|  |
| --- |
|  |

VOLUNTARY NON-REMUNERATED BLOOD DONATION

TEST PLAN

Project Code: VNBD

Document Code: VNBD\_Test Plan\_v1.0\_EN

**Ha Noi, 11/09/2013**

Record of change

\*A - Added M - Modified D - Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Effective Date | Changed Items | A\* M, D | Change Description | New Version |
| 11/09/2013 | Add new | A |  | v0.1 |
| 15/09/2013 | Update | M | Update comment of document review | v0.9 |
| 22/09/2013 | Update | M | Update comment of document review | v1.0 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

SIGNATURE PAGE

**ORIGINATOR:** Nguyen Thi Tam 11/09/2013

Test Leader (TL)

**REVIEWERS:** Ngo Tran Linh 15/09/2013

Project Manager (PM)

**APPROVAL:** Bui Dinh Chien 27/10/2013

Supervisor

TABLE OF CONTENTS

[1 INTRODUCTION 5](#_Toc368754412)

[1.1 Purpose 5](#_Toc368754413)

[1.2 Definitions, Acronyms, and Abbreviations 5](#_Toc368754414)

[1.3 References 6](#_Toc368754415)

[1.4 Background information 6](#_Toc368754416)

[1.5 Scope of testing 6](#_Toc368754417)

[1.6 Constraints 8](#_Toc368754418)

[1.7 Risk list 8](#_Toc368754419)

[1.8 Training needs 8](#_Toc368754420)

[2 Requirements for Test 11](#_Toc368754421)

[2.1 Test items 11](#_Toc368754422)

[2.2 Acceptance Test Criteria 11](#_Toc368754423)

[2.3 Feature not to be tested 12](#_Toc368754424)

[3 TEST STRATEGY 13](#_Toc368754425)

[3.1 Test types 13](#_Toc368754426)

[3.1.1 Function Testing 13](#_Toc368754427)

[3.1.2 User Interface Testing 13](#_Toc368754428)

[3.1.3 Data and Database Integrity Testing 14](#_Toc368754429)

[3.1.4 Performance testing 15](#_Toc368754430)

[3.1.4.1 Load Testing 15](#_Toc368754431)

[3.1.4.2 Volume Testing 16](#_Toc368754432)

[3.1.5 Regression Testing 17](#_Toc368754433)

[3.2 Test stages 17](#_Toc368754434)

[4 RESOURCE 19](#_Toc368754435)

[5 Test environment 20](#_Toc368754436)

[5.1 Tools 20](#_Toc368754437)

[5.2 Testing Environment 20](#_Toc368754438)

[6 TEST MILESTONES 21](#_Toc368754439)

[7 DELIVERABLES 22](#_Toc368754440)

# 

# INTRODUCTION

## Purpose

This document shows the testing plan, supports VNBD project. There are 6 sections:

1. Introduction: Overview test processing, contains:

* Purpose
* Definitions, Acronyms, and Abbreviations
* References
* Scope of testing
* Risk list

1. Requirements for Test: Feature to be/ not to be tested.
2. Test strategy: Approach, Stage, States and Tool for testing.
3. Resource: Human software and device for testing.
4. Test milestone: Time for each testing phase.
5. Deliverable: Delivery information of testing document.

## Definitions, Acronyms, and Abbreviations

| Abbreviations | Description | Note |
| --- | --- | --- |
| PM | Project Manager |  |
| QA | Quality Assurance |  |
| SRS | Software Requirement Specification |  |
| TC | Test Case |  |
| KLOC | 1000 line of code |  |

## 

## References

| Title/File name | Author | Version | Effective Date |
| --- | --- | --- | --- |
| SRS | LinhNT01709 | v1.0 | 02/10/2013 |
| Quality Assurance Text Book |  |  | 11/09/2013 |

## Background information

The primary focus of this plan is to help tester can manage test quality in each stage. Guarantee the system can run normally.

It means this website have to:

* Passed the stages of testing: Unit Test, Integration Test/System Test, and User Acceptance Test.
* Passed the types of testing: Interface, Function and Performance is good (do not so slow).
* Run normally in required devices/browsers.

## Scope of testing

“Voluntary non-remunerated blood donation” will be tested by 4 phases:

Phase 1: Unit test

* Unit testing will be done by developers and be approved by team leader and PM.
* When executing unit test case, if any bugs are found, developers have to log and fix it until it is correct.
* The unit test case file will be reviewed and passed by tester and PM of the project.
* Unit testing is also the information provided to tester to perform Integration test and System Test.

Rule for filling test result:

|  |  |
| --- | --- |
| Test result pass | Pass |
| Test result fail | Fail |
| Do not test | Untested |
| Cannot test | N/A (Not available) |

Phase 2: Integration test

* After finishing unit test, integration test will be performed by tester of the project.
* Material is test cases, design, code and test tools.
* Do test by flow of functions and items which have concern each other.
* When executing Integration test case, if any bugs are found, tester have to log to “DefectTrackingLog” file and assign to developer fix it and redo this process until it is correct.

Rule for filling test result:

|  |  |
| --- | --- |
| Test result pass | Pass |
| Test result fail | Fail |
| Do not test | Untested |
| Cannot test | N/A (Not available) |

Phase 3: System test

* After finishing Integration testing and developers collect all functions and items, tester will be performed system test, it means doing test whole system.
* If any bugs are found, developers have to fix and tester will verify them. System test is ended only when test cases are passed and no bug is found.
* For reduce testing time, the system test will be execute as free test.

Phase 4: User Acceptance Test

* Base on requirement specification, system is tested again, for ensure there is not lacking or mistake any requirement.
* If there is any problem, developers have to fix/update and tester will verify them.
* User Acceptance Test is ended only when whole system met requirement specification.

## Constraints

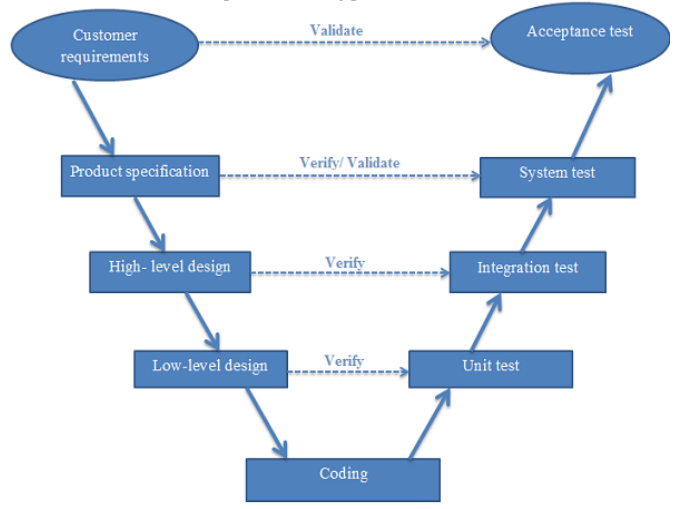
* Now, there are a lot of environments that tester must cover. In this project, there is one tester so tester can’t cover all environments.
* Constraints on schedules because tester doesn’t have experience about take plan test for software project.

## Risk list

* Lack of experience about web-service -> Study.
* Deployment, test environment -> Provide.
* Performance test: Cannot test the case which so many users connect to website.
* Change request: Proposed changes will be discussed with all members of the project. The project leader will determine the impact of the change. After that, it will be implemented.

## Training needs

“Voluntary Non-remunerated Blood Donation” project follows V-Model process, so we define or phases of testing associated with phases of development life-cycle.



**Figure 1: V-Model**

Due to requirement of project must always suitable with user, we choose V-Model to implement testing process. With V-Model, software development is separated into two appropriate phase’s group: development and testing. In this model, the verification and validate will be done side by side. It emphasizes the strict process flow to develop a quality product. The errors occurred in any phase will be corrected in that phase. Proactive defect tracking defects are found at early stages even may be in the development phase before application is tested (Unit test).

Testing progress is divided to 4 phases include: Unit test, Integration test, System test and Acceptance test.

* Unit testing:
* The purpose is to verify the internal logic code by testing every possible branch within the function, also known as test coverage.
* Unit test will be done by developers and tester.
* Finish unit test is help tester do integration testing and system testing.
* Integration testing:
* The separate modules will be tested together to expose faults in the interfaces and in the interaction between integrated components.
* Integration testing will be done by tester.
* System testing:
* Compare the system specifications against the actual system.
* System testing checks if the integrated product meets the specified requirements.
* Acceptance testing
* Acceptance testing will be performed by the test manager and development team leader.
* The acceptance test will be done for a period of 1 weeks after completion of the System/Integration test process.
* Programs will enter into Acceptance test after all critical and major defects have been corrected.
* Prior to final completion of acceptance testing all open critical and major defects must be corrected.

# Requirements for Test

## Test items

1. *User functions*

* Register as a new members.
* Register donate / receive blood.
* Search type of blood.
* Share a lot of story about yourself.
* Comment and share deal to another people by e-mail or social networks.

1. *Management functions*

* Manage deal (edit/show/hide).
* Manage source (enable/disable).
* Start collecting function manually.

1. *Performance*

* *Not define.*

## Acceptance Test Criteria

Criteria for Unit test of Development team, for Test team accepts to start testing:

* Number of UTC/KLOC: 100 UTC/KLOC
* Number defects/KLOC: 4-6 defects/KLOC
* Statement coverage: 100%
* Branch coverage: 100%
* Path coverage: 100%

Criteria which are based on to accept the products:

* Test coverage: 100%
* Successful Test coverage: 95%
* Number of Test cases (Unit/System Test cases): 60-70 TCs/KLOC
* Number of defects/Weighted defects: 3 defects/KLOC

## Feature not to be tested

* The stable of website when do not connect internet.
* Too much users connect on system.
* Feature auto collect.
* To reduce testing time, we only test on popular browsers and device.

# TEST STRATEGY

## Test types

### Function Testing

Function testing of the target-of-test should focus on any requirements for test that can be traced directly to use cases or business functions and business rules.

|  |  |
| --- | --- |
| Test Objective: | - Ensure proper target-of-test functionality, including navigation, data entry, processing, and retrieval. |
| Technique: | Execute each use case, use-case flow, or function, using valid and invalid data, to verify the following:  - The expected results occur when valid data is used.  - The appropriate error or warning messages are displayed when invalid data is used.  - Each business rule is properly applied.  - Use Test tool. |
| Completion Criteria: | - All planned tests have been executed.  - All identified defects have been addressed and closed. |
| Special Considerations: | - Identify or describe those items or issues (internal or external) that impact the implementation and execution of function test. |

### 

### User Interface Testing

User Interface (UI) testing verifies a user’s interaction with the software.

|  |  |
| --- | --- |
| Test Objective: | Verify the following:  - Navigation through the target-of-test properly reflects business       functions and requirements, including window-to-window, field-to-field, and use of access methods (tab keys, mouse movements, accelerator keys)  - Window objects and characteristics, such as menus, size, position, state, and focus conform to standards. |
| Technique: | - Create or modify tests for each window to verify proper navigation and object states for each application window and objects. |
| Completion Criteria: | - Each window successfully verified to remain consistent with benchmark version or within acceptable standard |
| Special Considerations: | - Not all properties for custom and third party objects can be accessed. |

### Data and Database Integrity Testing

The databases and the database processes should be tested as a subsystem within the Project.

|  |  |
| --- | --- |
| Test Objective: | - Ensure database access methods and processes function properly and without data corruption. |
| Technique: | - Invoke each database access method and process, seeding each with valid and invalid data or requests for data.  - Inspect the database to ensure the data has been populated as intended, all database events occurred properly, or review the returned data to ensure that the correct data was retrieved for the correct reasons. |
| Completion Criteria: | - All database access methods and processes function as designed and without any data corruption. |
| Special Considerations: | - Testing may require a DBMS development environment or drivers to enter or modify data directly in the databases.  - Processes should be invoked manually.  - Small or minimally sized databases (limited number of records) should be used to increase the visibility of any non-acceptable events. |

### Performance testing

#### Load Testing

Load testing is a performance test which subjects the target-of-test to varying workloads to measure and evaluate the performance behaviors and ability of the target-of-test to continue to function properly under these different workloads.

|  |  |
| --- | --- |
| Test Objective: | - Verify performance behavior time for designated transactions or business cases under varying workload conditions. |
| Technique: | - Use tests developed for Function or Business Cycle Testing.  - Modify data files to increase the number of transactions or the tests to increase the number of times each transaction occurs. |
| Completion Criteria: | - Multiple transactions or multiple users:  Successful completion of the tests without any failures and within acceptable time allocation. |
| Special Considerations: | - Load testing should be performed on a dedicated machine or at a dedicated time.  This permits full control and accurate measurement. The databases used for load testing should be either actual size or scaled equally. |

#### Volume Testing

Volume Testing subjects the target-of-test to large amounts of data to determine if limits are reached that cause the software to fail.

|  |  |
| --- | --- |
| Test Objective: | - Verify that the target-of-test successfully functions under the following high volume scenarios:  - Maximum (actual or physically- capable) number of clients connected, or simulated, all performing the same, worst case (performance) business function for an extended period.  - Maximum database size has been reached (actual or scaled) and multiple queries or report transactions are executed simultaneously. |
| Technique: | - Use tests developed for Performance Profiling or Load Testing.  - Multiple clients should be used, either running the same tests or complementary tests to produce the worst case transaction volume or mix (see Stress Testing above) for an extended period.  - Maximum database size is created (actual, scaled, or filled with representative data) and multiple clients used to run queries and report transactions simultaneously for extended periods. |
| Completion Criteria: | - All planned tests have been executed and specified system limits are reached or exceeded without the software or software failing. |
| Special Considerations: | - What period of time would be considered an acceptable time for high volume conditions, as noted above? |

### Regression Testing

Regression testing is a necessary maintenance activity aimed at showing that code has not been adversely affected by changes.

|  |  |
| --- | --- |
| Test Objective: | - Regression testing is to validate modified parts of the software, to make sure that the modification does not cause errors in other parts. |
| Technique: | - Reuse the set of test cases from an existing test suite to test a modified module.  - Use Rational Robot tool: Creating some functional test scripts. Define automated test execution schedule here**.**  - 80% Test cases is randomly select from existing Test cases.  - Construct a program-analysis infrastructure. We are building an extensible infrastructure to implement and evaluate a program-analysis.  Basing on the analysis result, we identify scope of regression test. |
| Completion Criteria: | - All test cases are performed and passed.  - All selected test cases are performed and passed. |
| Special Considerations: |  |

## Test stages

Clearly state the stage in which the test will be executed. Identified below are the stages in which common test are executed

| Type of Tests | Stage of Test | | | |
| --- | --- | --- | --- | --- |
| Unit | Integration | System | Acceptance |
| Function Test | X | X | X | X |
| User Interface test | X |  | X |  |
| Performance Tests  (Performance profiles of individual components) |  |  |  |  |
| Load, Stress, Volume test |  |  | X | X |
| Security test |  |  |  |  |
| Date integrity test |  |  |  |  |

# 

# RESOURCE

## Human Resource

|  |  |  |  |
| --- | --- | --- | --- |
| Worker/Doer | Role | Specific Responsibilities/Comments | Location |
| TamNT01730 | Test Leader | * Manage Test Resource and assign test tasks. * Create Test Plan. * Create and Self Review Test Case. * Execute test. * Collect data test. * Create Test Report. | FPT, Vietnam |

## Test management

* N/A

# Test environment

* Content maker tool will be tested in Window 7 professional, Window XP.
* VNBD website will be tested in browser: Firefox, Chrome, IE

## Hardware

* N/A

## Software

* N/A

## Infrastructure

List tools will be employed for this project.

|  |  |  |  |
| --- | --- | --- | --- |
| Purpose | Tool | Vendor/In-house | Version |
| Defect Log | MS-Excel | FPT-University | v1.0 |
| Test Effort | MS-Excel | FPT-University | v1.0 |

# TEST MILESTONES

Separating test milestones, it should be identified to communicate project status accomplishments.

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone Task | Effort (pd) | Start Date | End Date |
| Create Test Plan | 10 | 09/09/2013 | 20/09/2013 |
| Review & update TP | 3 | 23/09/2013 | 25/09/2013 |
| Create Unit Test case | 10 | 26/09/2013 | 09/10/2013 |
| Review & update UTC | 5 | 10/10/2013 | 16/10/2013 |
| Create Test View Point | 5 | 17/10/2013 | 23/10/2013 |
| Review & update Test View Point | 2 | 24/10/2013 | 25/10/2013 |
| Create Integration Test case | 10 | 28/10/2013 | 08/11/2013 |
| Review & Update Integration TC | 2 | 11/11/2013 | 12/11/2013 |
| Create System Test case | 10 | 13/11/2013 | 26/11/2013 |
| Review & Update System TC | 2 | 27/11/2013 | 28/11/2013 |
| Execute Unit Test | 7 | 29/11/2013 | 05/12/2013 |
| Execute Integration test | 7 | 06/12/2013 | 12/12/2013 |
| Execute System test | 7 | 13/12/2013 | 19/12/2013 |

# 

# DELIVERABLES

| No | Deliverables | Language | Delivered Date |
| --- | --- | --- | --- |
|  | Test Plan | English | 29/09/2013 |
|  | Unit Test cases | English | 15/11/2013 |
|  | Integration Test Cases | English | 20/11/2013 |
|  | System Test cases | English | 21/11/2013 |
|  | Defect log | English | 20/11/2013 |
|  | Test reports | English | 21/12/2013 |