|  |
| --- |
|  |

USEFUL JAPANSES DICTIONARY FOR VIETNAMESE

TEST PLAN

Project Code: UJD\_VN

Document Code: UJD\_VN\_Test Plan\_v1.0\_EN

**Ha Noi, 16/06/2014**

Record of change

\*A - Added M - Modified D - Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Effective Date | Changed Items | A\* M, D | Change Description | New Version |
| 16/06/2014 | Add new | A |  | v0.1 |
| 23/06/2014 | Update | M | Update comment of document review | v0.9 |
| 30/06/2014 | Update | M | Update comment of document review | v1.0 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

SIGNATURE PAGE

**ORIGINATOR:** Pham Thi Minh 16/06/2014

Test Leader (TL)

**REVIEWERS:** Le Dinh Nam 23/06/2014

Project Manager (PM)

**APPROVAL:** Nguyen Van Sang

Supervisor

TABLE OF CONTENTS

[1 INTRODUCTION 5](#_Toc392930573)

[1.1 Purpose 5](#_Toc392930574)

[1.2 Definitions, Acronyms, and Abbreviations 5](#_Toc392930575)

[1.3 References 6](#_Toc392930576)

[1.4 Background information 6](#_Toc392930577)

[1.5 Scope of testing 6](#_Toc392930578)

[1.6 Constraints 8](#_Toc392930579)

[1.7 Risk list 8](#_Toc392930580)

[1.8 Training needs 8](#_Toc392930581)

[2 Requirements for Test 10](#_Toc392930582)

[2.1 Test items 10](#_Toc392930583)

[2.2 Acceptance Test Criteria 12](#_Toc392930584)

[2.3 Feature not to be tested 12](#_Toc392930585)

[3 TEST STRATEGY 14](#_Toc392930586)

[3.1 Test types 14](#_Toc392930587)

[3.1.1 Function Testing 14](#_Toc392930588)

[3.1.2 User Interface Testing 15](#_Toc392930589)

[3.1.3 Data and Database Integrity Testing 15](#_Toc392930590)

[3.2 Test stages 16](#_Toc392930591)

[4 RESOURCE 17](#_Toc392930592)

[4.1 Human Resource 17](#_Toc392930593)

[5 Test environment 18](#_Toc392930594)

[5.1 Hardware 18](#_Toc392930595)

[5.2 Software 18](#_Toc392930596)

[5.3 Infrastructure 18](#_Toc392930597)

[6 TEST MILESTONES 20](#_Toc392930598)

[7 DELIVERABLES 21](#_Toc392930599)

# 

# INTRODUCTION

## Purpose

The purpose of this document is show scope of testing, test items, test strategy, testing approach, resources, test environment, schedule of intended testing activities …

## Definitions, Acronyms, and Abbreviations

| **Abbreviations** | **Description** | **Note** |
| --- | --- | --- |
| UJD\_VN | Useful Japanese Dictionary for Vietnamese |  |
| TL | Test Leader |  |
| PM | Project Manager |  |
| QA | Quality Assurance |  |
| SRS | Software Requirement Specification |  |
| TC | Test Case |  |
| TP | Test Plan |  |
| ST | System Test |  |
| IT | Integration Test |  |
| UT | Unit Test |  |
| GUI | Graphic User Interface |  |
| TR | Test Report |  |
| KLOC | 1000 line of code |  |

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

## References

| **Title/File name** | **Author** | **Version** | **Effective Date** |
| --- | --- | --- | --- |
| SRS | UJD\_VN Team | v1.0 | 20/06/2014 |
| UJD\_VN\_Project Plan\_v1.0\_EN | NamLD | v1.0 | 28/05/2014 |

**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/browsers.

## Scope of testing

* UJD\_VN will be tested by 5 phases:

**Phase 1: Unit testing**

* Unit testing will be done by developers
* Developers user While Box Testing technique to do
* When executing unit testing, if any bugs are found, developers have to log bug on “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**

* After finishing unit testing, component testing will be performed by testers.
* Material are unit test cases, low- level design
* Testers user Black Box Testing technique to do
* When executing component testing, if any bugs are found, testers have to log on “Defect Log Management” 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: Integration testing**

* After finishing component testing, integration testing will be performed by testers.
* Material are high- level design and test tools.
* Do test by flow of functions and items which have relation.
* When executing integration testing, if any bugs are found, testers have to log on “Defect Log Management” 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 4: System testing**

* After finishing integration testing and developers collect all functions and items, testers will be performed system testing, it means doing test whole system.
* If any bugs are found, developers have to fix and testers will verify them. System test is ended only when 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 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 ended only when whole system met requirement specification.

## Constraints

* There are much environments that UJD\_VN should be tested. But the number of tester can’t cover all environments.

## Risk list

* Performance test: Cannot test the case which many users connect to website at the same time.
* Not enough time to write all test case or re-fix bug
* Tester be ill

## Training needs

UJD\_VN 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 UJD\_VN and verify that they function properly in isolation.
* Unit testing is the first level of testing and is perform prior to component testing
* Unit testing will be done by developer.
* Component test:
* Component testing is used to validate a single minimal unit 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 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 team leader.
* The acceptance test will be done for a period of 1 weeks after completion of the system test process.

# Requirements for Test

## Test items

1. *Member functions*

* Search sentences
* Search conversation
* Search grammar
* Search video
* Search specialized Japanese
* Log in by Facebook account
* Log in by Google account
* Log in by registered account
* Log out
* Register
* Edit profile
* Forget password
* Contribute content
* Contribute opinion
* Send Q&A
* Doing test
* Tracking mark
* Training listening
* Listening conversation
* Reading document

1. *Admin functions*

* Add new admin
* Edit profile
* Delete admin
* Search member
* Ban/Unban member’s account
* Delete member
* Search vocabulary
* Add vocabulary
* Edit vocabulary
* Delete vocabulary
* Search grammar
* Add grammar
* Edit grammar
* Delete grammar
* Search reading document
* Add reading document
* Edit reading document
* Delete reading document
* Search conversation
* Add conversation
* Edit conversation
* Delete conversation
* Search video
* Add video
* Edit video
* Delete video
* Search listening article
* Add listening article
* Edit listening article
* Delete listening article
* Search test
* Add test
* Edit test
* Delete test
* Search contribute content
* Approve content
* Reply content’s user
* Delete content
* Search contribute opinion
* Reply opinion’s user
* Delete opinion

## Acceptance Test Criteria

* Criteria for Unit test of Development team, for Test team accepts to start testing:
* Number of UTC/KLOC: 60 UTC/KLOC
* Number defects/KLOC: 4-6 defects/KLOC
* Statement coverage: 100%
* Branch coverage: 100%
* Path coverage: 100%

## Feature not to be tested

* The stable of website when do not connect internet.
* Many users connect to system at the same time.

# TEST STRATEGY

## Test types

### Function Testing

* Functional testing is a type of software testing whereby the system is tested against the functional requirements/specifications.
* Functions are tested by feeding them input and examining the output. Functional testing ensure that the requirements are properly satisfied by the website. 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.

|  |  |
| --- | --- |
| Test Objective: | The type of this test is to ensure proper target-of-test functionality, including user interaction, all function defined in specification document implemented correctly. |
| 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 ensuring proper functionality of the GUI for a given web and making sure it conforms to its written specifications.
* 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.

|  |  |
| --- | --- |
| 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. |

**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

## 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 |
| Function Testing | X | X | X | X |
| User Interface Testing |  | X | X |  |

# 

**Table 3-4:** Test stage

# RESOURCE

## Human Resource

|  |  |  |
| --- | --- | --- |
| Worker/Doer | Role | Specific Responsibilities/Comments |
| MinhPT | Test Leader | * Manage Test Resource and assign test tasks. * Create and review Test Plan. * Create and review Test Case. * Execute test. * Create Test view points * Review Test Report |
| TuanNN | Tester | * Create and review Test Case. * Execute test. * Collect data test. * Create Test view points * Create Test Report. |

**Table 4-1:** Human resource

# Test environment

## Hardware

|  |  |  |
| --- | --- | --- |
| Name | Purpose | Detail |
| Laptop Asus | Device for create and execute test | Window 7 Ultimate Core i3 |
| Laptop Vaio | Device for create and execute test | Window 7 Ultimate Core i3 |

**Table 5-1:** Hardware

## Software

|  |  |  |
| --- | --- | --- |
| Name | Purpose | Detail |
| Test Plan | Managing test | Microsoft Word 2013, 2010 |
| Test case | Executing test | Microsoft Excel 2013, 2010 |
| Test report, Test checklist | Tracking test | Microsoft Excel 2013, 2010 |
| Chrome, CocCoc | Executing test | Chrome 35.0, CocCoc 35.0 |

**Table 5-2:** Software

## Infrastructure

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

**Table 5-3:** Infrastructure

# TEST MILESTONES

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone Task | Effort (pd) | Start Date | End Date |
| Create Test Plan | 6 | 16/06/2014 | 22/06/2014 |
| Review & update TP | 1 | 23/06/2014 | 23/06/2014 |
| Create Unit Test case | 12 | 23/06/2014 | 28/06/2014 |
| Review & update UTC | 1 | 29/0762014 | 29/06/2014 |
| Create Integration Test case | 4 | 03/07/2014 | 04/07/2014 |
| Review & Update Integration TC | 1 | 04/07/2014 | 04/07/2014 |
| Create System Test case | 2 | 07/07/2014 | 07/07/2014 |
| Review & Update System TC | 1 | 07/07/2014 | 07/07/2014 |
| Create Test Checklist | 2 | 08/07/2014 | 08/07/2014 |
| Execute Integration test phase 1 | 4 | 15/07/2014 | 16/07/2014 |
| Execute Integration test phase 2 | 4 | 22/07/2014 | 23/07/2014 |
| Execute System test phase 1 | 2 | 17/07/2014 | 17/07/2014 |
| Execute System test phase 2 | 12 | 24/07/2014 | 29/07/2014 |

**Table 6-1:** Test milestones

# DELIVERABLES

| No | Deliverables | Language | Delivered Date |
| --- | --- | --- | --- |
|  | Test Plan | English | 23/06/2014 |
|  | Unit Test cases | English | 30/06/2014 |
|  | Integration Test Cases | English | 05/07/2014 |
|  | System Test cases | English | 08/07/2014 |
|  | Defect Log Management | English | 07/07/2014 |
|  | Test reports | English | 30/07/2014 |

**Table 7-1**: Deliverables