|  |
| --- |
| Meeting Room Book System – Team1 |
| Software Testing Plan |
| Version 1.3 |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Content |
| Huyen Le | 07/15/2012 | Create document | 1.0 |
| Huyen Le | 07/16/2012 | Update information | 1.1 |
| Huyen Le | 07/17/2012 | Update information | 1.2 |
| Huyen Le | 07/20/2012 | Update detail | 1.3 |

Table of Contents

[1 Introduction 5](#_Toc331799080)

[1.1 Purpose 5](#_Toc331799081)

[1.2 Scope 5](#_Toc331799082)

[1.3 Project Overview 5](#_Toc331799083)

[1.4 References 5](#_Toc331799084)

[2 Test Strategy 5](#_Toc331799085)

[2.1 Unit Test 6](#_Toc331799086)

[2.2 Integration Test 6](#_Toc331799087)

[2.3 System test 6](#_Toc331799088)

[2.4 Acceptance Test 6](#_Toc331799089)

[3 Test Environment 7](#_Toc331799090)

[3.1 Hardware: 7](#_Toc331799091)

[3.2 Software: 7](#_Toc331799092)

[4 Entrance and Exit Criteria 7](#_Toc331799093)

[4.1 Entrance Criteria 7](#_Toc331799094)

[4.2 Exit Criteria 7](#_Toc331799095)

[5 Test Schedule 7](#_Toc331799096)

[6 Testing Process 8](#_Toc331799097)

[6.1 Roles and Responsibility 8](#_Toc331799098)

[6.2 Testing Process Activity: 9](#_Toc331799099)

[6.3 Measurement 10](#_Toc331799100)

[7 Test Deliverables 10](#_Toc331799101)

[8 Risk 11](#_Toc331799102)

[8.1 Schedule 11](#_Toc331799103)

[8.2 Technical 11](#_Toc331799104)

[8.3 Management 11](#_Toc331799105)

[8.4 Personal 11](#_Toc331799106)

[8.5 Requirements 11](#_Toc331799107)

LIST OF TABLE

[Table 1: Reference Document 5](#_Toc331798449)

[Table 2: Test Strategy 6](#_Toc331798450)

# Introduction

## Purpose

The purpose of this document is to describe testing activities and strategies during project lifecycle. Test manager will ensure that software testing is performed according to the process defined in test plan and project plan

## Scope

Scope of test plan is to help test members to do tasks following test plan and schedule that are defined and approved. This document describes strategy for Unit test, Integration test and acceptance test. It is also manifest the processes of reviewing and evaluation

## Project Overview

The system is developing by my team is a module of overall system that will allow for further grow and provide additional features. This module allow user to manage meeting room booking, managing the clashes of control needs just in time to register, places. Management priorities for the registration of identical objects and statistics situation of the use

## References

|  |  |
| --- | --- |
| No | Document Name |
| 1 | MRBS\_AcceptanceTest |
| 2 | MRBS\_IntegrationTest |
| 3 | MRBS\_SystemTest |
| 4 | MRBS\_TestReport |
| 5 | MRBS\_SoftwareTestingPlan |

Table 1: Reference Document

# Test Strategy

Our quality assurance strategy is to define goals for each testing stage. The quality assurance techniques are chosen to satisfy each goal. For some goals, two or more approaches are used to complement one another

|  |  |  |
| --- | --- | --- |
| **Quality Assurance Techniques** | | **Supported tools** |
| Testing | Unit Testing | Visual studio 2010 |
| Integration Testing | Test Manager 2010 |
| System Testing | Test Manager 2010 |
| User Acceptance Testing | Test Manager 2010 |
| Reviewing | Requirement Review | Check list document |
| Architecture Evaluation | Check list document |
| Detailed Design Review | Check list document |
| Code Review | Unit test report |

Table 2: Test Strategy

## Unit Test

The purpose of unit testing is to cover:

* Correct functionality: Validate that the code does what it is supposed to
* Boundary value analysis: Check the boundary conditions of n-1, n and n+1 especially for loops like *for*, *while* etc.
* Exception Handling: Verify the recording of all detected and reportable errors that the program is designed to find and report. Some examples could be null pointer exceptions, array index out of bound exceptions and so on
* Mathematical limit checking: Check the out of range values that could cause a mathematical function to calculate erroneous results

Approach: Unit testing is the responsibility of each individual team member. The team will follow the unit testing process

## Integration Test

Objective: the integration testing is meant to ensure that all modules within a release version of the system are inter-operating. Integration testing will focus on testing the interfaces between code units, components, and subsystems. Integration Testing will be conducted at the end of each of incremental iterations.

Approach: all code units will undergo integration testing. The team will follow the integration testing process

## System test

Objective: To demonstrate that system works and obtain signoff by the client that system passed system tests. The purpose of these tests is to confirm that the system is developed according to the specified user requirements and is ready for operational use.

Approach: The System test procedure will be created to cover Use Case requirements written in the Specification Requirement.

## Acceptance Test

Objective: To demonstrate that system works and obtain signoff by the client that system passed acceptance tests. The purpose of these tests is to confirm that the system is developed according to the specified user requirements and is ready for operational use.

Approach: The Acceptance test procedure will be created to cover the functional and quality attribute requirements written in the architecture driver’s specification.

# Test Environment

## Hardware:

Laptop: 2 machines.

Barcode scanner: Support API - There is a connection cable

Printer: Support API - There is a connection cable

## Software:

Computer of software developer team and testing team

MS Visual Studio 2010 Ultimate

MS SQL server 2008

MS Office 2010

Net framework 4.0

TFS - Team Foundation Server and other tools

Computer of customer

Configuration: window 7 professional, setup web browser (Firefox; Chrome; IE).

CPU: Core Duo - RAM 1.0 GB - Hard Disk: 5 GB free

# Entrance and Exit Criteria

## Entrance Criteria

* All source codes are components to be tested
* All QA resources has enough functional knowledge
* Hardware and software tools are available
* Test plans and test cases are reviewed and signed off

## Exit Criteria

* Run all Test Cases
* Pass 75% Test Cases
* Planned deliverables are ready
* Critical or High severity defects are fixed
* Less than 5 medium severity defects
* Less than 7 low severity defects

# Test Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Description | Beginning Date | Ending Date | Human Resources |
| Test documentation | | | | |
| 1 | Test Plan | 15/07/2012 | 22/07/2012 | Huyen Le |
| 2 | System Test Specification | 22/07/2012 | 27/07/2012 | Huyen Le |
| 3 | Acceptant Test Specification | 27/07/2012 | 01/08/2012 | Huyen Le |
| 4 | Review Test Plan, System Test Specification, Acceptant Test Specification | 01/08/2012 | 03/08/2012 | Team1\_K15T1 |
| Unit testing | | | | |
| 1 | Execute Unit Testing | 22/07/2012 | 10/08/2012 | Team1\_K15T1 |
| System testing | | | | |
| 1 | Execute System Testing | 27/07/2012 | 10/08/2012 | Team1\_K15T1 |
| 2 | Review System Testing | 27/07/2012 | 10/08/2012 | Team1\_K15T1 |
| User acceptance testing | | | | |
| 1 | Demo & Presentation | 05/08/2012 | 10/08/2012 | Team1\_K15T1 |

Table 3: Test Schedule

# Testing Process

## Roles and Responsibility

Test leader: Generate the schedule for performing test cases, monitoring activities of the testers and reporting results directly to project manager. Responsible for instructions, plan and track the progress of the work of testing team.

Tester – Tam Do – Tin Nguyen - Son Dang: Execute test for system under testing specifications and discuss with developers about fix bugs. Test reports directly to the leader.

Document writer – Huyen Le – Van Le: Responsible for the development of test specifications, Perform tasks following the test plan and assignments by test leader (Test cases, test report)

## Testing Process Activity:



Figure 1: Testing Process Activity

|  |  |  |
| --- | --- | --- |
| Input | Testing activities | Output |
| * Project Charter * Project Plan * Source Code * System Build Version * Initial Requirement Document * Requirement specification * Architecture Specification | * Make Test Plan * Make Acceptance Test Case * Make System Test Case * Make Integration Test Case * Acceptance Test * System Test * Integration Test | * Test plan * Acceptance Test Case * System Test Case * Integration Test Case |

Table 4: Summary Critical of Testing Process

|  |  |  |  |
| --- | --- | --- | --- |
| No | Activity | Description | General Output |
| 1 | Make test plan | Leader of Test team develop a plan for test phase | Test Plan Document and Test schedule |
| 2 | Make Acceptance Test Case | Tester develop Acceptance test case (Check List) for product. Required Initial Requirement Document | Acceptance test cases |
| 3 | Make System Test Case | Tester develops functional test case and nonfunctional checklist. Required Requirement Specification. | Functional and nonfunctional test cases |
| 4 | Make Integration Test Case | Tester develop test case for testing that combine between two or more functional. Required Architecture Specification | Integration test cases |
| 5 | Integration Test | Tester execute integration test case. After developer release two relation module (or function) | Integration test report |
| 6 | System Test | Tester execute functional test case and check nonfunctional checklist. After developer release a system draft version | System test report |
| 7 | Acceptance Case | Tester validate product base on acceptance checklist. After release an alpha version. | Acceptance test report |

Table 5: Process Description

## Measurement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Goal | Question | Metric | Source | Responsibility |
| Productivity of tester | How long a tester make a test case | Time/ Test case | Member Time log and TFS | Test Leader |
| How long a tester test a function | Time / function |

Table 6: Measurement

# Test Deliverables

All documents are the responsibility of the project team members. The lists of documents that will be created and maintained under version control and TFS include:

* Test Report – Reports results during test phase including .
* Unit test report – Using it to report results to coders
* Integration test report – describing results after integration testing is finished
* Acceptance test report – describing results after alpha and beta testing stage is finished

# Risk

## Schedule

The schedule for each phase is very strained and could affect testing. A problem in the schedule in one of the other phases could result in a subsequent in the test phase.

## Technical

Since this is a new online store system, in the event of a failure the old system can be used. We will run our test in parallel with the production system so that there is no downtime of the current system.

## Management

Management support is required so when the project falls behind, the test schedule does not get squeezed to make up for the delay. Management can reduce the risk of delays by supporting the test team throughout the testing phase and assigning people to this project with the required skills set

## Personal

Due to the aggressive schedule, it is very important to have experienced testers on this project. Unexpected turnovers can impact the schedule. If attrition does happen, all efforts must be made to replace the experienced individual

## Requirements

Any changes to the requirements could affect the test schedule and will need to be approved by the team members