**Van Lang University**

11

**Test Plan**

**Team Assignment 03**

**TEAM 02**



TEST PLAN DESCRIBLE DOCUMENT

TEST PLAN – Restaurant Management System

Version 0.2 – 12/05/2011

This document describe test plan for Restaurant Management System project

**Test Plan Reviewer**

|  |  |  |  |
| --- | --- | --- | --- |
| Title | Reviewer | Review date | Content |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |

**Test Plan Approver**

|  |  |  |  |
| --- | --- | --- | --- |
| Title | Approver | Approve date | Content |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |

**Test Plan Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Date | Revision | Content | Editor |
| 1 | 06/05/2011 | 0.1 | Test Environment | Hiệp Tạ |
| 2 | 06/05/2011 | 0.1 | Entrance and Exit criteria | Tuấn Lại |
| 3 | 06/05/2011 | 0.1 | Human resource | Đạt Trần |
| 4 | 06/05/2011 | 0.1 | Appendix | Đạt Trần |
| 5 | 06/05/2011 | 0.1 | Introduction | Thanh Giang |
| 6 | 06/05/2011 | 0.1 | Schedule | Thanh Giang |
| 7 | 11/05/2011 | 0.2 |  |  |
| 8 | 11/05/2011 | 0.2 |  |  |

TABLE OF CONTENTS

1. INTRODUCTION 6

1.2. Project Scope 6

1.3. Test Phases 6

1.3.1. Unit Testing 6

1.3.2. System testing 7

1.3.2.1. Functional testing 7

1.3.2.2. Quality attribute testing 7

1.4. Test results 8

1.4.1. Unit testing 8

1.4.2. System testing 8

2. SCHEDULE 10

3. HUMAN RESOURCES IN TEST TEAM 11

3.1. Information about test team 11

3.2. Tranning plan 11

3.3. Roles & Responsibilities 11

4. TEST ENVIRONMENT 12

4.1. Hardware requirement 12

Figure 3: Hardware 12

4.2. Software requirement 12

Figure 4: Software 12

4.3. Test tools 12

4.4. Test databases 13

All test result must be save in Configuration management system. 13

4.5. Test Documents 13

**Figure List**

**Figure 1 Implement plan 8**

**Figure 2 Training plan 10**

**Figure 3 Hardware requirement 11**

**Figure 4 Software requirement 11**

**Figure 5 Test documents 12**

**Figure 6 Testcase priority descibe 15**

**Figure 7 Bug describe 15**

# INTRODUCTION

* 1. **Document purpose**

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

* 1. **Project Scope**

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

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

* 1. **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 testcases & 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 develop test script for next Unit code.
* When develop Unit testcases & 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 show in System test specification.
* When develop Functional testcases & 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 RMS will be test: usability, performance, integrity, security
* When develop quality attribute testcases & 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 testcases 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 testcases 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 RMS 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 | Write test plan | 7/5/2011 | 12/5/2011 | All member |
| Design Unit Test |  | 13/5/2011 | 16/5/2011 | Anh Tuấn , Dũng Đạt, Chấn Huy |
| Design System Test |  | 18/5/2011 | 20/5/2011 | Anh Tuấn , Dũng Đạt, Chấn Huy |
| Unit testing (In-house testing) | | | | |
|  | | | | |
| Unit test |  | 24/5/2011 | 27/05/2011 | Anh Tuấn , Dũng Đạt, Chấn Huy |
| Report |  | 28/5/2011 | 28/5/2011 | Hồng Phúc , Hà Thanh |
| System testing (In-house testing) | | | | |
|  |  |  |  |  |
| System test |  | 2/6/2011 | 4/6/2011 | Anh Tuấn , Dũng Đạt , Chấn Huy |
| Report |  | 5/6/2011 | 5/6/2011 | Hồng Phúc , Hà Thanh |

**Figure 1: Implement plan**

# HUMAN RESOURCES IN TEST TEAM

## Information about test team

The test team will consist of:

* Test leader: Hiệp Tạ
* Tester: Tuấn Lại, Huy Huỳnh, Đạt Trần
* Document writer: Thanh Giang, Phúc Nguyễn

## Tranning plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Describe | Start date | End date | Human resource | Guider |
| 1 | Train working environment on TFS | 10/05/11 | 10/05/11 | All team | Quoc Nguyen |
| 2 | Train programming techniques (5 weeks) | 09/05/11 | 09/06/11 | All team | Nam Vu |

**Figure 2: Training Plan**

## 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

Depend on average

|  |  |  |
| --- | --- | --- |
| STT | Describe | Minimum |
| 1 | Computer | * CPU: Core 2 Duo 2.4 * RAM 2.0GB * Screen: SVGA Color Monitor - resolution 1024x768 * Hard Disk: 10GB free space |
| 2 | Printer | * Cable * API integrate |

## *Figure 3: Hardware*

## Software requirement

|  |  |  |
| --- | --- | --- |
| STT | Describe | Software |
| 1 | Web Server / Application server | .Net framework 4.0  Visual Studio 2010  SQL server 2008  OLEDB for FoxPro |
| 2 | Work station | IE8.x , Firefox |

## *Figure 4: Software*

## Test tools

Predict:

* + - Use Visual studio 2010 to implements unit test, system test scripts.
    - Use TFS (Team foundation server) 2010 to manages test result and fix bugs.

## Test databases

## All test result must be save in Configuration management system.

## Test Documents

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

## *Figure 5: Test Documents*

# ENTRANCE CRITERIA AND EXIT CRITERIA

* 1. **Entrance criteria**

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

* + 1. Unit testing
* UT specification must be implemented fully..
* The UT scripts must be built into the specification UT.
* The production code must be developed by the method of TDD.
* The production code should be reviewed.
  + 1. 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
  + 1. Exit criteria
    2. 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 |

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

## *Figure 6: Test case priority describe table*

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.

|  |  |
| --- | --- |
| Bug severity | Describe |
| 1 | Bug very important , must fix immediately because it made the system crash or loss data .Bug because testcase Priority 1 fail. |
| 2 | Bug very important , must fix soon because it made critical defect such as wrong features and system processor or system crashes in some. Bug because testcase Priority 2 fail. |
| 3 | Small bug need to fix as soon as possible because it can made the defect of system processor or made difficult for user. Bug because testcase Priority 3 fail |
| 4 | Features advanced should make and maybe because testcase Priority 1 fail |
| Bug status | **Describe** |
| New | New Bug |
| Feedback | Need more information about bug |
| Acknowledged | Bug had been notice but not confirmed or assigned |
| Confirmed | Confirmed and can reconstruct bug |
| Assigned | Had been assign to developer |
| Resolved | Bug had been fix and waiting for Confirmation |
| Closed | Close bug |

## *Figure 7: Bug describe*