|  |
| --- |
|  |

**VIETNAMESE STUDY SYSTEM FOR JAPANESE**

**TEST PLAN**

**Project Code: Veazy**

**Document Code: Veazy\_Test\_Plan\_v1.0\_EN**

**Ha Noi, 23/07/2016**

**Record of change**

\*A - Added M - Modified D - Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Effective Date | Changed Items | A\* M, D | Change Description | New Version |
| 23/07/2016 | Create new | A | Create new document | v1.0 |
|  |  |  |  |  |

**SIGNATURE PAGE**

**AUTHOR:** Nguyen Hong Quan 23/07/2016

Developer

**REVIEWERS:** Nguyen Trong Duy

Developer

**APPROVAL:** Nguyen Van Sang

Supervisor

Contents

[1 INTRODUCTION 4](#_Toc457569394)

[1.1 Purpose 4](#_Toc457569395)

[1.2 Definitions, Acronyms, and Abbreviations 4](#_Toc457569396)

[1.3 References 5](#_Toc457569397)

[1.4 Background information 5](#_Toc457569398)

[1.5 Scope of testing 5](#_Toc457569399)

[1.6 Constraints 7](#_Toc457569400)

[1.7 Risk list 8](#_Toc457569401)

[1.8 Training needs 9](#_Toc457569402)

[2 Requirements for Test 11](#_Toc457569403)

[2.1 Test items 11](#_Toc457569404)

[2.2 Acceptance Test Criteria 13](#_Toc457569405)

[2.3 Feature to be tested 14](#_Toc457569406)

[2.4 Feature not to be tested 14](#_Toc457569407)

[3 TEST STRATEGY 14](#_Toc457569408)

[3.1 Test types 14](#_Toc457569409)

[3.1.1 Function Testing 14](#_Toc457569410)

[3.1.2 User Interface Testing 15](#_Toc457569411)

[3.1.3 Data and Database Integrity Testing 16](#_Toc457569412)

[3.1.4 Performance Testing 17](#_Toc457569413)

[3.2 Test stages 18](#_Toc457569414)

[4 RESOURCE 19](#_Toc457569415)

[4.1 Human Resource 19](#_Toc457569416)

[5 Test environment 20](#_Toc457569417)

[5.1 Hardware 20](#_Toc457569418)

[5.2 Software 21](#_Toc457569419)

[5.3 Infrastructure 21](#_Toc457569420)

[6 TEST MILESTONES 22](#_Toc457569421)

[7 DELIVERABLES 23](#_Toc457569422)

# INTRODUCTION

## Purpose

This is the comprehensive test plan of the Veazy project. The purpose of the document describes scopes of test and activities which need to be taken during test process of project.

## Definitions, Acronyms, and Abbreviations

|  |  |  |
| --- | --- | --- |
| Abbreviations | Description | Note |
| Veazy | Vietnamese Study System For Japanese |  |
| TL | Test Leader |  |
| TC | Test Plan |  |
| TC | Test Case |  |
| CT | Component Test |  |
| ST | System Test |  |
| IT | Integration Test |  |
| UT | Unit Test |  |
| GUI | Graphical User Interface |  |
| KLOC | 1000 line of code |  |
| SRS | Software Requirement Specification |  |
| TR | Test Report |  |

**Table 1-1**: Definitions and acronyms

## References

|  |  |  |  |
| --- | --- | --- | --- |
| Title/File name | Author | Version | Effective Date |
| Veazy\_SRS\_en\_v1.3.docx | Veazy | 1.3 | 20/06/2016 |
| Veazy\_ProjectPlan | Veazy |  |  |

**Table 1-2:** Reference files

## Background information

* The target of testing is ensured all functions will be run correctly as SRS description. In addition, restrict maximum of defect during the user access in the application. To do this target, website will have to:
* Passed the stages of testing: Unit Testing, Component Testing, Integration Testing, System Testing, Acceptance Testing
* Passed the types of testing: Function Testing, User Interface Testing , Data and Data Integrity Testing
* Run normally in required devices.

## Scope of testing

Five phases of testing

**Phase 1: Unit testing**

* Unit testing will be done by developers using White Box Testing technique
* When executing unit test, if any bugs are found, developers have to log bug into “Defect Log Management” file and fix it 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 2: Component testing**

* Component testing will be performed by testers using Black Box Testing technique.
* Material are component test cases based on low- level design.
* When executing component test, if any bugs are found, testers have to log bug into “Defect Log Management” file and assign developer to fix it and repeat 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: Integration testing**

* After finishing component testing, Integration testing will be performed by testers.
* Material are integration test cases, high- level design and test tools.
* Doing test by checking flows between functions and items which have relation.
* When executing integration test, if any bugs are found, testers have to log on “Defect Log Management” file and assign developer to fix it and repeat 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 4: System testing**

* After finishing integration testing, developers collect all functions and items then testers will perform system test, which means test the whole system.
* Materials are system test case, SRS
* If any bugs are found, developers have to fix then testers will verify them. System test is complete only when all test cases are passed and no bug is found.

*Rule for filling test result:*

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

**Phase 5: Acceptance testing**

* Base on user 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.
* Acceptance testing is complete only when whole system met requirement specification.

## Constraints

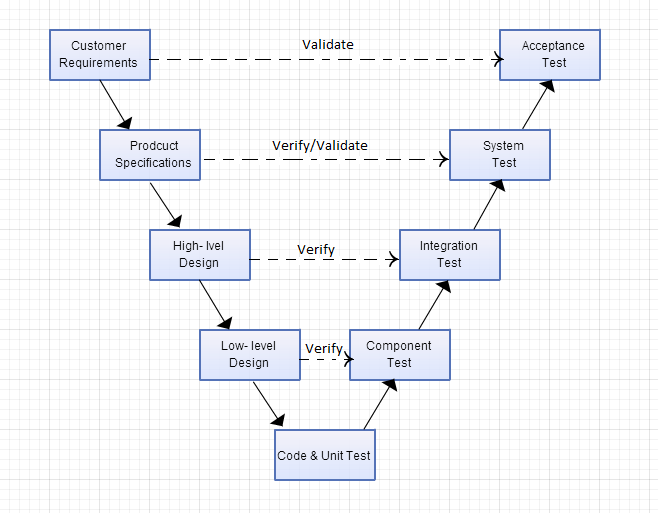
* Veazy website was built with bootstrap, java, angularjs and Sql database, thus need to master the language, as well as how to operate SQL.
* Must prepare fully testing tools and PC.
* Ensure website runs on IE, Chrome, Firefox.
* All team members must ensure the completion of the schedule in the Test Plan.
* All issues arising in the process of testing must contact TL to solution and report regularly on these issues.
* Working process must be reported daily to TL.
* Members participate fully in testing instruction sessions and make testing comments for each module of the website.

## Risk list

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Risk** | **Solution** | **Impact** |
| 1 | Qualifications of project team members are not enough to complete all the work of the project. | - Training existing resources to respond necessary skills.  - The members of the group support each other. | High |
| 2 | Lack of resources, hardware and software to ensure the process of testing. | - Check existing computer systems, taking suggestions from members about hardware and software of computer capable of catering the process of testing. | High |
| 3 | Later than the scheduled time. | - Using MS Project to manage and assign tasks to each member. | High |
| 4 | Not enough time to write all test cases, execute test or re-test for fixed bug. | - Regular monitoring, reminding and supporting members.  - Increase working time. | High |
| 5 | Tester can be ill during the testing phase. | - All members of the group must be instructed to backup testing for ill members. | High |
| 6 | The computer may be damaged or stolen. | - Prepare 1 or 2 backup device.  - Backup your data to multiple copies. | High |

## Training needs

Veazy project follows V-Model process:



**Figure 1: V-Model**

Testing progress is divided to 5 phases include: Unit test, Component test, Integration test, System test and Acceptance test

* Unit test:
* Unit testing is used to verify a single minimal unit of source code. The purpose of unit testing is to isolate the smallest testable parts of Veazy and verify that they function properly in isolation.
* Unit testing is the first level of testing and is performed prior to component testing
* Unit testing will be done by developer.
* Component test:
* Component testing is used to validate a combined many minimal units of source code.
* Component testing is performed after unit testing and before integration testing
* Component testing will be done by tester
* Integration test:
* Integration testing is a level of the software testing process where individual units or component are combined and tested as a group.
* The purpose is to expose faults in the interaction between integrated units.
* Integration testing is performed after component testing
* Integration testing will be done by tester
* There are two methods of doing integration testing: Bottom-up Integration testing and Top Down Integration testing:

|  |  |
| --- | --- |
| No | Integration Testing Method |
| 1 | Bottom-up integration  This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules. |
| 2 | Top- Down integration  This testing, the highest-level modules are tested first and progressively lower-level modules are tested after that |

**Table 1-3:** Integration test

* System test:
* System Testing is a level of the software testing process where a complete, integrated system is tested
* The purpose is to evaluate the system’s compliance with the specified requirements
* System testing is performed after integration testing
* System testing will be done by tester
* Acceptance test:
* Acceptance testing is performed after system testing
* Acceptance testing will be performed by the test leader and people outside project team.
* The acceptance test will be done for a period of 1 weeks after completion of the system test process.

# Requirements for Test

## Test items

|  |  |  |
| --- | --- | --- |
| No |  | Functions |
| 1 | Guest | * Register * Login * Take Test * View Result Test * Search Dictionary * Search Lesson * View Lesson List * View Lesson Detail |
| 2 | Member | * Log in / Log out * Take Test * View Result Test * View List Test History * Retake Test * Report Question * Search Dictionary * View Lesson List * View Lesson Detail * Report Question * View Profile * Edit Profile * Change Password * View Test Statistic |
| 3 | Content Editor | * Log in / Log out * View Profile * Edit Profile * Change Password * View Lesson List * Edit Lesson * Delete Lesson * View Lesson Detail * Search Lesson * Create Lesson * View Report * View Question List * Edit Question * Delete Question * Search Question * View Question Details * Create Question |
| 4 | Administrator | * Log in / Log out * View Profile * Change Password * View Uses List * View User Profile * Ban User * Unban User * Set User Role * Search User * View System Statistics |

**Table 2-1:** Test item

## Acceptance Test Criteria

* Criteria for Unit test of Development team, for Test team accepts to start testing:
* Number of UTC/KLOC: 40 UTC/KLfOC
* Number defects/KLOC: 3-4 defects/KLOC
* Statement coverage: 97%
* Branch coverage: 100%
* Path coverage: 100%
* Criteria for Integration test:
* Number of UTC/KLOC: 30 UTC/KLOC
* Number defects/KLOC: 2-3 defects/KLOC
* Criteria for System test:
* Number of UTC/KLOC: 60 UTC/KLOC
* Number defects/KLOC: 4-6 defects/KLOC
* Criteria for Acceptance test:
* Number defects/KLOC: 1-2 defects/KLOC

## Feature to be tested

Refer [Test items](#h.z337ya) for more details

## Feature not to be tested

* The stability and availability of system without connecting to the network (3G, Wifi)
* Many users connect to system at the same time.

# TEST STRATEGY

## Test types

### Function Testing

* Functionality testing of all implemented functions on the Veazy system.
* Functionality testing is performed to verify that a software application performs and functions correctly according to the functional requirements/specifications.
* Functions are tested by feeding them input and examining the output. This type of testing is not concerned with how processing occurs, but rather, with the results of processing.
* During functional testing, [Black Box Testing](http://softwaretestingfundamentals.com/black-box-testing/) technique is used in which the internal logic of the system being tested is not known to the tester.
* Functional testing typically involves these steps:
  + The identification of functions that the software is expected to perform
  + The creation of input data based on the function's specifications
  + The determination of output based on the function's specifications
  + The execution of the test case
  + The comparison of actual and expected outputs

|  |  |
| --- | --- |
| **Test Objective** | Verify the application and its internal processes by interacting with the application via the Graphical User Interface (GUI) and analyzing the outputs or results |
| **Technique** | Executing 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. |
| **Completion Criteria** | - All planned tests have been executed.  - All identified defects have been addressed and closed. |
| **Special Considerations** | Testing may be stopped when   * Time runs out * A certain number of defects found * Test coverage > 97% * Stop when testing becomes unproductive |

**Table 3-1:** Function Testing

### User Interface Testing

* GUI testing is the process of testing a product's graphical user interface to ensure it meets its specifications
* GUI test will be performed fully on all screens.
* GUI testing evaluates design elements such as layout, colors, [fonts](http://whatis.techtarget.com/definition/font), font sizes, labels, text boxes, text formatting, captions, buttons, lists, icons, links, content and more.
* For instance, it must be able to provide inputs to the input fields.
  + Check Error Messages are displayed correctly.
  + Check Font used in application is readable.
  + Check the Color of the font and warning messages is aesthetically pleasing.
  + Check the alignment of the text is proper.
  + Check that the images have good clarity.
  + Check the positioning of GUI elements for different screen resolution.

|  |  |
| --- | --- |
| **Test Objective** | Navigation through the target-of-test properly reflects business functions and requirements, including screen to screen, field-to-field, and use of access methods. 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. |

**Table 3-2:** GUI Testing

### Data and Database Integrity Testing

* The databases and the database processes should be tested as a subsystem within the Project. These subsystems should be tested without the target-of-test’s User Interface as the interface to the data.  Additional research into the Database Management System (DBMS) needs to be performed to identify the tools and techniques that may exist to support the testing identified below.

|  |  |
| --- | --- |
| **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. |

**Table 3-3:** Data and Data Integrity Testing

### Performance Testing

|  |  |
| --- | --- |
| **Test Objective:** | Verify performance behaviors for designated transactions or business functions under the following conditions:  + Normal anticipated workload( Normal time to load a page)  + Anticipated worst case workload (Longest time to load a page) |
| **Technique:** | Use Test Procedures developed for Function or Business Cycle Testing.  Modify data files to increase the number of transactions or the scripts to increase the number of iterations each transaction occurs.  Scripts should be run on one machine (best case to benchmark single user, single transaction) and be repeated with multiple clients (virtual or actual). |
| **Completion Criteria:** | Single Transaction or single user: Successful completion of the test scripts without any failures and within the expected or required time allocation per transaction.  Multiple transactions or multiple users: Successful completion of the test scripts without any failures and within acceptable time allocation. |
| **Special Considerations:** | Comprehensive performance testing includes having a background workload on the server.  There are several methods that can be used to perform this, including:  Use multiple physical clients, each running test scripts to place a load on the system.  Performance testing should be performed on a dedicated machine or at a dedicated time. This permits full control and accurate measurement.  The databases used for Performance Testing should be either actual size or scaled equally. |

**Table 3-4:** Data and Data Integrity Testing

## 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** | **Component** | **Integration** | **System** | **User Acceptance** |
| **Function Testing** | **X** | **X** | **X** | **X** | **X** |
| **User Interface Testing** |  |  | **X** | **X** |  |
| **Data & Data Integrity Testing** |  |  | **X** | **X** |  |
| **Performance Testing** | **X** |  | **X** |  |  |

# RESOURCE

## Human Resource

|  |  |  |
| --- | --- | --- |
| **Worker/Doer** | **Role** | **Specific Responsibilities/Comments** |
| **Nguyen Trong Duy** | **Test Leader** | Provides management oversight.  Responsibilities include:  • Manage Test Resource and assign test tasks.  • Create and review Test Plan.  • Review Test Case.  • Review Test Report |
| **Nguyen Hong Quan** | **Tester** | • Create and review Test Case.  • Execute test.  • Create Test viewpoints  • Create and review Test Report |
| **Pham Duc Thang** | **Tester** | • Create and review Test Case.  • Execute test.  • Create Test viewpoints  • Create and review Test Report |

**Table 4-1:** Human resource

# Test environment

## Hardware

|  |  |  |
| --- | --- | --- |
| **System resources** | | |
| **Resource** | **Quantity** | **Name and Type** |
| Tester PCs | 3 |  |
| CPU | 1 | Intel Dual Core i3 |
| RAM | 2 | 2G |
| HardWare | 1 | 250GB |

**Table 5-1:** Hardware

## Software

|  |  |  |
| --- | --- | --- |
| **Name** | **Version** | **Type and Others Notes** |
| Windows | 7, 8, 8.1, 10 | Operating system |
| IE | 8, 9, 10, 11 | Web browser |
| Firefox | 47.0.1 | Web browser |
| Chrome | 52.0.2743.82 | Web browser |
| Office | 2010, 2013 | Microsoft office |

**Table 5-2:** Software

## Infrastructure

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Purpose** | **Detail** | **Vendor/In-house** | **Version** |
| Veazy\_Defect Log Management | Tracking bug during testing time | Microsoft Excel 2010, 2013 | FPT University | v1.0 |
| Test Effort | Effort execute test | Microsoft Excel 2010, 2013 | FPT University | v1.0 |

**Table 5-3:** Infrastructure

# TEST MILESTONES

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone Task** | **Effort (pd)** | **Start Date** | **End Date** |
| Create TP | 3 | 21/07/2016 | 23/07/2016 |
| Review & update TP | 2 | 24/07/2016 | 25/07/2016 |
| Create Component Test case | 3 | 26/07/2016 | 28/07/2016 |
| Review & update Component TC | 2 | 29/07/2016 | 30/07/2016 |
| Create Integration TC | 3 | 31/07/2016 | 02/08/2016 |
| Review & Update Integration TC | 2 | 03/08/2016 | 04/08/2016 |
| Create System TC | 3 | 05/08/2016 | 07/08/2016 |
| Review & Update System TC | 2 | 08/08/2016 | 09/08/2016 |
| Execute CT phase 1 | 2 | 10/08/2016 | 11/08/2016 |
| Execute CT phase 2 | 2 | 12/08/2015 | 13/08/2016 |
| Execute IT phase 1 | 1 | 14/08/2016 | 14/08/2016 |
| Execute IT phase 2 | 1 | 15/08/2016 | 15/08/2016 |
| Execute ST phase 1 | 1 | 16/08/2016 | 16/08/2016 |
| Execute ST phase 2 | 1 | 17/08/2016 | 17/08/2016 |

**Table 6-1:** Test milestones

# DELIVERABLES

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Deliverables** | **Language** | **Delivered Date** |
|  | Test Plan | English | 25/07/2016 |
|  | Component Test case | English | 30/07/2016 |
|  | Integration Test case | English | 04/08/2016 |
|  | System Test case | English | 09/08/2016 |
|  | Defect Log Management | English | 10/08/2016 |
|  | Test report | English | 10/08/2016 |

**Table 7-1**: Deliverables