# SOFWARE TESTING ソフトウェアテスト

## Introduction　はじめに

### Purpose　目的

This is the comprehensive test plan of the Jobs Finding System project. The purpose of this chapter describes scopes of test plan and activities which need to be taken during test process of project. It addresses the following items:

* Scope of Testing
* Requirement for Testing
* Test Strategy
* Test Resources
* Test Milestones
* Test Deliverables

### References　参照資料

| Title/File name | Author | Version |
| --- | --- | --- |
| JFS\_Software-Requirements-Specification\_v1.0 | DuyDD | v1.0 |
| JFS\_DataDesign\_v1.0 | NamTH | v1.0 |
| JFS\_Architecture-Desgin\_v1.0 | DuyDD, NamLH | v1.0 |
| JFS\_Screen-Desgin\_v1.0 | DuyDD | v1.0 |
| JFS\_TestCase\_v1.0 | SonPT | v1.0 |
| JFS\_UnitTest\_v1.0 | SonPT | v1.0 |
| JFS\_Project-Plan\_v1.0 | ThuyTT | v1.0 |

### Background information　背景情報

The target of testing is ensured all functions will be run correctly as SRS description. In addition, restrict maximum of defect during the using of Website. To do this target, Website have to:

* Pass all of the following stages of testing: Unit Testing, Integration Testing, System Testing and Acceptance Testing.
* Pass all of the following types of testing: Function Testing, User Interface Testing, Data and Data Integrity Testing
* Run normally in popular browsers and many types of device or display resolution.

### Scope of testing　テスト範囲

The scope of test will be limited to testing on 3 browser: Google Chrome, Mozilla Firefox, Microsoft Edge.

Functional items and non-functional items will be verified and passed by eJob development team, then be validated and approved by project manager and supervisor of JFS project.

## Constraints　制約事項

* Deadline for testing only can be met if development progress is on time.
* Working process must be reported daily to PTL.
* Have more environments should be tested: Window XP, Window 8 … and more browsers: Chrome 53, Firefox, Microsoft Edge 38 …

### Risk listリスク一覧

|  |  |  |  |
| --- | --- | --- | --- |
| No | Risk | Mitigation | Contingencies |
|  | PC have problem during test | Fix a problem | Find a other PC |
|  | Tester may be sick in test phase or may leave the project before completion | + Discussing  + Committing  + Assigning tasks appropriately | + Persuading  + Reviewing and re-planning the whole project |
|  | Scope: SRS/SAD may be changed | + Discussing  + Reviewing the whole project frequently | + Re-planning test phase  + Re-creating test case |
|  | Time of Coding phase may last too long | + Committing  + Assigning tasks | Working over time |

Table 5.2.1 Risk list

## Requirements for Test　テスト要求

### Test items　テスト頭目

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name of features and functions | Outline of features and functions | Note |
|  | Guest\_Common | Register |  |
|  | View Employer’s Posts |  |
|  | View Job Seeker’s Posts |  |
|  | Search Employer’s Posts |  |
|  | Search Job Seeker’s Posts |  |
|  | View detail Post |  |
|  | Member\_Common | Login |  |
|  | Logout |  |
|  | Forgot password |  |
|  | Change password |  |
|  | Search Employer’s Posts |  |
|  | Search Job Seeker’s Posts |  |
|  | View Employer’s Posts |  |
|  | View Job Seeker’s Posts |  |
|  | Apply suitable Job |  |
|  | Get suitable Job Seeker |  |
|  | Report other Member’s Post |  |
|  | Payment to get hidden Profile of Job Seekers |  |
|  | Online chat with Admin |  |
|  | Comment and report issue |  |
|  | Member\_Manage Personal Page | View Profile |  |
|  | Edit Profile |  |
|  | View personal Page |  |
|  | Member\_View other Member’s Personal Page | View other Member’s Posts |  |
|  | View other Member’s basic Profile |  |
|  | Member\_Manage Post | Create Post |  |
|  | Edit Post |  |
|  | Delete Post |  |
|  | View selected Posts |  |
|  | View Posts |  |
|  | View detail Post |  |
|  | View people choose |  |
|  | View matching Posts |  |
|  | Admin\_Common | Admin Login |  |
|  | Admin Logout |  |
|  | Admin Forgot password |  |
|  | Online chat with Member |  |
|  | Send notify by email |  |
|  | Add a Admin |  |
|  | Admin\_Manage Members | View Members |  |
|  | Search Members |  |
|  | View Member Profile |  |
|  | Ban Member |  |
|  | Unban Member |  |
|  | Delete Member |  |
|  | Add money into Employer’s Account |  |
|  | Admin\_Manage Posts | View Employer’s Posts |  |
|  | View Job Seeker’s Posts |  |
|  | Search Employer’s Posts |  |
|  | Search Job Seeker’s Posts |  |
|  | Delete Post |  |
|  | View detail Post |  |
|  | Admin\_Manage Feedback | View comments and reported issues of website |  |
|  | Delete comment and reported issues of website |  |
|  | Admin\_Statistic | Statistic earned money from Members |  |
|  | View Chart |  |
|  | Admin\_Reports | View Reports about Post |  |
|  | Delete Reports about Post |  |
|  | Admin\_Manage Category | Add Category |  |
|  | View Category |  |
|  | Delete Category |  |
|  | Edit Category |  |
|  | System | Send hidden profile of a Job Seeker to Employer |  |
|  | Send notification to Member’s email |  |
|  | Delete Member |  |
|  | Ban Member |  |
|  | Deduct Employer’s money |  |
|  | Create chart to compare between job types |  |
|  | List matching posts |  |

Table 5.3.1 Test items

### Acceptance Test Criteria　テスト受入基準

List of criteria to define what levels of test quality are sufficient to move to the next testing phase:

* Test coverage : 100%
* Successful Test coverage : 90%
* Number of defects/Weighted defects : 3~4 defects/KLOC

List of criteria which are based on to accept the products, for Test team to accept source code after Unit test of Development team:

* Number of UTC/KLOC: 40 UTC/KLOC
* Number or Weighted defects/KLOC : 3~4 defects/KLOC
* Statement coverage: 100%
* Branch coverage: 100%
* Path coverage: 100%

List of criteria which are based on to accept the products, for Integration test:

* Number of UTC/KLOC: 30 UTC/KLOC
* Number or Weighted defects/KLOC : 2~3 defects/KLOC
* Statement coverage: 100%
* Branch coverage: 100%
* Path coverage: 100%

List of criteria which are based on to accept the products, for System test:

* Number of UTC/KLOC: 20 UTC/KLOC
* Number or Weighted defects/KLOC : 5~6 defects/KLOC
* Statement coverage: 100%
* Branch coverage: 100%
* Path coverage: 100%

List of criteria which are based on to accept the products, for Acceptance test:

* Number or Weighted defects/KLOC : 1~2 defects/KLOC
* Statement coverage: 100%
* Branch coverage: 100%
* Path coverage: 100%

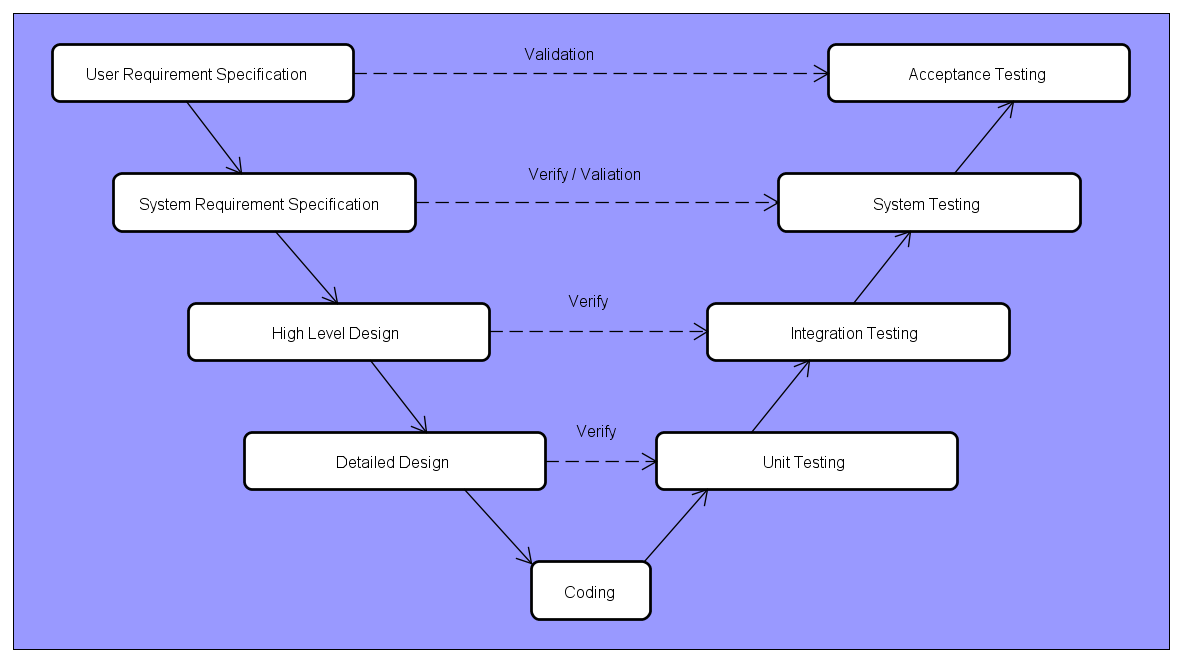
### Feature not to be tested　テストされない特徴

* The stable of website when do no connect internet.
* Manage Website’s payment activities**.** Manage the activity that user pay for system when they use paid-required service.

## Test strategy　テスト戦略

### Test Model　テストモデル

JFS follows V-Model process: The entire process of our Project is divided into 2 corresponding stages: Developing and Testing. Each developing process will be performed in parallel with the corresponding testing process. Bugs will be detected early.



JFS will be tested by 4 phases:

**Phase 1: Unit testing**

* Unit testing will be performed by developers
* Developers user White 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: Integration testing**

* After finishing component testing, integration testing will be performed by testers.
* Material are integration test cases, high- level design and test tools.
* Integration test focuses on specific areas of use cases when all requirements are completed.
* Integration test should be performed to ensure all components incorporate well.
* 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 3: 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.
* Material area system test case, SRS
* 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 4: Acceptance testing**

* Acceptance testing is performed after the system testing
* Base on business/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 ended only when whole system met requirement specification.

### Test types テストタイプ

### Function Testing　機能テスト

* Function 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 ensures that the requirements are properly satisfied by application. This type of testing is not concerned with how processing occurs, but rather, with results of processing.
* During functional testing, **Black Box Testing** technique is used in which the internal logic of the system being tested is not known to the tester.

|  |  |
| --- | --- |
| 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 |

Table 5.4.3 Function Testing

### User Interface Testing　ユーザーインターフェーステスト

* User interface testing, a testing technique used to identify the presence of defects is a product/software under test by using GUI.
* 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 and content.

|  |  |
| --- | --- |
| 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 5.4.4 User Interface Testing

### Data and Database Integrity Testing　でーたおよびデータベース整合性テスト

Data integrity corresponds to the quality of data in the databases and to the level by which users examine data quality, integrity and reliability. Data integrity testing verifies that the data in the database is accurate and functions as expected within a given application.

|  |  |
| --- | --- |
| 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 5.4.5 Data and Database Integrity Testing

### Test stages　テストステージ

| Type of Tests | Stage of Test | | | |
| --- | --- | --- | --- | --- |
| Unit | Integration | System | Acceptance |
| Function Test | X | X | X | X |
| User Interface test | X |  | X |  |
| Date integrity test |  | X | X |  |

Table 5.4.6 Test Stages

## Resource　リソース

### Human Resource　人的リソース

|  |  |  |  |
| --- | --- | --- | --- |
| Worker/Doer | Role | Specific Responsibilities/Comments | Location |
| SonPT | Test Leader | Manage Test resource and assign test tasks  Create TP  Review TC  Create and review Test Report | FPT University Vietnam |
| ThuyTT | Tester | Create and review TC for modules  Create and review UT  Execute test  Create Test view points | FPT University Vietnam |
| NamLH | Tester | Create and review TC for modules  Execute test  Report test result  Create and review Test Report | FPT University Vietnam |

Table 5.5.1 Human Resource

### Test management　テスト管理

#### Test management　テスト管理

Test management is practice of organizing and controlling the process and artifacts required for testing effort.

The general goal of test management is to allow teams to plan, develop, execute, and assess all testing activities within the overall software development effort. This includes coordinating efforts of all those involved in the testing effort, tracking dependencies and relationships among test assets and, most importantly, defining, measuring, and tracking quality goals.

* **Test planning** is the overall set of tasks that address the questions of why, what, where, and when to test. The reason why a given test is created is called a test motivator (for example, a specific requirement must be validated). What should be tested is broken down into many test cases for a project. Where to test is answered by determining and documenting the needed software and hardware configurations. When to test is resolved by tracking iterations (or cycles, or time period) to the testing.
* **Test authoring** is a process of capturing the specific steps required to complete a given test. This addresses the question of how something will be tested. This is where somewhat abstract test cases are developed into more detailed test steps, which in turn will become test scripts (either manual or automated).
* **Test execution** entails running the tests by assembling sequences of test scripts into a suite of tests. This is a continuation of answering the question of how something will be tested (more specifically, how the testing will be conducted).
* **Test reporting** is how the various results of the testing effort are analyzed and communicated. This is used to determine the current status of project testing, as well as the overall level of quality of the application or system.

### Defect management　欠陥管理

Defect management is crucial to closing the loop between requirements, implementation and verification and validation. Traditional defect tracking management, implemented in a standalone fashion, can no longer address the complexity and pace of change in modern software development. Defect management processes must be tightly interlinked with all of the other software development processes. The defect management process contains the following elements:

* **Defect Discovery** – Identification and reporting of potential defects. The defect tracking software must be simple enough so that people will use it, but ensure that the minimum necessary information is captured. The information captured here should be enough to reproduce the defect and allow development to determine root cause and impact.
* **Defect Analysis & Prioritization** – The development team determines if the defect report corresponds to an actual defect, if the defect has already been reported, and what the impact and priority of the defect is. Prioritization and scheduling of the defect resolution is often part of the overall change management process for the software development organization.
* **Defect Resolution –** Here the development team determines the root cause, implements the changes needed to fix the defect, and documents the details of the resolution in the defect management software, including suggestions on how to verify the defect is fixed. In organizations using [Software Product Lines](http://www.mks.com/challenges/product-lines) approaches, or other shared component approaches, defect resolution may need to be coordinated across multiple branches of development.
* **Defect Verification** – The build containing the resolution to the defect is identified, and testing of the build is performed to ensure the defect truly has been resolved, and that the resolution has not introduced side effects or regressions. Once all affected branches of development have been verified as resolved, the defect can be closed.
* **Defect Communication** – This encompasses automatic generation of defect metrics for management reporting and process improvement purposes, as well as visibility into the presence and status of defects across all disciplines of the software development team.

## Test environment　たテスト環境

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Screen Resolution | Browser | Purpose | Detail |
| Laptop HP 4430s |  | Google Chrome, Mozilla Firefox, Microsoft Edge | Device for create and execute test | Microsoft Windows 10 Professional |
| Laptop Dell Vostro 3560 | 1366x768 | Google Chrome, Mozilla Firefox, Internet Explorer | Device for create and execute test | Microsoft Windows 8.1 Professional |

### Hardware ハードウェア

Table 5.6.1 Hardware

### Software　ソフトウェア

|  |  |  |
| --- | --- | --- |
| Name | Purpose | Detail |
| Chrome | Executing test | Chrome 53.0 |
| Firefox | Executing test | Firefox |
| Microsoft Edge | Executing test | Microsoft Edge 38 |

# 

## Test Milestones　テストマイルストーン

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone Task | Effort (pd) | Start Date | End Date |
| Create Test Plan | 6 | October 10, 2016 | October 15, 2016 |
| Review & update TP | 2 | October 16, 2016 | October 16, 2016 |
| Create Unit Test case | 2 | October 17, 2016 | October 18, 2016 |
| Review & update UTC | 1 | October 19, 2016 | October 19, 2016 |
| Create Integration Test case | 3 | October 20, 2016 | October 25, 2016 |
| Review & Update Integration TC | 1 | October 26, 2016 | October 26, 2016 |
| Create System Test case | 3 | October 27 , 2016 | November 01 , 2016 |
| Review & Update System TC | 1 | November 02, 2016 | November 02, 2016 |
| Execute Unit Test Iterative 1 | 3 | November 03, 2016 | November 04, 2016 |
| Execute Integration test Iterative 1 | 3 | November 05, 2016 | November 09, 2016 |
| Execute System test Iterative 1 | 3 | November 10, 2016 | November 15, 2016 |
| Execute Unit Test Iterative 2 | 3 | November 10, 2016 | November 15, 2016 |
| Execute Integration test Iterative 2 | 3 | November 10, 2016 | November 15, 2016 |
| Execute System test Iterative 2 | 3 | November 10, 2016 | November 15, 2016 |

Table 5.7 Test Milestones

## Deliverables　納品物

| No | Deliverables | Responsibilities | Delivered Date |
| --- | --- | --- | --- |
|  | Test Plan | Test Lead | October 20th, 2016 |
|  | Unit Test cases | Tester | November 15th, 2016 |
|  | Common Test cases | Tester | November 15th, 2016 |
|  | Integration Test Cases | Tester | November 15th, 2016 |
|  | System Test cases | Tester | November 15th, 2016 |
|  | Defect log Management | Tester | December 04th, 2016 |
|  | Test reports | PM | December 04th, 2016 |

Table 5.8 Deliverables

## Test Case　テストケース

Refer to:

* + JFS\_Integration Test Case\_v1.0.xlsx
  + JFS\_System Test Case\_v1.0.xlsx
  + JFS\_Common Test Case\_v1.0.xlsx
  + JFS\_UnitTestCase\_GuestCommon\_v1.0.xlsx
  + JFS\_UnitTestCase\_MemberCommon\_v1.0.xlsx
  + JFS\_UnitTestCase\_MemberManagePersonalPage\_v1.0.xlsx
  + JFS\_UnitTestCase\_MemberManagePost\_v1.0.xlsx
  + JFS\_UnitTestCase\_AdminStatistic\_v1.0.xlsx
  + JFS\_UnitTestCase\_AdminManageReports\_v1.0.xlsx
  + JFS\_UnitTestCase\_AdminManagePosts\_v1.0.xlsx
  + JFS\_UnitTestCase\_AdminManageMembers\_v1.0.xlsx
  + JFS\_UnitTestCase\_AdminManageFeedback\_v1.0.xlsx
  + JFS\_UnitTestCase\_AdminManageCategory\_v1.0.xlsx

## Test Report　テストレポート

### Unit Test Report 単体テスト

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Function code** | **Passed** | **Failed** | **Untested** | **N** | **A** | **B** | **Total Test Cases** |
| 1 | Guest Register | 15 | 0 | 0 | 15 | 0 | 0 | 15 |
| 2 | Guest View Employer's Posts | 10 | 0 | 0 | 10 | 0 | 0 | 10 |
| 3 | Guest View Job Seeker's Posts | 10 | 0 | 0 | 10 | 0 | 0 | 10 |
| 4 | Guest Search Employer's Posts | 15 | 0 | 0 | 15 | 0 | 0 | 15 |
| 5 | Guest Search Job Seeker's Posts | 12 | 0 | 0 | 12 | 0 | 0 | 12 |
|  | **Sub total** | 62 | 0 | 0 | 62 | 0 | 0 | 62 |

Table 5.10.1 Guest Common

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Function code** | **Passed** | **Failed** | **Untested** | **N** | **A** | **B** | **Total Test Cases** |
| 1 | Member Login | 10 | 0 | 0 | 10 | 0 | 0 | 10 |
| 2 | Member Logout | 5 | 0 | 0 | 5 | 0 | 0 | 5 |
| 3 | Member Change Password | 8 | 0 | 0 | 8 | 0 | 0 | 8 |
| 4 | Member Search Employer's Posts | 15 | 0 | 0 | 15 | 0 | 0 | 15 |
| 5 | Member Search Job Seeker's Posts | 12 | 0 | 0 | 12 | 0 | 0 | 12 |
| 6 | Member Report Other Member's Post | 7 | 0 | 0 | 7 | 0 | 0 | 7 |
| 7 | Member feedback | 4 | 0 | 0 | 4 | 0 | 0 | 4 |
|  | **Sub total** | 61 | 0 | 0 | 61 | 0 | 0 | 61 |

Table 5.10.2 Member Common

### Common Test Report　一般テストレポート

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Module code** | **Pass** | **Fail** | **Untested** | **N/A** | **Number of**  **test cases** |
| 1 | Common | 12 | 0 | 0 | 0 | 12 |
| 2 | Security | 60 | 0 | 0 | 0 | 60 |
| 3 | UI | 26 | 0 | 0 | 0 | 26 |
|  | **Sub total** | 98 | 0 | 0 | 0 | 98 |

Table 5.10.3 Common Test report

### Integration Test Report　統合テストレポート

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Module code** | **Pass** | **Fail** | **Untested** | **N/A** | **Number of**  **test cases** |
| 1 | [Guest\_Function](file:///C:\Users\ThuyTuong\Documents\GitHub\CapstoneProject\1_Doc1_EN_Report\Report%205\JFS_Integration_TestCase_v1.1.xlsx#Guest_Function!A1) | 60 | 0 | 2 | 0 | 62 |
| 2 | [Admin Function](file:///C:\Users\ThuyTuong\Documents\GitHub\CapstoneProject\1_Doc1_EN_Report\Report%205\JFS_Integration_TestCase_v1.1.xlsx#Admin_Function!A1) | 146 | 0 | 0 | 0 | 146 |
| 3 | [Member Function](file:///C:\Users\ThuyTuong\Documents\GitHub\CapstoneProject\1_Doc1_EN_Report\Report%205\JFS_Integration_TestCase_v1.1.xlsx#Member_Function!A1) | 154 | 0 | 0 | 0 | 154 |
| 4 | [System Function and Transaction](file:///C:\Users\ThuyTuong\Documents\GitHub\CapstoneProject\1_Doc1_EN_Report\Report%205\JFS_Integration_TestCase_v1.1.xlsx#System_FunctionAndTransaction!A1) | 68 | 0 | 0 | 0 | 68 |
|  | **Sub total** | 428 | 0 | 2 | 0 | 430 |

Table 5.10.4 Integration Report

### System Test Report　システムテストレポート

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Module code** | **Pass** | **Fail** | **Untested** | **N/A** | **Number of test cases** |
| 1 | Common | 24 | 2 | 2 | 0 | 28 |
| 2 | Homepage | 22 | 0 | 0 | 0 | 22 |
| 3 | Member's Account | 68 | 0 | 0 | 0 | 68 |
| 4 | Create Employer's Post | 66 | 0 | 0 | 0 | 66 |
| 5 | Create Job Seeker's Post | 62 | 0 | 0 | 0 | 62 |
| 6 | Post Detail | 46 | 0 | 0 | 0 | 46 |
| 7 | Matching Post | 48 | 0 | 0 | 0 | 48 |
| 8 | Admin Fuction | 150 | 0 | 0 | 0 | 150 |
|  | **Sub total** | 486 | 2 | 2 | 0 | 490 |

Table 5.10.5 System Test Report

### Test Report　テストレポート

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Phase** | **Round 1** | | | | **Round 2** | | | |
| Pass | Fail | Untested | N/A | Pass | Fail | Untested | N/A |
| **Unit Test** | 450 | 0 | 0 | 0 | 450 | 0 | 0 | 0 |
| **Integration Test** | 480 | 46 | 0 | 0 | 526 | 0 | 0 | 0 |
| **System Test** | 508 | 4 | 0 | 0 | 512 | 0 | 0 | 0 |
| **Total** | **1438** | **50** | **0** | **0** | **1488** | **0** | **0** | **0** |

Table 5.10.5 Test Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Defects** | **Logged** | **%Closed** | **%Invalid** | **%Open** |
| **Critical** | 0 | 0% | 0% | 0% |
| **High** | 2 | 0% | 0% | 0% |
| **Medium** | 15 | 100% | 0% | 0% |
| **Low** | 33 | 100% | 0% | 0% |
| **Total** | **50** | **100%** | **0%** | **0%** |

Table 5.10.5.1 Defect report