|  |
| --- |
|  |
| Capstone Project Document |

KARYWELL

Test Plan

|  |  |  |
| --- | --- | --- |
| **Karywell** | | |
| **Group Members** | Le Van Quy Hoang | SE90184 |
| Tran Dinh Hoang Huy | SE90201 |
| Nguyen Khac Hoang | SE02564 |
| Kieu Cao Khanh | SE02960 |
| Nguyen Van Quyen | SE02884 |
| Nguyen Thi Hong Nhung | SE02437 |
| **Supervisor** | Mr. Tran Binh Duong | |
| **Project code** | KW | |

**- Hanoi, 12/2015 -**

Contents

[I. TEST PLAN 4](#_Toc437846215)

[1. Introduction 4](#_Toc437846216)

[1.1. Purpose 4](#_Toc437846217)

[1.2. Scope of testing 4](#_Toc437846218)

[1..3. Contraints 4](#_Toc437846219)

[1.4. Risks List 5](#_Toc437846220)

[1.2. Requirement For Test 5](#_Toc437846221)

[1.2.1. Features to be tested 5](#_Toc437846222)

[1.2.2. Features not to be tested 7](#_Toc437846223)

[1.3. Acceptance Test Criteria 7](#_Toc437846224)

[1.4. Test strategy 8](#_Toc437846225)

[1.4.1. Test type 8](#_Toc437846226)

[a. Funtion Testing 8](#_Toc437846227)

[b. User Interface Testing 8](#_Toc437846228)

[c. Performance Profiling 9](#_Toc437846229)

[d. Regression Testing 10](#_Toc437846230)

[1.4.2. Test stages 11](#_Toc437846231)

[a. Unit Test 11](#_Toc437846232)

[b. Integration Test 11](#_Toc437846233)

[c. System Test 11](#_Toc437846234)

[d. Acceptance Test 12](#_Toc437846235)

[1.4.3 Test Model 12](#_Toc437846236)

[1.4.4. Testing Process 14](#_Toc437846237)

[1.5. Resource 14](#_Toc437846238)

[1.5.1. Human resource 14](#_Toc437846239)

[1.5.2. Environment recource 15](#_Toc437846240)

[1.6. Test Milestones 15](#_Toc437846241)

[1.7. Deliverables 16](#_Toc437846242)

[II. TEST CASE 17](#_Toc437846243)

[1. Unit test 17](#_Toc437846244)

[2. Integration test 17](#_Toc437846245)

[3. System test 17](#_Toc437846246)

[III. TEST REPORT 17](#_Toc437846247)

[1. Unit test 17](#_Toc437846248)

[2. Integration test 22](#_Toc437846249)

[3. System test 23](#_Toc437846250)

[3.1. Admin 23](#_Toc437846251)

[3.2. Store 23](#_Toc437846252)

[3.3. Shipper 24](#_Toc437846253)

[3. Bug control chart 24](#_Toc437846254)

[IV. CHECK LIST 26](#_Toc437846255)

# I. TEST PLAN

## Introduction

### 1.1. Purpose

The purpose of testing is to qualify a software program’s quality by measuring its attributes and capabilities against expectations and applicable standards. Gaining confidence in and providing information about the level of quality. And to make sure that the end result meets the business and user requirements

### 1.2. Scope of testing

The purpose of this document is to outline the test strategy and overall test approach for the Karywell project. This includes test methodologies, traceability, resources required, testing environment and estimated schedule.

* **Stages of testing:**

There are 4 phases in Testing Process: Unit testing, Integration testing, System testing and Acceptance testing.

* **Types of testing:**

The test team has to test the following type on both Google Chrome and Firefox

* Function test
* GUI test
* Regression test
* Performance test
* **Range of testing:**
* Team performs all functions defined in the SRS based on the approved version.

### 1.3. Contraints

The following constraints may apply when testing is performed on system:

* Deadline for testing only can be met if development progress is on time
* Test execution can be performed when system passes Unit Test Inspection
* At least one round of testing must be performed for requirements

### 1.4. Risks List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Risks** | **Source** | **Probability** | **Contingency** |
| 1 | Test phrases is not completed due to testers’ lack of experience | Human | Medium | Training |
| 2 | Requirements are not base-lined on time | Human | High | KW team must follow-up to get requirements base-lined on time |
| 3 | Late response on issues | Human | Medium | The feedback must be provided in 3 working day when an issue is raised |
| 3 | Virus effectively | Device | High | High light this everyone in team, update the newest version for anti virus program |

## 2. Requirement For Test

### 2.1. Features to be tested

a. Admin

|  |  |  |
| --- | --- | --- |
| **No** | **Usercase Name** | **Features to be tested** |
| 1 | UC-101: Login | Login |
| 2 | UC-102: Logout | Logout |
| 3 | UC- 105: Add Store | Add Store |
| 4 | UC- 106: Search Store | Search Store |
| 5 | UC- 107: View Store’s list | View Store’s list |
| 6 | UC- 108: Block/Unblock store | Block/Unblock store |
| 7 | UC- 109: View Store’s details | View Store’s details |
| 8 | UC- 111: Confirm Store’s payment | Confirm Store’s payment |
| 9 | UC- 112: View transaction history of Store | View transaction history of Store |
| 10 | UC- 113: Add Shipper | Add Shipper |
| 11 | UC- 114: View Shipper’s list | View Shipper’s list |
| 12 | UC- 115: View Order’s list | View Order’s list |
| 13 | UC- 116: Search Shipper | Search Shipper |
| 14 | UC- 117: Search Order | Search Order |
| 15 | UC- 118: View Shipper’s profile | View Shipper’s profile |
| 16 | UC- 119: View Order’s details | View Order’s details |
| 17 | UC- 120: Update User’s Profile | Update User’s Profile |
| 18 | UC- 121: Update Store’s Information | Update Store’s Information |
| 19 | UC- 124: Assign Task | Assign Task |
| 20 | UC- 125: View Dashboard | View Dashboard |
| 21 | UC- 126: View IssueBox | View IssueBox |
| 22 | UC-127: Resovle Issue | Resovle Issue |
| 23 | UC- 128: View Task’s list | View Task’s list |
| 24 | UC- 129: View Report | View Report |
| 25 | UC- 130: Search Task | Search Task |
| 26 | UC- 131: Search Transaction | Search Transaction |
| 27 | UC- 132: Search Issue | Search Issue |

b. Store

|  |  |  |
| --- | --- | --- |
| **No** | **Usercase Name** | **Features to be tested** |
| 1 | UC-201: Login | Login |
| 2 | UC-202: Logout | Logout |
| 3 | UC-206: View profile | View profile |
| 4 | UC-208: Create order | Create order |
| 5 | UC-210: Edit Order | Edit Order |
| 6 | UC-211: Add goods | Add goods |
| 7 | UC-212: Edit goods | Edit goods |
| 8 | UC-213: Delete goods | Delete goods |
| 9 | UC-214: View order’s history | View order’s history |
| 10 | UC-215: Search orders | Search orders |
| 11 | UC-216: Cancel order | Cancel order |
| 12 | UC-217: Delete order | Delete order |
| 13 | UC-218: View order’s details | View order’s details |
| 14 | UC-219: View transaction history | View transaction history |
| 15 | UC-220: Search transaction history | Search transaction history |
| 16 | UC-221: View notifications | View notifications |
| 17 | UC-201: View dashboard | View dashboard |

c. Shipper

|  |  |  |
| --- | --- | --- |
| **No** | **Usercase Name** | **Features to be tested** |
| 1 | UC-301: Login | Login |
| 2 | UC-302: Logout | Logout |
| 3 | UC-303: View Profile | View Profile |
| 4 | UC-307: Confirm Code | Confirm Code |
| 5 | UC-308: Grab Order | Grab Order |
| 6 | UC-309: View Task List | View Task List |
| 7 | UC-3011: View Task’s Detail | View Task’s Detail |
| 8 | UC-313: Send Issue’s Information | Send Issue’s Information |
| 9 | UC-314: Change Working Status | Change Working Status |
| 10 | UC-315: View History | View History |
| 11 | UC-316: Continue When Pending | Continue When Pending |

### 2.2. Features not to be tested

Out of scope features will not to be tested.

## 3. Acceptance Test Criteria

|  |  |  |
| --- | --- | --- |
| **No** | **Test Stages** | **Qualified ratios** |
| **1** | Unit Test | - To pass this stage, all unit test cases must be tested and passed 100%. All defects should be fixed and re-tested.  - Weight defects/KLOC: 8 - 9 bugs/1KOLC. |
| **2** | Integration Test | - To pass this stage, all unit test cases must be tested and passed 100%. All defects should be fixed and re-tested.  - Weight defects/KLOC: 2 - 4 bugs/1KOLC. |
| **3** | System Test | - To pass this stage, all unit test cases must be tested and passed 100%. All defects should be fixed and re-tested.  - Weight defect/KLOC: 2 - 4 bugs/1KLOC. |

## 4. Test strategy

### 4.1. Test type

#### a. 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. The goals of these tests are to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules. This type of testing is based upon black box techniques; that is verifying the application and its internal processes by interacting with the application via the Graphical User Interface (GUI) and analyzing the output or results. Identified below is an outline of the testing recommended for each application:

|  |  |
| --- | --- |
| 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. |
| Completion Criteria: | * • All planned tests have been executed. * • All identified defects have been addressed. |
| 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. The goal of UI testing is to ensure that the User Interface provides the user with the appropriate access and navigation through the functions of the target-of-test. In addition, UI testing ensures that the objects within the UI function as expected and conform to corporate or industry standards.

|  |  |
| --- | --- |
| **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 |

#### Performance Profiling

* Performance profiling is a performance test in which response times, transaction rates, and other time-sensitive requirements are measured and evaluated. The goal of Performance Profiling is to verify performance requirements have been achieved. Performance profiling is implemented and executed to profile and tune a target-of-test's performance behaviors as a function of conditions such as workload or hardware configurations.

|  |  |
| --- | --- |
| **Test Objective:** | * Verify performance behaviors for designated transactions or business functions under the following conditions: * Normal anticipated workload * Anticipated worst case workload |
| **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, see Special Considerations below). |
| **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: * “Drive transactions” directly to the server, usually in the form of Structured Query Language (SQL) calls. * Create “virtual” user load to simulate many clients, usually several hundred. Remote Terminal Emulation tools are used to accomplish this load. This technique can also be used to load the network with “traffic”. * 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. |

#### d. Regression Testing

Regression testing is a necessary maintenance activity aimed at showing that code has not been adversely affected by changes. Define trigger for regression test (applied for maintenance projects), period and scope of regression test.

|  |  |
| --- | --- |
| **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. * 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:** | N/A |

### Test stages

#### a. Unit Test

The programmer usually does unit Test. This phase should be taken as soon as possible in the programming phase and throughout the software development cycle. The purpose of Unit Test is to ensure that the information processed and exported from the unit is accurate, in correlation with input data and functions of unit.

As well as other tests, Unit Test requires advanced preparation for situations (test case) or scenarios (script) in which specified input, the steps taken and the data expected to come out. The situations (test cases) and scenarios (scripts) should be retained for reuse.

#### Integration Test

Integration test combines the components of an application and checks it as a complete application. While the unit test checks components and individual units, the Integration Test matches them together and checks the communication between them.

Integration Test has two main goals:

* Detect communication errors occur between units.
* Integrate single units into subsystem and finally a complete system to prepare for the test at the system level.

Except for a few exceptions, Integration Test should only be done on the unit, which has been carefully checked previously by Unit Test and all levels Unit error has been corrected.

#### System Test

Purpose of System Test is to check the design and whole system after integration, whether they satisfy requirements or not.

System Test starts when all the parts of the software were successfully integrated. The key difference between Integration Test and System Test is that the System Test focuses on behaviors and system errors, while Integration Test focuses on communication between individuals or objects when they work together. Usually we have to perform Unit Test and Integration Test to make sure all Units and the interaction between them work correctly before performing System Test.

The planning for System Test should start at formation and requirement analysis phase. System Test checks all functional behavior of software as well as the requirements of quality such as reliability, convenience of use, performance and security.

#### Acceptance Test

Usually, the phase after System Test is Acceptance Test that is done by customers. The purpose of Acceptance Test is to demonstrate that the software meets all requirements of the customer, and whether customer accepts it or not.

Acceptance Test has a very important significance, although in most cases, the tests of System Test and Acceptance Test are almost the same, but the nature and implementation method are very different.

Acceptance Test is usually associated with a group of services and attached documents, such as the installation guide, user guide…etc. All attached documents must be updated and checked strictly.

|  |  |  |  |
| --- | --- | --- | --- |
| Stage  Type | Unit | Intergation | System |
| Funtion | x | x | x |
| GUI |  | x | x |
| Performance |  |  | x |
| Regresstion | x | x | x |

### 4.3 Test Model

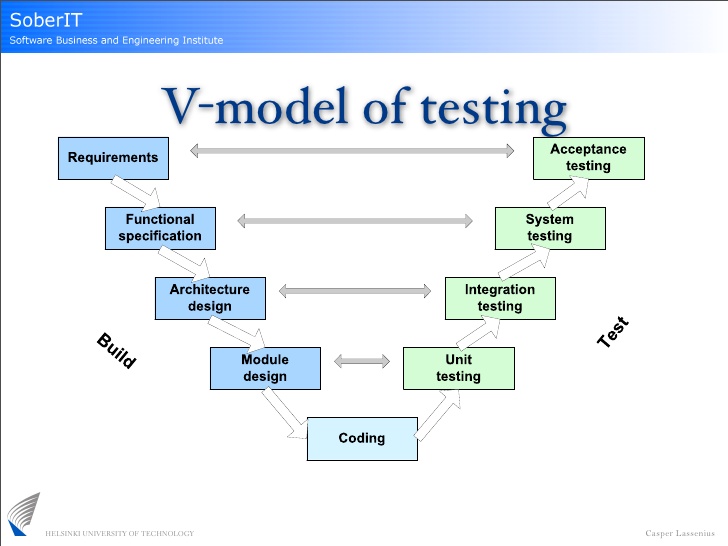


Figure 1: V-model

Due to requirement of project must be always suitable with user, we choose V-model to implement testing process. With V-Model, software development is separated into two appropriate phase’s groups: development and testing. In this model, the verification and validation 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, which are found at early stages even, may be in the development phase before application being tested

### 4.4. Testing Process

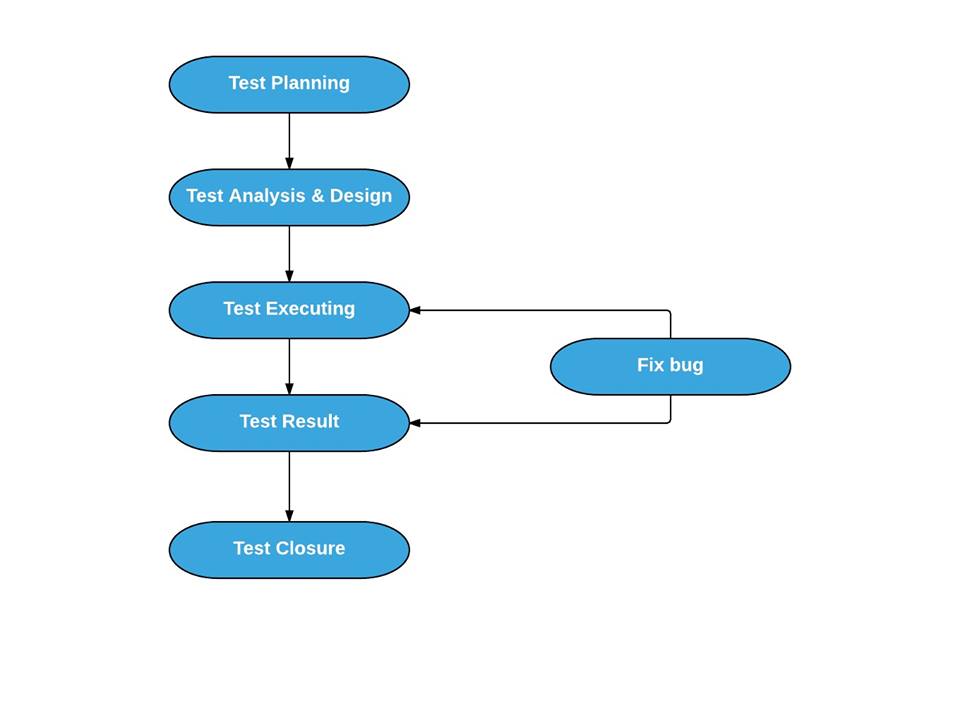


Figure 2: Test Process

## 5. Resource

### 5.1. Human resource

|  |  |  |
| --- | --- | --- |
| Name | Roles | Responsibility |
| NhungNTH | Test leader | Manage Test resource and assign test tasks  Create TP ST IT TC Review TC  Execute test |
| HoangLVQ | PM + Developer | Review test plan  Review test report  Create UT & execute test |
| HoangNK | Technical leader | Create UT & execute test  Report test result |
| HuyTDH | Developer | Create UT & execute test  Report test result |
| QuyenNV | Developer | Create UT & execute test  Report test result |
| KhanhKC | Developer | Create UT & execute test  Report test result |

### 5.2. Environment recourse

|  |  |
| --- | --- |
| Software | Hardware |
| Window 7  Google Chrome 46.0.2490.86 m  Fire fox 8.0.1 | IPhone 5 iOS version 7.1.1  Google Nexus 5 Android version 6.0  Laptop Window 7 |

## 6. Test Milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone Task** | **Effort (pd)**  **(pd)** | **Start Date** | **End Date** |
| * **Increment 1** | | | |
| Create test plan v1.0 | 4 | 22/10/2015 | 26/10/2015 |
| Create system test cases | 4 | 26/10/2015 | 30/11/2015 |
| Create integration test case | 3 | 01/11/2015 | 03/11/2015 |
| Execute test | 4 | 04/11/2015 | 08/11/2015 |
| Test report v1.0 | 1 | 08/11/2015 | 08/11/2015 |
| * **Increment 2** | | | |
| Create test plan ver2.0 | 1 | 09/11/2015 | 09/11/2015 |
| Update test cases | 4 | 09/11/2015 | 13/11/2015 |
| Execute test | 12 | 16/11/2015 | 30/11/2015 |
| Create and execute unit test cases | 7 | 23/11/2015 | 01/12/2015 |
| Test report v2.0 | 1 | 30/11/2015 | 30/11/2015 |
| * **Increment 3** | | | |
| Execute test | 4 | 09/12/2015 | 13/12/2015 |
| Submit final report of testing | 1 | 13/12/2015 | 13/12/2015 |

## 1.7. Deliverables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Deliverables** | **Delivered Date** | **Delivered by** | **Delivered to** |
| * **Increment 1** | | | | |
| 1 | Test plan ver1.0 | 08/11/2015 | NhungNTH | HoangLVQ |
| 2 | System test cases ver1.0 | 08/11/2015 | NhungNTH | HoangLVQ |
| 3 | Integration test cases ver1.0 | 08/11/2015 | NhungNTH | HoangLVQ |
| * **Increment 2** | | | | |
| 5 | Test plan ver2.0 | 20/11/2015 | NhungNTH | HoangLVQ |
| 6 | Unit test cases and result | 01/12/2015 | Developer | HoangLVQ |
| 7 | System test case ver2.0 | 13/11/2015 | NhungNTH | HoangLVQ |
| 8 | System result 1.0 | 01/12/2015 | NhungNTH | HoangLVQ |
| 9 | Integration test cases ver2.0 | 14/11/2015 | NhungNTH | HoangLVQ |
| 10 | Integration result | 01/12/2015 | NhungNTH | HoangLVQ |
| * **Increment 3**   **31/5/2015**  **MaiNT** | | | | |
| 11 | System result 2.0 | 13/12/2015 | NhungNTH | HoangLVQ |
| 12 | Final report of testing | 13/12/2015 | HoangLVQ | Supervisor |

# II. TEST CASE

## 1. Unit test

* Reference: KW\_UnitTest\_QuyenNV.xlsx
* Reference: KW\_UnitTest\_HuyTDH.xlsx
* Reference: KW\_UnitTest\_HoangNK.xlsx
* Reference: KW\_UnitTest\_KhanhKC.xlsx
* Reference: KW\_UnitTest\_HoangLVQ.xlsx

## 2. Integration test

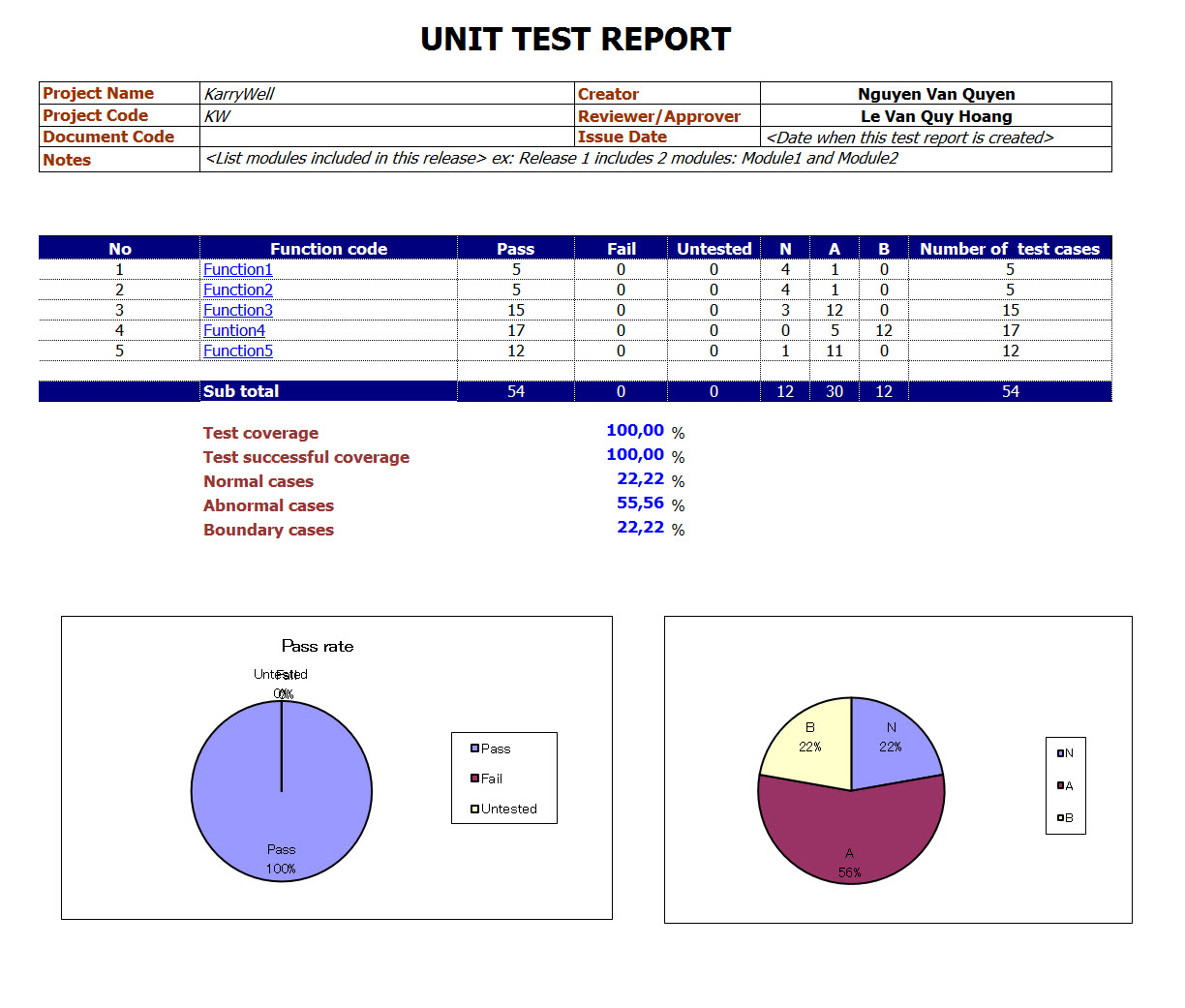
* Reference: KW\_IntegrationTest.xlsx

## 3. System test

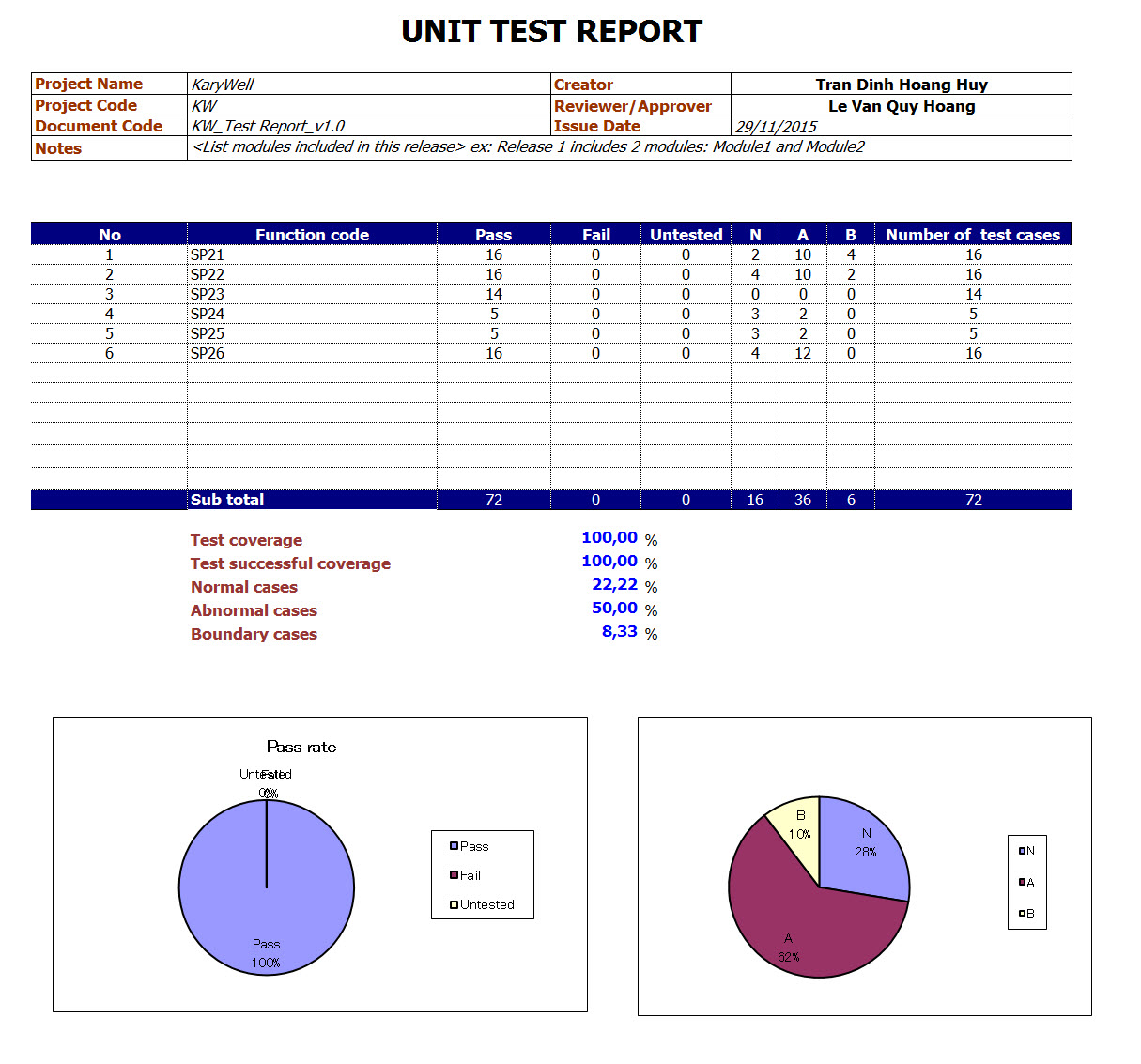
* Reference: KW\_Admin\_TestCase.xlsx
* Reference: KW\_Store\_TestCase.xlsx
* Reference: KW\_Shipper\_TestCase.xlsx

# III. TEST REPORT

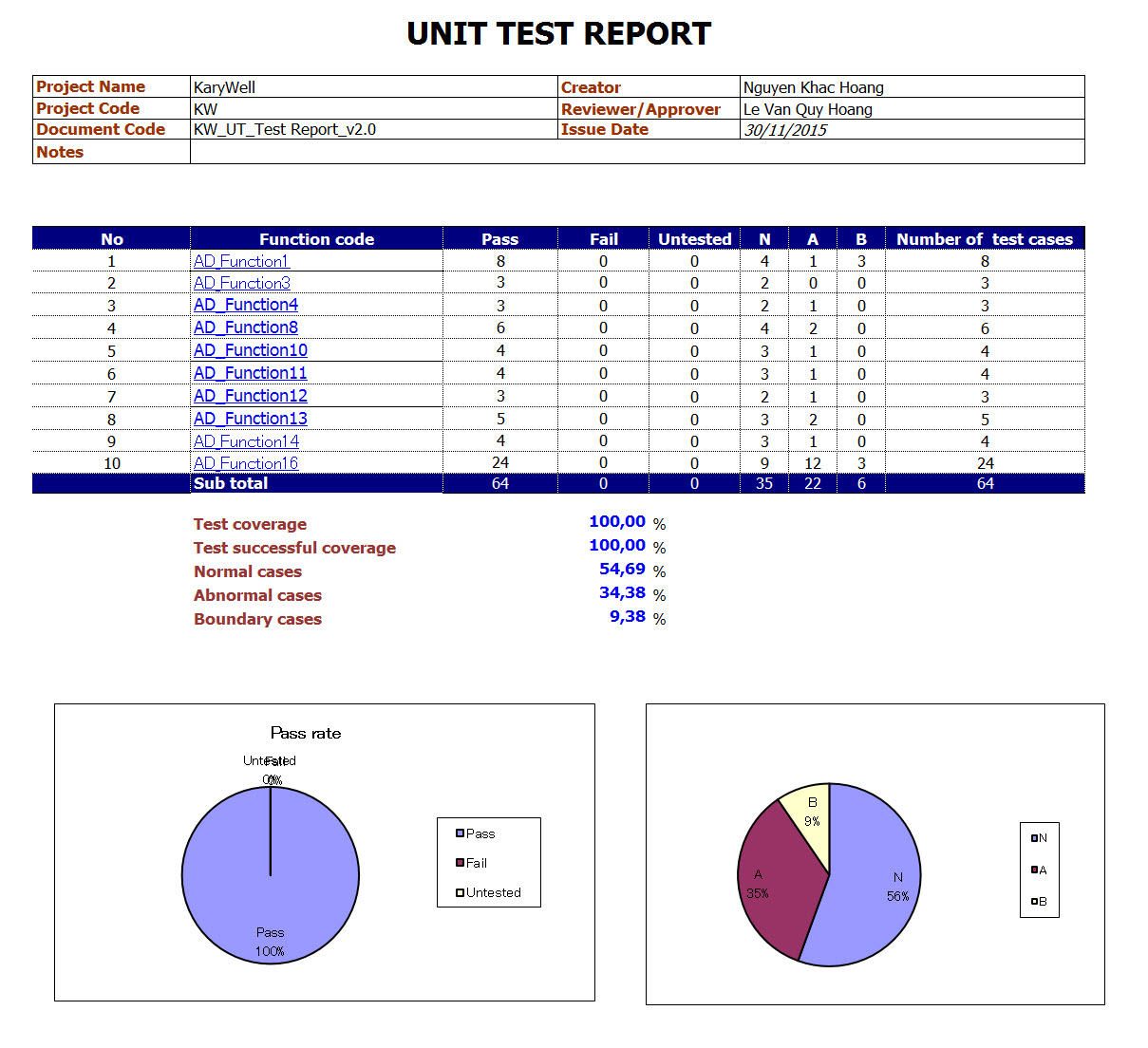
# 1. Unit test

- KW\_UnitTest\_QuyenNV: 

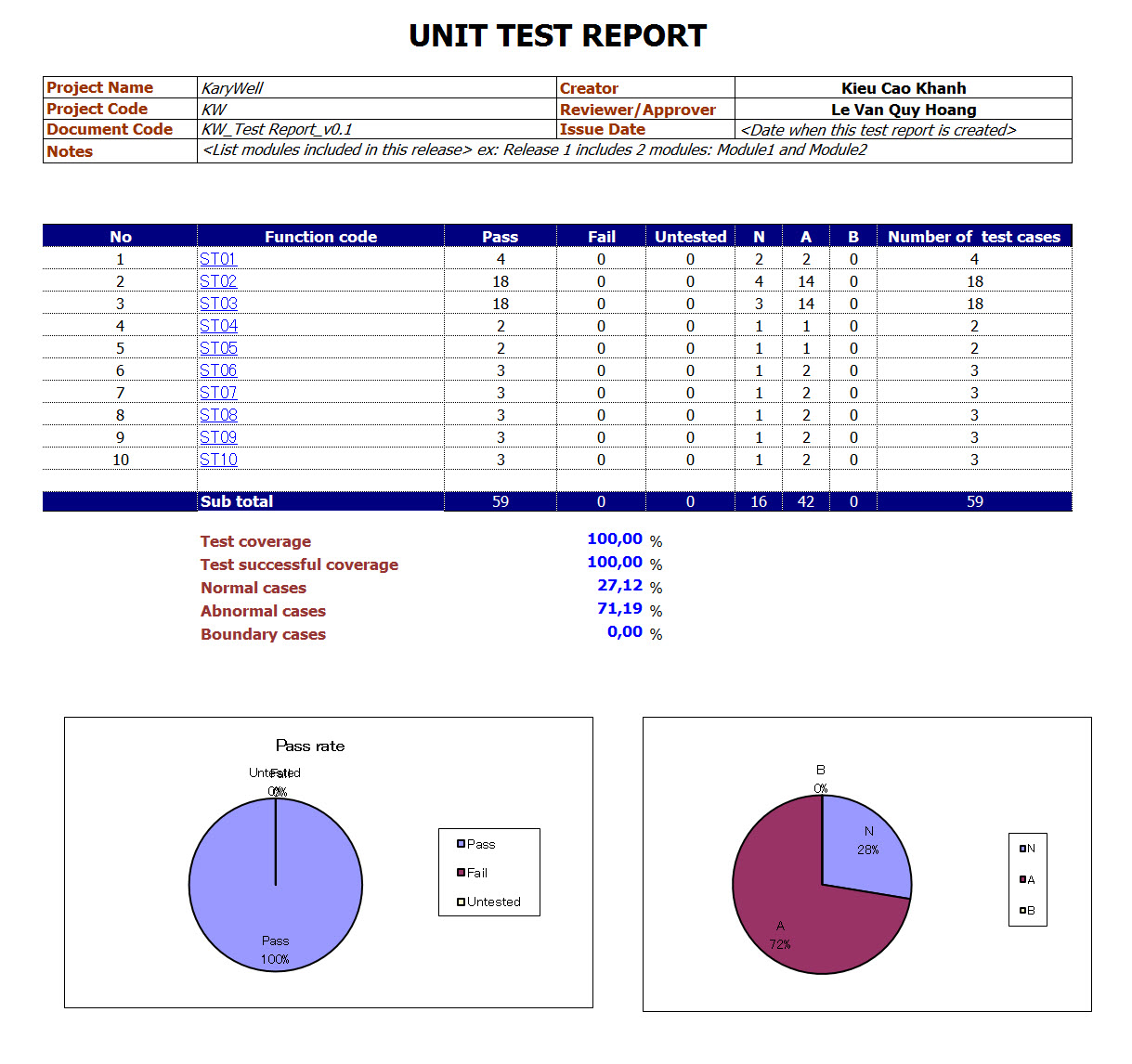
* KW\_UnitTest\_HuyTDH



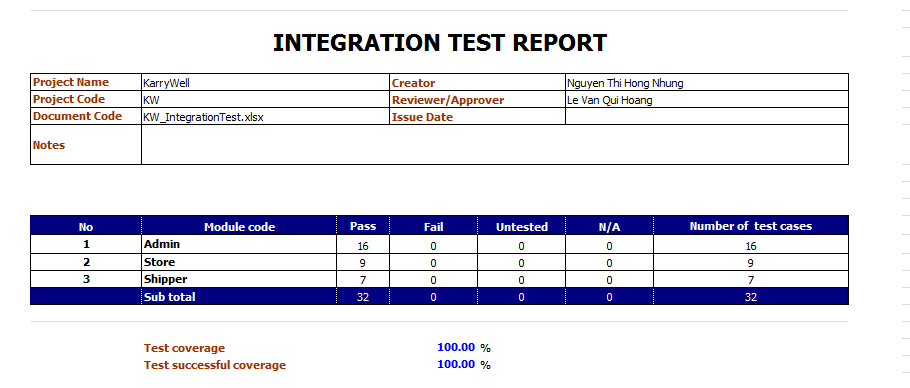
* KW\_UnitTest\_HoangNK



* KW\_UnitTest\_KhanhKC

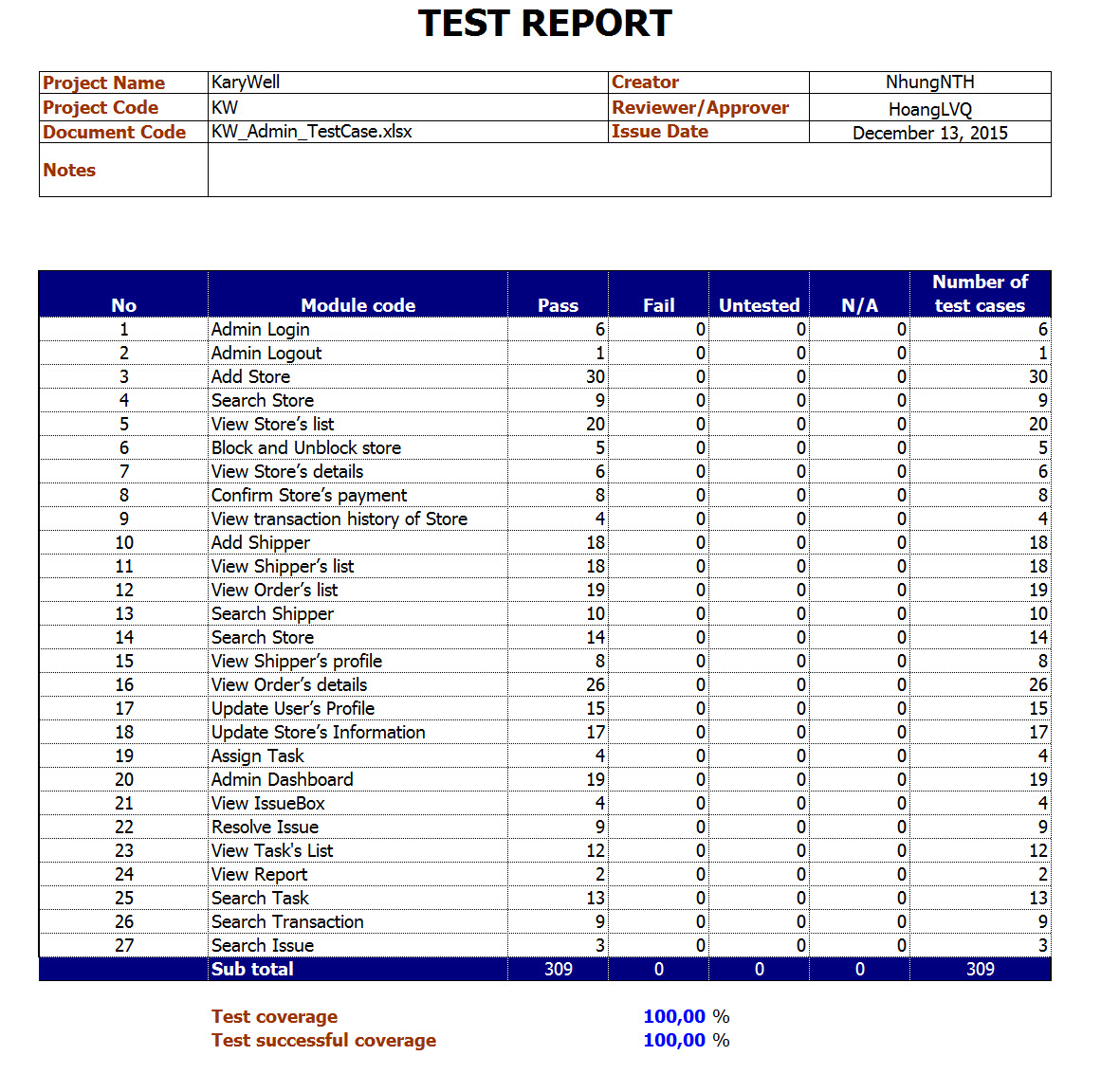


## Integration test

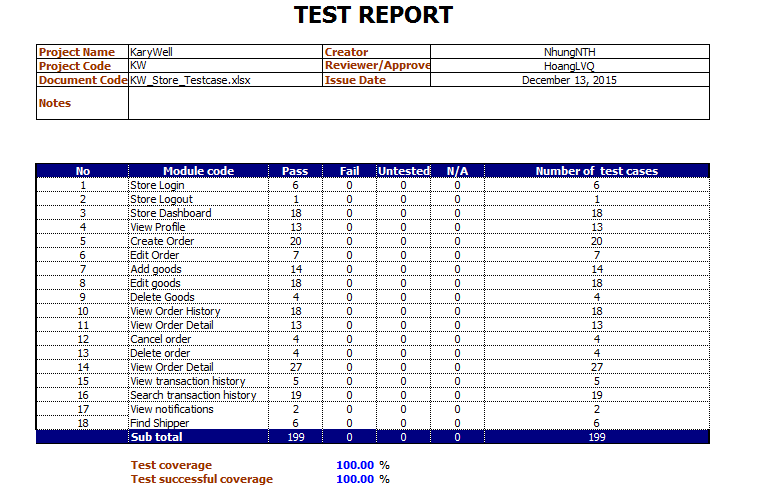


## 3. System test

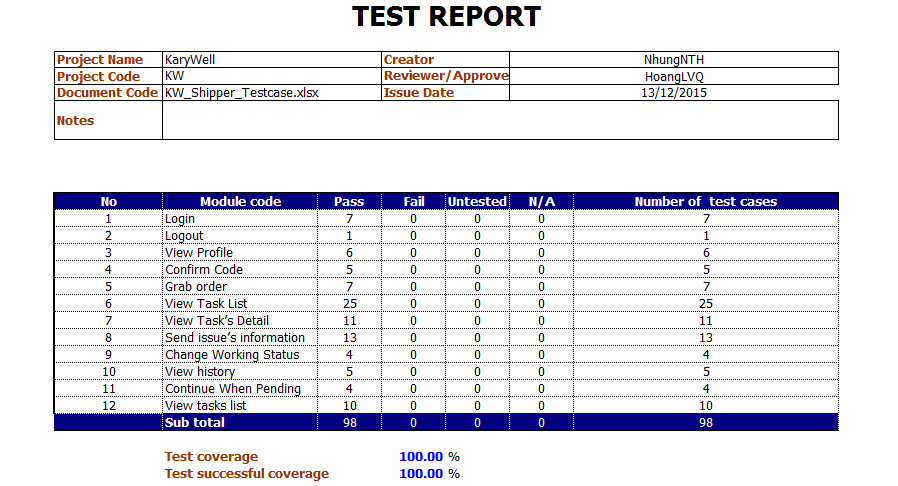
### 3.1. Admin



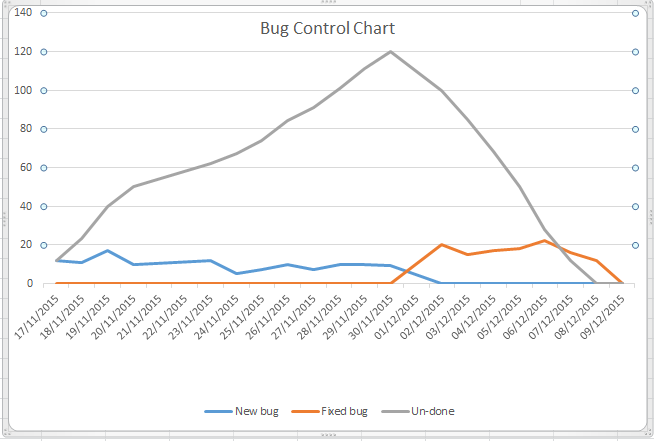
### 3.2. Store



### 3.3. Shipper



## Bug control chart



# IV. CHECK LIST

