mind * Master the software development life cycle. * Ability to press people to track and measure their work. * Ability to communicate to stake holders & team members. * Ability to prepare schedule and monitor work progress. * Problem solving skills * Communication skills | * Van Phan |

**Table 7. Human Resource**

* 1. **Training need**

|  |  |  |
| --- | --- | --- |
| **No.** | **Descriptions** | **Members** |
| 1 | Training to develop Functional test cases | Qui Ngo |
| 2 | Training to develop User Acceptance test cases | Qui Ngo |
| 3 | Training to develop Integration test cases | Qui Ngo |
| 4 | Training to develop System test cases | Qui Ngo |

**Table 8. Training need**

* 1. **Schedules**
     1. **Test Effort Estimation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Description** | **Beginning Date** | **Ending Date** | **Human Resources** |
| 1 | Test Plan | 16/05/2016 |  | Qui Ngo |

**Table 9. Test Effort Estimation**

* + 1. **Test Deliverables**

|  |  |  |
| --- | --- | --- |
| **No.** | **Deliverable Name** | **Descriptions** |
| 1 | ECB\_TE\_TestPlan\_VerX.X | This document describes Test Strategy and plan when executing Testing phase of Project |
| 2 | ECB\_TE\_SystemTestCase\_VerX.X | This document describes all test cases for System test phase when executing Testing of this project |
| 3 | ECB\_TE\_IntegrationTestCase\_VerX.x | This document describes all detail test cases for Integration test phase when executing Testing of this project |
| 4 | ECB\_TE\_UserAcceptanceTestCase\_VerX.X | This document describes all detail test cases for User Acceptance test phase when executing Testing of this project |
| 5 | ECB\_TE\_TestReport\_VerX.X | This document describes the total number of pass and fails of test cases after being executed. |

**Table 10. Test Deliverables**

* 1. **Test Risks and Mitigation Factors**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Risk** | **Severity** | **Mitigation plan** |
| 1 | Failure to integrate with systems | High | The team must identify the root cause of the problem that make the schedule delay |
| 2 | Integration testing environments aren't available | High | The team must plan for integration test |
| 3 | Defect | High | The team must tool test for during project to avoid the defect happened |
| 4 | Failure to integrate components | High | The team need to integrate trial several times |

**Table 11. Test Risks and Mitigation Factors**

1. **Glossary**

|  |  |  |
| --- | --- | --- |
| **No.** | **Glossary** | **Descriptions** |
| 1. | ECB | Electronic Contact Book |
| 2. | UAT | User Acceptance Test |
| 3. | QA | Quality assurance |