MASTER TEST PLAN

**Version1.3**

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Content** | **Status** |
| 1.0 | 25/05/2016 | Quyen Nguyen Thi Thuy | Create document | Done |
| 1.1 | 26/05/2016 | Quyen Nguyen Thi Thuy | Update figure Detail Process | Done |
| 1.2 | 02/06/2016 | Quyen Nguyen Thi Thuy | Update figure Defect report and tracking process | Done |
| 1.3 | 09/06/2016 | Quyen Nguyen Thi Thuy | Update figure Review document | Done |
|  |  |  |  |  |

**Team Name**

1. **Tan Duy Nguyen**
2. **Son Thien Hoang**
3. **Quan Minh Ngo**
4. **Qui Long Ngo**
5. **Quyen Thi Thuy Nguyen**
6. **Hiep Le**

**Table of Contents**

[1. Introduction 7](#_Toc453234338)

[1.1. Project overview 7](#_Toc453234339)

[1.2. Purpose of Test plan 7](#_Toc453234340)

[1.3. Scope of Test plan 7](#_Toc453234341)

[2. Process 7](#_Toc453234342)

[2.1. Testing process 7](#_Toc453234343)

[2.2. Detail Process 8](#_Toc453234344)

[2.3. Defect report and tracking process 10](#_Toc453234345)

[3. Test Criteria 11](#_Toc453234346)

[3.1. Test Entry/ Exit Criteria 11](#_Toc453234347)

[3.1.1. Test Entry Criteria 11](#_Toc453234348)

[3.1.2. Test Exit Criteria 11](#_Toc453234349)

[3.2. Suspension/ Resumption Criteria 11](#_Toc453234350)

[4. Test Strategy 11](#_Toc453234351)

[4.1. Review Document 11](#_Toc453234352)

[4.2. Unit Testing 12](#_Toc453234353)

[4.3. System Testing 13](#_Toc453234354)

[4.3.1. Description 13](#_Toc453234355)

[4.3.2. Types 13](#_Toc453234356)

[4.4. Integration Testing 13](#_Toc453234357)

[4.4.1. Description 13](#_Toc453234358)

[4.4.2. Types 13](#_Toc453234359)

[4.5. Acceptance Testing 14](#_Toc453234360)

[4.5.1. Description 14](#_Toc453234361)

[4.5.2. Types 14](#_Toc453234362)

[4.6. Automation Testing 14](#_Toc453234363)

[4.6.1. Deccription 14](#_Toc453234364)

[4.6.2. Types of function to be automated 14](#_Toc453234365)

[5. Feature to be test 14](#_Toc453234366)

[6. Feature not to be test 14](#_Toc453234367)

[7. Milestones and Schedule 14](#_Toc453234368)

[7.1. Milestones 14](#_Toc453234369)

[7.2. Schedule 15](#_Toc453234370)

[8. Risk and Mitigation 15](#_Toc453234371)

[9. Measure and Metrics 15](#_Toc453234372)

[10. Human Resources Participate Test 16](#_Toc453234373)

[10.1. Tester Information 16](#_Toc453234374)

[10.2. Roles and Responsibilities 16](#_Toc453234375)

[10.3. Structure of Meeting 17](#_Toc453234376)

[10.4. Training Schedule 17](#_Toc453234377)

[11. Test Environments 17](#_Toc453234378)

[11.1. Hardware 17](#_Toc453234379)

[11.2. Software 17](#_Toc453234380)

[11.3. Test Tool 17](#_Toc453234381)

[12. Deliverable and Glossary 18](#_Toc453234382)

[12.1. Deliverable 18](#_Toc453234383)

[12.2. Glossary 18](#_Toc453234384)

**List of Table**

[Table 1.Severity 10](#_Toc453234264)

[Table 2. Priority 10](#_Toc453234265)

[Table 3. Review Document 11](#_Toc453234266)

[Table 4. Feature to be test 13](#_Toc453234267)

[Table 5. Feture not to be test 13](#_Toc453234268)

[Table 6. Milestones 14](#_Toc453234269)

[Table 7. Schedule 14](#_Toc453234270)

[Table 8. Risk and Mitigation 14](#_Toc453234271)

[Table 9. Percent of test case Pass 15](#_Toc453234272)

[Table 10. Percent of test case Fail 15](#_Toc453234273)

[Table 11. Defect by Severity 15](#_Toc453234274)

[Table 12. Percentage test case executed 15](#_Toc453234275)

[Table 13. Percentage test case not executed 15](#_Toc453234276)

[Table 14. Test Information 15](#_Toc453234277)

[Table 15. Roles and Respondibilities 16](#_Toc453234278)

[Table 16. Structure of Meeting 16](#_Toc453234279)

[Table 17. Training Schedule 16](#_Toc453234280)

[Table 18. Hardware 16](#_Toc453234281)

[Table 19. Software 16](#_Toc453234282)

[Table 20. Test Tool 17](#_Toc453234283)

[Table 21. Deliverable 17](#_Toc453234284)

[Table 22. Glossary 17](#_Toc453234285)

**List of Figure**

[Figure 1. Testing Process 8](#_Toc453838033)

[Figure 2. Detail Process 9](#_Toc453838034)

[Figure 3. Defect report and tracking process 10](#_Toc453838035)

[Figure 4. Review Document 13](#_Toc453838036)

1. **Introduction**

This document describes the plan for testing the Faculty Information System(FIS).

* 1. **Project overview**
* Current, the Faculty of Information Technology don’t have a channel contact with students. The shared information between faculty staffs and students mainly viva directly meeting, email or papers posted on the note board.
* This project will create a new system help faculty resolve issuers.
  1. **Purpose of Master Test plan**

This test plan document supports the following objectives:

* To uncover errors in function, logic, or implementation for any representation of the software
* To verify that the software under review meets its requirements
* To ensure that the software has been represented according to predefined standards
* To achieve software that is developed in a uniform manner
* To make projects more manageable
  1. **Scope of Master Test plan**

Scope of test plan is to help test members to do tasks following test plan and schedule that are defined and approved. This specification describes strategies and plan to implement phases of the testing in the FIS system project. The detailed descriptions of test cases in each phase will be executed in the test specifications.

1. **Process**
   1. **Testing process**

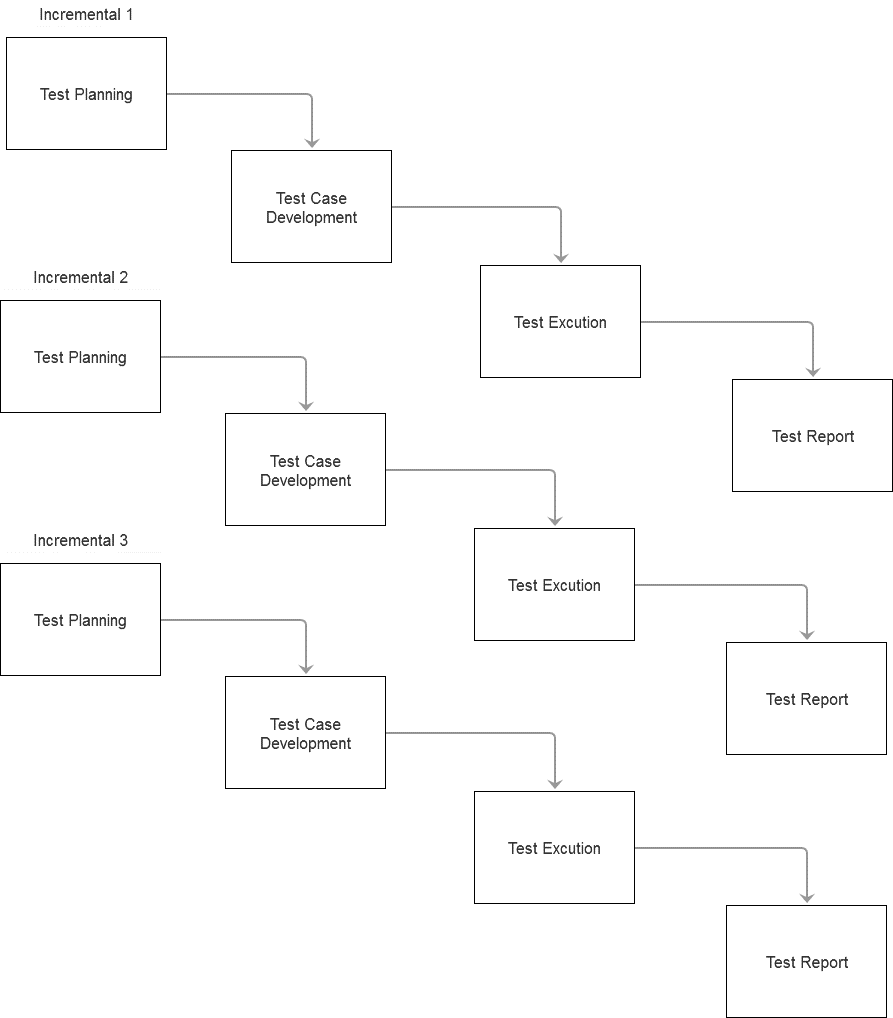


Figure 1. Testing Process

* 1. **Detail Process**

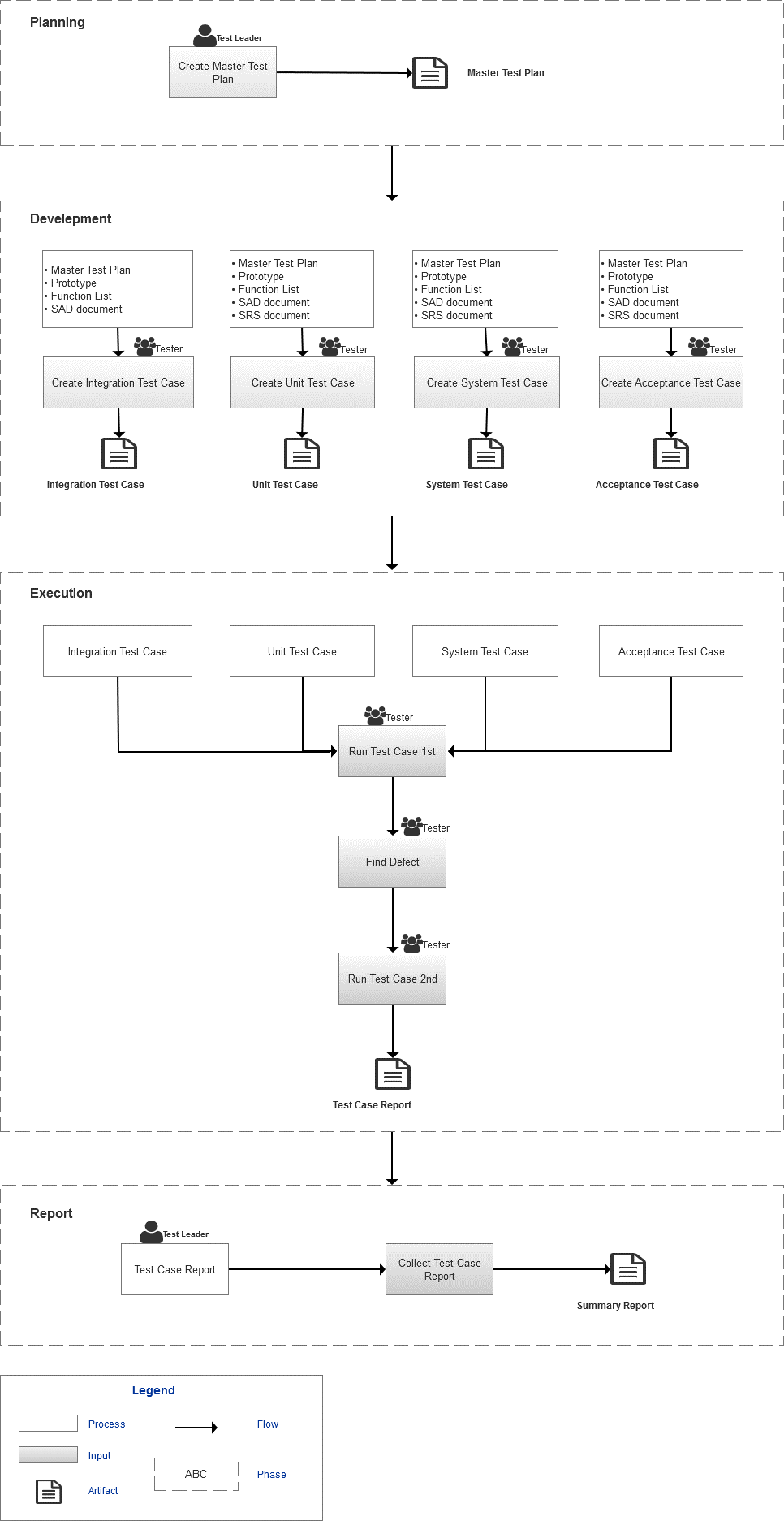


Figure 2. Detail Process

* 1. **Defect report and tracking process**

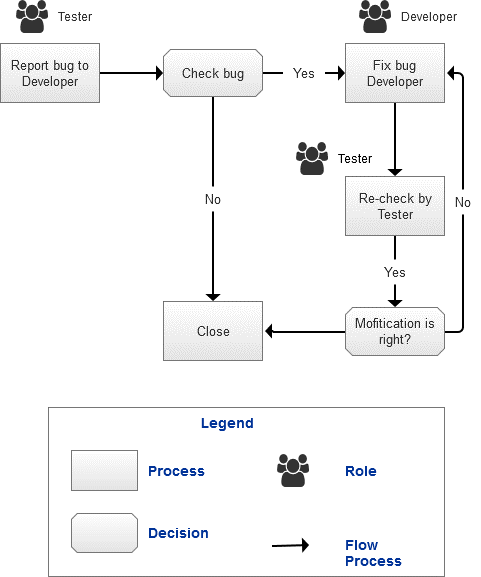


Figure 3. Defect report and tracking process

**Decription:** Tester report bug to developer, they would be stored in Exel file manage. Bug will be started with Open status .

* When the software bug is found, it is started with Open status. Tester would like to note it in Defect report Excel file and transfers to implementation team. After developers check and fix (if they are bugs), developers would re-transfer to testers.
* When the software bugs are fixed and re-transferred, testers would like to re-test. The status changes to Fix status. When test case is passed, the status changes to Close/ Resolved status.

Defect Report use Severity & Priority state to determine when those software bugs need to be fixed.

|  |  |
| --- | --- |
| **Severity** | |
| Critical | Defect causes system crash, data loss, application can’t install, causes can’t continue execute testing. |
| High | The main function of product doesn’t work. |
| Medium | The sub function, non-critical functionality of product doesn’t work. |
| Low | Defect contains typing mistake, unclear wording or error messages in low visibility fields. |

Table 1.Severity

|  |  |
| --- | --- |
| **Priority** | |
| High | The defect must be resolved as soon as possible because the defect is affecting the application or the product severely. The system cannot be used until the repair has been done. |
| Medium | The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created. |
| Low | The defect should be repaired, but repair can be deferred until after more serious defect have been fixed. |

Table 2. Priority

1. **Test Criteria**
   1. **Test Entry/ Exit Criteria**
      1. **Test Entry Criteria**

All inspection preconditions are met as specified in the inspection policies, procedures, and plans.

* + 1. **Test Exit Criteria**

All identified problems have been resolved and re-inspected by the group or moderator.

* 1. **Suspension/ Resumption Criteria**

Testing will be suspended if:

* A failure of severity critical and high is observed.
* A hardware failure occurs.

Testing will be resume when defect causing a software failure is repaired or hardware failure have been repaired

1. **Test Strategy**
   1. **Review Document**

|  |  |
| --- | --- |
| **Operation** | **Objectives** |
| Planning | * Materials to be inspection must meet inspection entry criteria. * Arrange the availability of the right participants. * Arrange suitable meeting place and time. |
| Overview | Group education of participants in what is to be inspected.  Assign inspection roles to participants. |
| Preparation | Participants learn th material and prepareto fulfill their assigned roles. |
| Inspection | Find defect.(Solution hunting and discussion of design alternatives is discouraged) |
| Re-work | The author rework all defect |
| Follow-up | Verification by the inspection mod- erator or the entrie inspection team to assure that all fixes are effective and that no secondary defect have been introduced. |

Table 3. Review Document

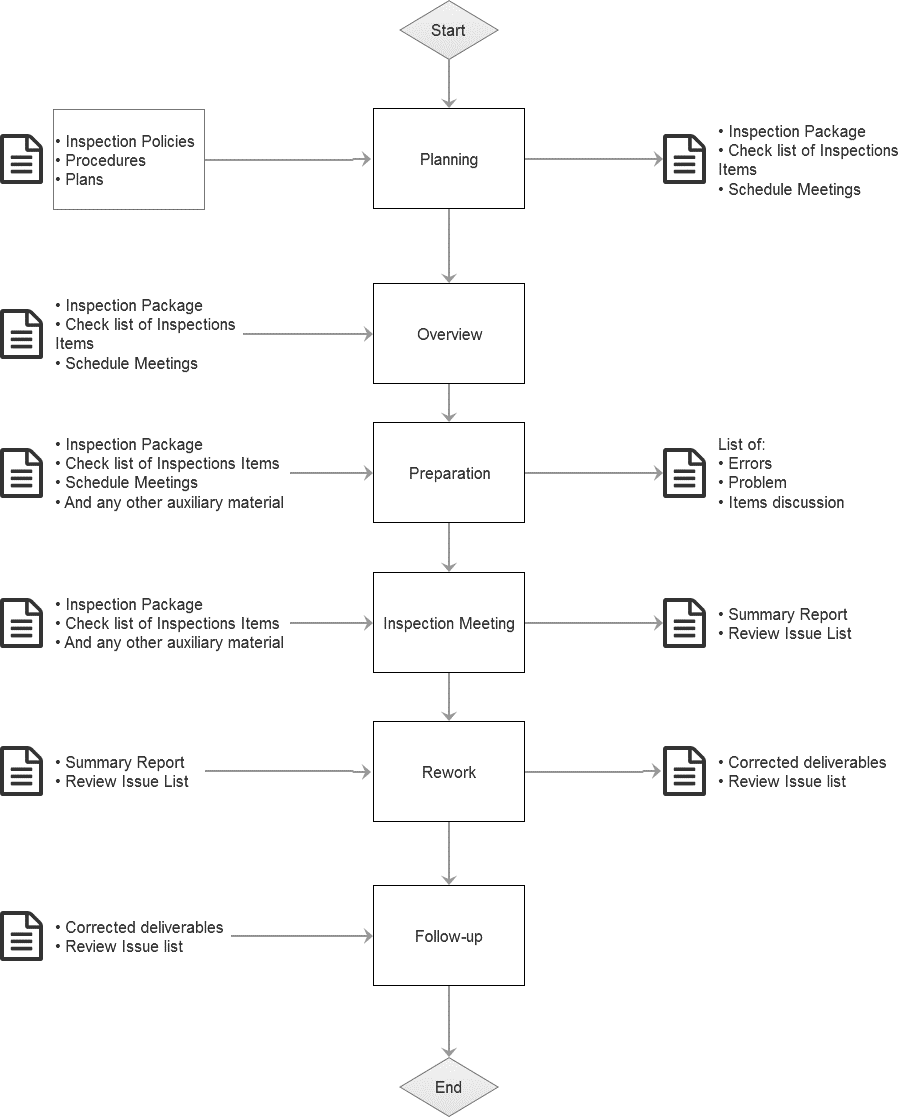


Figure 4. Review Document

* 1. **Unit Testing**

Developer will perform development of unit test scripts and check test scripts before they implement functions (Unit code).

* Develop test cases & test data.
* Develop test script.
* Execute test script has been developed.
* Develop Unit code to prove test script is correct.
* Check Unit code with test script has been developed.
* Repair Unit code if test script run incorrectly.
* When test script run correctly is developer refactor Unit code.
* Develop test script for next Unit code.

When develop Unit test cases & test data need to apply White box testing techniques, such as:

* Branch testing
* Loop testing
* Just use two methods to implement because it meets the test's outcomes and it check almost cases
  1. **System Testing**
     1. **Description**

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system.

* + 1. **Types**

**System testing:** All functional Requirements as described in Software Requirement Specification Document (SRS) will be evaluated. The following black box test design techniques will be used.and boundary – value analysis (ECP/BVA).

**Security testing:** is focused on the verification that the software is protected from external attacks. In particular, security testing verifies the confidentiality, integrity, and availability of the systems and its data. Usually, security testing includes verification against misuse and abuse of the software or system (negative testing).

**Usability testing:** In usability testing basically the testers tests the ease with which the user interfaces can be used. It tests that whether the application or the product built is user-friendly or not.

Performance testing: Performance testing is testing that is performed, to determine how fast some aspect of a system performs under a particular workload. It can serve different purposes like it can demonstrate that the system meets performance criteria.

**Scalability testing:** It is the testing of a software application for measuring its capability to scale up in terms of any of its non-functional capability like load supported, the number of transactions, the data volume etc.

* 1. **Integration Testing**
     1. **Description**

Integration testing is the process of verifying the interactions among software components. Classicalintegration testing strategies, such as topdown and bottom-up, are often used with hierarchically structured software.

* + 1. **Types**

**Big Bang approach:** In this approach, most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing

* 1. **Acceptance Testing**
     1. **Description**

User Acceptance Testing is performed by customer to compare with entry requirement. This is the final testing. To develop Acceptance testing, customer need to apply Black box testing techniques. Detail Acceptance testing will be described in Acceptance Test Plan document.

* + 1. **Types**

**Alpha Test:** It is done by internal staff before it is released to external testers or customers and takes place in the development environment. The feedback – collected from those alpha testers – then is used to fix certain issues or bugs and improve the usability of the product.

* 1. **Automation Testing**
     1. **Deccription**

Automation testing is used for verify functional, Before that, it has been tested by writing test script and auto run test script.

Automation testing also used to perform stress test and load test.

The objective of automation testing is to simplify as much of the testing effort as possible.

Test script is written by tester when test cases are developed before that, tester selects test case to be transferred to test script

* + 1. **Types of function to be automated**

1. **Feature to be test**

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

Table 4. Feature to be test

1. **Feature not to be test**

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

Table 5. Feture not to be test

1. **Milestones and Schedule**
   1. **Milestones**

|  |  |  |
| --- | --- | --- |
| **Task** | **Milestones Date** | **Actual Date** |
| Training test |  |  |
| Master Test Plan |  |  |
| Finish testing Incremental 1 (Inbound + Configuration) |  |  |
| Finish testing Incremental 2 (Outbound + Execution) |  |  |
| Finish testing Incremental 3 (Report + Utility) |  |  |

Table 6. Milestones

* 1. **Schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Assigned to** | **Start** | **Finish** |
| Create Master Test Plan |  |  |  |
| Training test |  |  |  |
| Develop Integration test cases - Incremental 1 |  |  |  |
| Develop System test cases - Incremental 1 |  |  |  |
| Develop User Acceptance test cases - Incremental 1 |  |  |  |
| Develop Unit test cases - Incremental 1 |  |  |  |
| Run Integration Test cases - Incremental 1 |  |  |  |
| Run System test cases - Incremental 1 |  |  |  |
| Run User Acceptance test cases - Incremental 1 |  |  |  |
| Run Unit test cases - Incremental 1 |  |  |  |
| Develop Integration test cases - Incremental 2 |  |  |  |
| Develop System test cases - Incremental 2 |  |  |  |
| Develop User Acceptance test cases - Incremental 2 |  |  |  |
| Develop Unit test cases - Incremental 2 |  |  |  |
| Run Integration Test cases - Incremental 2 |  |  |  |
| Run System test cases - Incremental 2 |  |  |  |
| Run User Acceptance test cases - Incremental 2 |  |  |  |
| Run Unit test cases - Incremental 2 |  |  |  |
| Run Integration Test cases - Incremental 2 + 1 |  |  |  |
| Develop Integration test cases - Incremental 3 |  |  |  |
| Develop System test cases - Incremental 3 |  |  |  |
| Develop Unit test cases – Incremental 3 |  |  |  |
| Develop User Acceptance test cases - Incremental 3 |  |  |  |
| Run Integration Test cases - Incremental 3 |  |  |  |
| Run Integration Test cases - Incremental 3 + 2 + 1 |  |  |  |
| Run System test cases - Incremental 3 |  |  |  |
| Run User Acceptance test cases - Incremental 3 |  |  |  |
| Run Unit test cases – Incremental 3 |  |  |  |

Table 7. Schedule

1. **Risk and Mitigation**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Risk** | **Severity** | **Mitigation plan** |
|  |  |  |  |
|  |  |  |  |

Table 8. Risk and Mitigation

1. **Measure and Metrics**

|  |  |
| --- | --- |
| **Percent of test cases Pass (PTCP)** | |
| Definition |  |
| Goals |  |
| Analysis Procedure |  |
| Metric |  |

Table 9. Percent of test case Pass

|  |  |
| --- | --- |
| **Percent of test cases Fail (PTCF)** | |
| Definition |  |
| Goals |  |
| Analysis Procedure |  |
| Metric |  |

Table 10. Percent of test case Fail

|  |  |
| --- | --- |
| **Defects by Severity (DS)** | |
| Definition |  |
| Goals |  |
| Analysis Procedure |  |
| Metric |  |

Table 11. Defect by Severity

|  |  |
| --- | --- |
| **Percentage test cases executed (PTCE)** | |
| Definition |  |
| Goals |  |
| Analysis Procedure |  |
| Metric |  |

Table 12. Percentage test case executed

|  |  |
| --- | --- |
| **Percentage test cases not executed (PTCNE)** | |
| Definition |  |
| Goals |  |
| Analysis Procedure |  |
| Metric |  |

Table 13. Percentage test case not executed

1. **Human Resources Participate Test**
   1. **Tester Information**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Name** | **Team** | **Role** | **Email** |
| 1. | Quyen Nguyen | FTeam | Test Leader | quyennguyen5094@gmail.com |
|  |  |  |  |  |

Table 14. Test Information

* 1. **Roles and Responsibilities**

|  |  |  |
| --- | --- | --- |
| **No.** | **Role** | **Responsibilities** |
| 1. | Test Leader | * Provide technical direction * Design Test Plan * Prepare summary Report * Create Test Plan |
| 2. | Tester | * Test project document * Developer test case * Run Test Case * Tracking defect |
| 3. | Developer | * Attend in meeting as defect report * Write unit test, perform unit test |
| 4. | Project Manager | * Review Test plan * Review Test schedule * Attend test report meeting |
| 5. | Customer | * Perform User Acceptance testing |

Table 15. Roles and Respondibilities

* 1. **Structure of Meeting**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Description** | **Frequency** | **Participant** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table 16. Structure of Meeting

* 1. **Training Schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Description** | **Beginning Date** | **Ending Date** | **Member** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table 17. Training Schedule

1. **Test Environments**
   1. **Hardware**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Minimum configuration requirements** |
|  |  |  |
|  |  |  |
|  |  |  |

Table 18. Hardware

* 1. **Software**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Software** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table 19. Software

* 1. **Test Tool**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Tool** |
| 1 | To execute automation test | Selenium |

Table 20. Test Tool

1. **Deliverable and Glossary**
   1. **Deliverable**

|  |  |
| --- | --- |
| **No.** | **Document** |
| 1 | [FIS\_MasterTestPlan\_##](../Template/FIS_TE_TestPlan_Template.docx) |
| 2 | FIS\_UnitTestCase\_## |
| 3 | FIS\_SystemTestCase\_## |
| 4 | FIS\_IntegrationTestCase\_## |
| 5 | FIS\_AcceptanceTestCase\_## |
| 6 | FIS\_DefectLog\_## |
| 7 | FIS\_TestReport\_## |

Table 21. Deliverable

* 1. **Glossary**

|  |  |  |
| --- | --- | --- |
| **No.** | **Glossary** | **Description** |
| 1 | SRS | Software Requirement Specification |
| 2 | PTCP | Percent of test cases Pass |
| 3 | PTCF | Percent of test cases Fail |
| 4 | DS | Defects by Severity |
| 5 | PTCE | Percentage test cases executed |
| 6 | PTCNE | Percentage test cases not execute |
| 7 | SAD | Software Architect & Design |

Table 22. Glossary