Software Testing Plan

HIT Team

Version 1.0

# Table of Contents

[Table of Contents ii](#_Toc331513752)

[1. Introduction 4](#_Toc331513753)

[1.1. Document purpose 4](#_Toc331513754)

[1.2. Project Scope 4](#_Toc331513755)

[1.3. Test Phases 4](#_Toc331513756)

[1.3.1. Unit Testing 4](#_Toc331513757)

[1.3.2. System testing 5](#_Toc331513758)

[1.4. Test results 6](#_Toc331513759)

[1.4.1. Unit testing 6](#_Toc331513760)

[1.4.2. System testing 6](#_Toc331513761)

[2. Schedule 8](#_Toc331513762)

[3. Human resources in testing team 9](#_Toc331513763)

[3.1. Information about test team 9](#_Toc331513764)

[3.2. Roles & Responsibilities 9](#_Toc331513765)

[4. Test Environment 10](#_Toc331513766)

[4.1. Hardware requirement 10](#_Toc331513768)

[4.2. Software requirement 10](#_Toc331513769)

[4.3. Test Documents 10](#_Toc331513770)

[5. Entrance criteria and exit criteria 11](#_Toc331513771)

[5.1. Entrance criteria 11](#_Toc331513772)

[5.1.1. Unit testing 11](#_Toc331513773)

[5.1.2. System testing 11](#_Toc331513774)

[5.1.3. Exit criteria 11](#_Toc331513775)

[6. Appendix: Definitions 13](#_Toc331513776)

Information of document

|  |  |
| --- | --- |
| Title | Architecture Driver Document |
| Author(s) | All team |
| Reviewer(s) | All team |
| Team name | HIT Team |
| Team members | Thanh Giang, Hiep Ta, Phuc Nguyen, Giang Nguyen, Dat Tran, Huy Huynh |
| Project mentor | Mr. Tong Hung Anh |
| Project customer | Mr. Bui Minh Phung |
| Editor | Hiep Ta |
| Type of report | Software Testing |
| Software used | MS Word, MS Excel |

Document Reviewer Information

|  |  |  |
| --- | --- | --- |
| **Reviewer Name** | **Review Attendance (R/S)** | **Comments** |
|  |  |  |
|  |  |  |
|  |  |  |

Document Approver Information

|  |  |  |
| --- | --- | --- |
| **Approver Name** | **Approver Function** | **Comments** |
|  |  |  |
|  |  |  |
|  |  |  |

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Date of Issue | Author(s) | Brief Description of Change |
| 1.0 | 02/07/2012 | Hiep Ta | Create Test plan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Document purpose

This document detailed description about plan and testing implement method for SRM project. At the same time the requirement for personnel, tools and materials needed for testing is also defined here.

## Project Scope

Documents describing strategy & plan when implement testing phase in SRM project. The detailed description of test cases in each phase will be implemented in the test specification documents.

This document is used by PM, Tech leader who participated in the SRM project.

## Test Phases

### Unit Testing

* Developer will perform development Unit test scripts and test scripts well before begin development functions (Unit code).
* Applied strictly Test driven development methods in the order as below:
  + Develop test cases & test data.
    - Create 3 test case for Unit test is: create new account, change password and show report
  + Develop test script.
    - Write 3 test script for 3 test case described
  + Implement developed test script.
  + Develop Unit code to prove test script is right.
  + Check test script with developed Unit code have developed.
  + Edit Unit code if script run fail.
  + When script run right, conduct refactoring Unit code.
  + Conduct develops test script for next Unit code.
* When develop Unit test cases & test data should apply White box testing techniques like:
  + Basic path testing.
  + Branch testing.
  + Loop testing.

### System testing

#### Functional testing

* Create 3 test case for System test is: view new account, change password and show bill
* Use cases which describes functionalities in SRS must be fully show in System test specification
* Graphic User Interface (GUI) of each screen in GUI Specification must be fully showed in System test specification.
* When develop Functional test cases & test data, team decided apply Black box testing techniques like:
* Boundary Testing: test bounds of input value
* Error Guess Testing: depend on team member’s experiment to make a identify about that bug and make a test case

#### Quality attribute testing

* All quality attribute of SRM will be test: usability, performance, integrity, security
* When develop quality attribute test cases & test data should apply Black box testing techniques like:
* Stress testing
* Performance testing
* Usability testing

|  |  |  |  |
| --- | --- | --- | --- |
| No | Quality Attribute | Exit criteria | Priority |
| 1 | Usability | Pass 80% | 1 |
| 2 | Performance | Pass 100% | 2 |
| 3 | Integrity | Pass 100% | 1 |
| 4 | Security | Pass 100% | 2 |

## Test results

All test results must be saved in Configuration management system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Test case | Quantity | Pass | False | Coverage |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |

*Test case coverage*

### Unit testing

* Unit code must be clear, comment completely.
* Unit test cases and Unit test reports must be complete.
* Test scripts must satisfy requirements below:
  + Can be auto implement and reuse.
  + Easy to develop.
  + Develop in one time and use in future.
  + Can be implement by any object. (usability)
  + Implement in one push button.
  + Best perform when implemented.

### System testing

* Quality attributes in SRS must be full test.
* Functionalities are described on use cases in SRS must be full test.
* System test cases and test reports must be clear and complete. These artifacts will be updated continuous in project life cycle.
* Bug log report and bug list remain.
* Bug status and bug lifecycle are continuous reported.

# Schedule

Test plan for the SRM will be established and implemented for testing and System Integration testing. Training plan and make Unit testing will be specifically described in the Master Plan.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Action | | Describe | Start date | | Finish date | | Human Resource |
| Test documentation | | | | | | | |
| Test Plan |  | | | 02/07/2012 | | 10/07/2012 | Hiep Ta |
| Design Unit Test |  | | |  | |  | Huy Huynh,  Giang Nguyen |
| Design System Test |  | | |  | |  | Huy Huynh,  Giang Nguyen |
| Unit testing | | | | | | | |
| Unit test |  | | |  | |  | Huy Huynh,  Giang Nguyen |
| Report |  | | |  | |  | Huy Huynh,  Giang Nguyen |
| System testing | | | | | | | |
| System test |  | | |  | |  | Huy Huynh,  Giang Nguyen |
| Report |  | | |  | |  | Huy Huynh,  Giang Nguyen |

#### Figure 1: Implement plan

# Human resources in testing team

## Information about test team

The test team will consist of:

* Test leader: Hiep Ta
* Tester: Huy Huynh, Giang Nguyen
* Document writer: Thanh Giang, Hiep Ta

## Roles & Responsibilities

* Test leader: Ensures the overall success of the test cycles. He/she will coordinate weekly meetings and will communicate the testing status to the project team. Reports directly to the PM.
* Tester: Responsible for performing the actual system testing following test documents. Discussions with developers about bugs fix. Reports directly to the test leader
* Document writer: Responsible for developing test documents (Test plan, test cases, test report)

# Test Environment



## Hardware requirement

|  |  |  |
| --- | --- | --- |
| No. | Describe | Minimum |
| 1 | Computer | * CPU: Core 2 Duo or higher * RAM 2Gb or higher * Hard Disk space: 05 Gb free |
| 2 | Printer | * Cable * API integrate |

#### Figure 3: Hardware

## Software requirement

|  |  |  |
| --- | --- | --- |
| No. | Describe | Software |
| 1 | Web Server / Application server | Visual Studio 2010  SQL server 2005 or above  MVC .NET Framework 3.0 |
| 2 | Web browser | Chrome, Firefox |

#### Figure 4: Software

## Test Documents

|  |  |  |
| --- | --- | --- |
| STT | Artifact’s name | Describes |
| 1 | SRM Test Plan | This document describes Test strategy & plan when testing SRM project |
| 2 | SRM System Test Specification | This document describes test cases for System test phase when implement testing SRM project |
| 3 | SRM Unit Test Specification | This document describes Unit Test scenarios for each Unit in SRM. |

#### Figure 5: Test Documents

# Entrance criteria and exit criteria

## Entrance criteria

These criteria should be defined before the test began and was carried out by Test leader.

### Unit testing

* Unit Test specification must be implemented fully
* The Unit test scripts must be built into the specification Unit test.
* The production code must be developed by the method of TDD.
* The production code should be reviewed.

### System testing

* System test plan must be approved.
* System Test Cases must be ready.
* Personnel must be ready.
* Environment and tools must be available to test.
* The test material must be reviewed documents

### Exit criteria

**Unit testing**

* Code reviewed (peer review, desk check)
* 100% Branch coverage
* 100% Condition coverage (which may be relaxed, depending on language)
* Any problems remaining unresolved from Unit Test been documented and explained why they are unresolved.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Function | Exit criteria | Priority | Severity | Beginning status | Ending status |
| Check account | Pass 100% | 1 | 1 | New | Close |
| Change password | Pass 100% | 2 | 3 | New | Close |
| Check items | Pass 100% | 1 | 1 | New | Close |

**System testing**

* All System Test Cases to be done.
* Some pre-defined number of defects discovered
* Documented list of issues and bugs remaining.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Function | Exit criteria | Priority | Severity | Beginning status | Ending status |
| View account | Pass 100% | 2 | 2 | New | Close |
| Change password | Pass 100% | 2 | 3 | New | Close |
| Show bill | Pass 100% | 1 | 1 | New | Close |

# Appendix: Definitions

|  |  |
| --- | --- |
| Priority | Describe |
| 1 | Test cases very important and can’t fail. Must fix immediately if test case fail |
| 2 | Test cases very important and can’t fail. Must fix soon if test case fail |
| 3 | Test cases not important and maybe perform or not, depend on the test plant. |
| 4 | Test cases not important and not necessary to perform, but when perform it help improve program quality |

#### Figure 6: Test case priority describe

In the process of testing, the bugs will be managed by tools such as Excel, Mantis. The bug will be classified and assigned to the developer to fix.