

**CAPSTONE PROJECT REPORT**

**Report 5 – Software Test Documentation**

– Hanoi, August 2022 –

**Table of Contents**

[I. Record of Changes](#_heading=h.gjdgxs) **3**

[II. Testing Documentation](#_heading=h.30j0zll) **4**

[1. Scope of Testing](#_heading=h.1fob9te) 4

[1.1 Testing Targets](#_heading=h.xmo74oaea5tr) 4

[1.2 Testing Levels](#_heading=h.hg5hxwexb2co) 4

[2. Test Strategy](#_heading=h.3znysh7) 5

[2.1 Test Model](#_heading=h.viae0iid2jr1) 5

[2.2 Testing Types](#_heading=h.2et92p0) 5

[2.3 Supporting Tools](#_heading=h.4d34og8) 6

[3. Test Plan](#_heading=h.2s8eyo1) 6

[3.1 Human Resources](#_heading=h.17dp8vu) 6

[3.2 Test Environment](#_heading=h.3rdcrjn) 7

[3.3 Test Milestones](#_heading=h.26in1rg) 7

[4. Test Cases](#_heading=h.lnxbz9) 8

[4.1 Unit testing](#_heading=h.9chd8na4as92) 8

[4.2 Integration Testing & System Testing](#_heading=h.9oez6k2hf0yh) 8

[4.3 Acceptance Testing](#_heading=h.4ifo69hfhdju) 8

[5. Test Reports](#_heading=h.35nkun2) 8

# I. Record of Changes

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| 6/17/2022 | A | NhungNTH | Create Test Documentation |
| 8/18/2022 | M | NhungNTH | Update Test Documentation |

\*A - Added M - Modified D - Deleted

# II. Testing Documentation

## 1. Scope of Testing

### 1.1 Testing Targets

***Features – Functions:***All features – defined in section 3.1 of [SRS] document version 1.0

***Non-functional:***

* *User Interface*
  + Font for regular text for iOS is San Francisco, for Android is Roboto
  + Color of buttons: FEC54B
  + Color of error text: FF6442
  + All pages that display a lot of content in repairer and customer app must be scrollable, in admin app must be pageable
  + Text is left aligned except for the title of the confirmation popup
  + The letters in the title are in bold
  + All password text fields should have a hide/show password icon to hide/show password which user entered.
* *Performance Requirements*
  + UI change time when user ticks on buttons/checkbox/combobox/slider or writes on textbox does not exceed 0.5s.
  + Waiting time to create, update, delete, load and save a new thing is about 3s.
  + Waiting time to receive a response of any search does not exceed 2s.
  + An empty state (and especially an error state) should not be an error, instead it will tell the user exactly what action is required so that the content can appear and the application will work properly
* *Security Requirements*
  + Database should provide the capability of data protection from unauthorized users and applications.
  + A user MUST authenticate to the device using a password containing a minimum of 6 characters and maximum of 10 characters, not containing spaces, with at least 1 number.

### 1.2 Testing Levels

We have 4 levels of testing: Unit Test, Integration Test, System Test, and Acceptance Test.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Level** | **In charge** | **Time** | **Focuses** | **Acceptance criteria** |
| Unit Test | Developers | Implementing | Searches for defects in and verifies the functioning of software components that are separately testable. | 100% branch conditions of the entire source code must be covered and executed successfully without unforeseeable exceptions. |
| Integration Test | Testers | After accepting UT | Ensure all components incorporate well. | Ensure the data transmitted/received between controller layer and service layer, repository, finally, the database is exchanged correctly. |
| System Test | Testers | After accepting IT | Integrate different modules and test the interface between them to check data integrity. | All logic flows work exactly as designed.  Non-functional requirements and system interfaces perform properly in the production. |
| Acceptance Test | End-users | After release full ST | Pure functional testing to check system behavior with real data. | The system meets customer business needs, product availability. |

## 2. Test Strategy

### 2.1 Testing Types

|  |  |  |
| --- | --- | --- |
| **No** | **Level** | **Description** |
| 1 | Functional Testing (Black-box Testing) | Functional Testing is a type of software testing that validates the software system against the functions.  The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements. |
| 2 | White-box Testing | White-box testing is a method of software testing that tests internal structures or workings of an application. The tester chooses inputs to exercise paths through the code and determine the expected outputs. White-box testing is applied at the unit, integration levels of the software testing process. |
| 3 | UI/UX testing | - UI testing verifies a user’s interaction with the software. The goal of UI/UX testing is to ensure that the UI/UX provides the user with appropriate access and navigation through the functions of the target-of-test. In addition, UI testing ensures that the objects within the UI/UX function as expected and conform to the requirement.  - UI/UX test will be performed fully on all screens.  - UI/UX test targets to cover the verification of the overall look and feel of the system including initial position, font, text size, color, focus, initial button, tab order, label, screen sizes, sentences width and animation.  - Check all the UI elements for color, size, position, width, length and acceptance of characters or numbers. |

### 2.2 Supporting Tools

|  |  |  |  |
| --- | --- | --- | --- |
| **Purpose** | **Tool** | **Provider** | **Version** |
| Test API | Postman-tool | Postman, Inc | v9.26.0 |
| Integration test | App Tester | Google | v0.128 |

## 3. Test Plan

### 3.1 Human Resources

|  |  |  |
| --- | --- | --- |
| **Worker/Doer** | **Role** | **Specific Responsibilities/Comments** |
| Nguyễn Thị Hồng Nhung | Tester | - Manage test resources and assign test tasks  - Create a test plan  - Create test cases  - Create test reports  - Create unit test  - Execute test  - Test log report |
| Phạm Chí Dũng | Developer | - Fix bugs |
| Bùi Đức Thắng | Developer | - Fix bugs |
| Vũ Ngọc Thịnh | Developer | - Fix bugs |

### 3.2 Test Environment

|  |  |  |  |
| --- | --- | --- | --- |
| **Purpose** | **Tool** | **Provider** | **Version** |
| Setup and configuration for test front end website | Google Chrome | Google | 87.0.4280.67 |
| Setup and configuration for test notification and send/receive message | Firebase | Google |  |
| Tracking any database  transaction (evaluate the  performance of the SQL  queries) | My SQL Server | Oracle | 8.0 |
| Setup and configuration for test back end code | Spring Boot | VMware | 4.10.0 |
| Setup and configuration for test front end code | Android | Google | 10 |

### 3.3 Test Milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone Task** | **Effort (md)** | **Start Date** | **End Date** |
| **Sprint 1** | | | |
| Create Test Plan | 2 | 5/23/2022 | 5/24/2022 |
| Integration test Part 1 | 10 | 5/25/2022 | 6/7/2022 |
| System Test Part 1 | 1 | 6/8/2022 | 6/8/2022 |
| Unit Testing Part 1 | 4 | 5/25/2022  6/9/2022 | 5/27/2022  6/9/2022 |
| UI/UX testing | 1.5 | 6/9/2022 | 6/10/2022 |
| **Sprint 2** | | | |
| Integration test Part 2 | 7 | 6/15/2022 | 6/24/2022 |
| System Test Part 2 | 4 | 6/27/2022 | 6/27/2022 |
| Unit Testing Part 2 | 4 | 6/15/2022  6/30/2022 | 6/17/2022  6/30/2022 |
| UI/UX testing | 1.5 | 6/30/2022 | 7/1/2022 |
| **Sprint 3** | | | |
| Integration test Part 3 | 9 | 7/6/2022 | 7/18/2022 |
| System Test Part 3 | 2 | 7/19/2022 | 7/20/2022 |
| Unit Testing Part 3 | 4 | 7/6/2022  7/20/2022 | 7/8/2022  7/20/2022 |
| UI/UX testing | 1.5 | 7/21/2022 | 7/22/2022 |
| Update Test Plan | 0.5 | 7/22/2022 | 7/22/2022 |
| **Sprint 4** | | | |
| Integration test Part 4 | 8 | 7/27/2022 | 8/3/2022 |
| System Test Part 4 | 2 | 8/4/2022 | 8/9/2022 |
| Acceptance Testing | 1 | 8/10/2022 | 8/10/2022 |
| Unit Testing Part 4 | 4 | 7/27/2022  8/11/2022 | 7/29/2022  8/11/2022 |
| UI/UX testing | 1.5 | 8/11/2022 | 8/12/2022 |
| **Close** | | | |
| System Testing | 1 | 8/14/2022 | 8/14/2022 |
| Acceptance Testing | 4 | 8/15/2022 | 8/18/2022 |

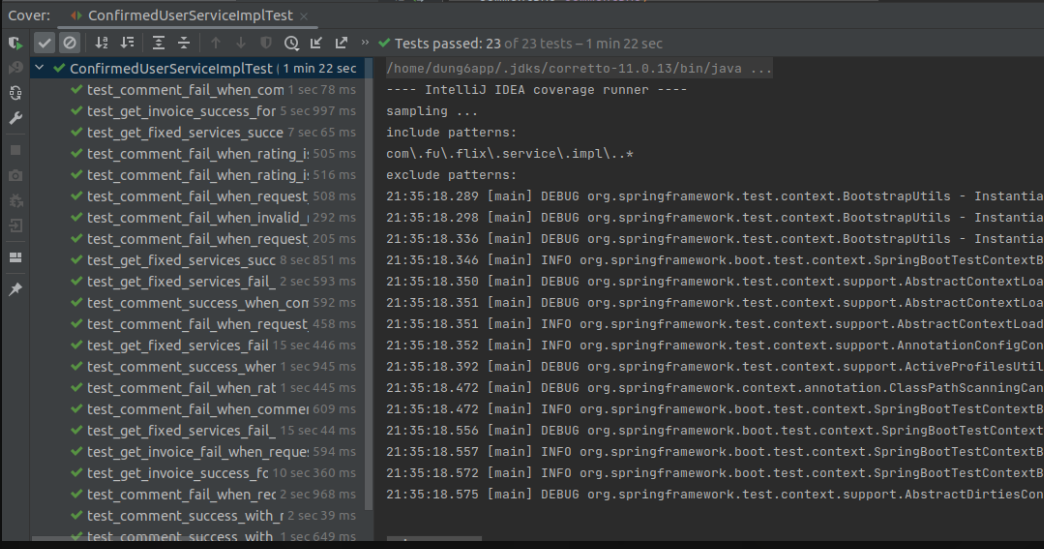
## 4. Test Cases

### 4.1 Unit testing

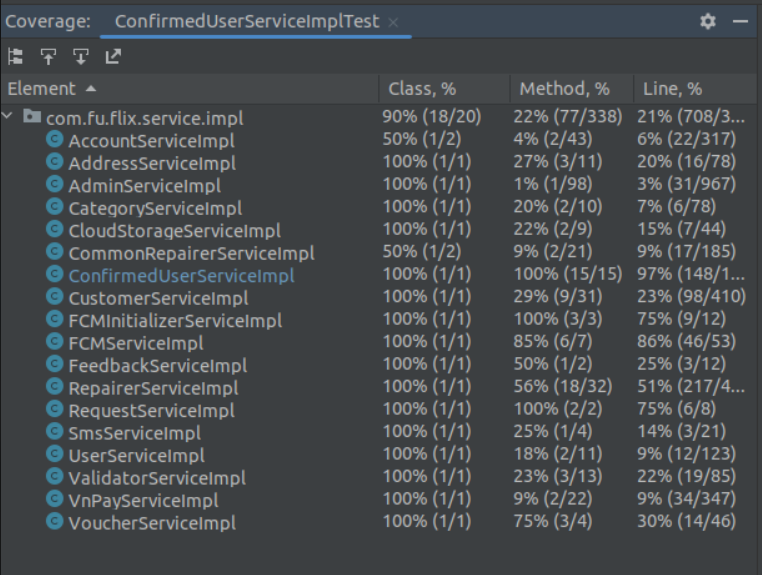
All unit tests are located in the **flix-be\src\test** on the source code. Here is an example of unit test cases:



*Figure 4.1.1 Unit Test example*

**

*Figure 4.1.2 Unit test result example*

**

*Figure 4.1.3 Unit test result example*

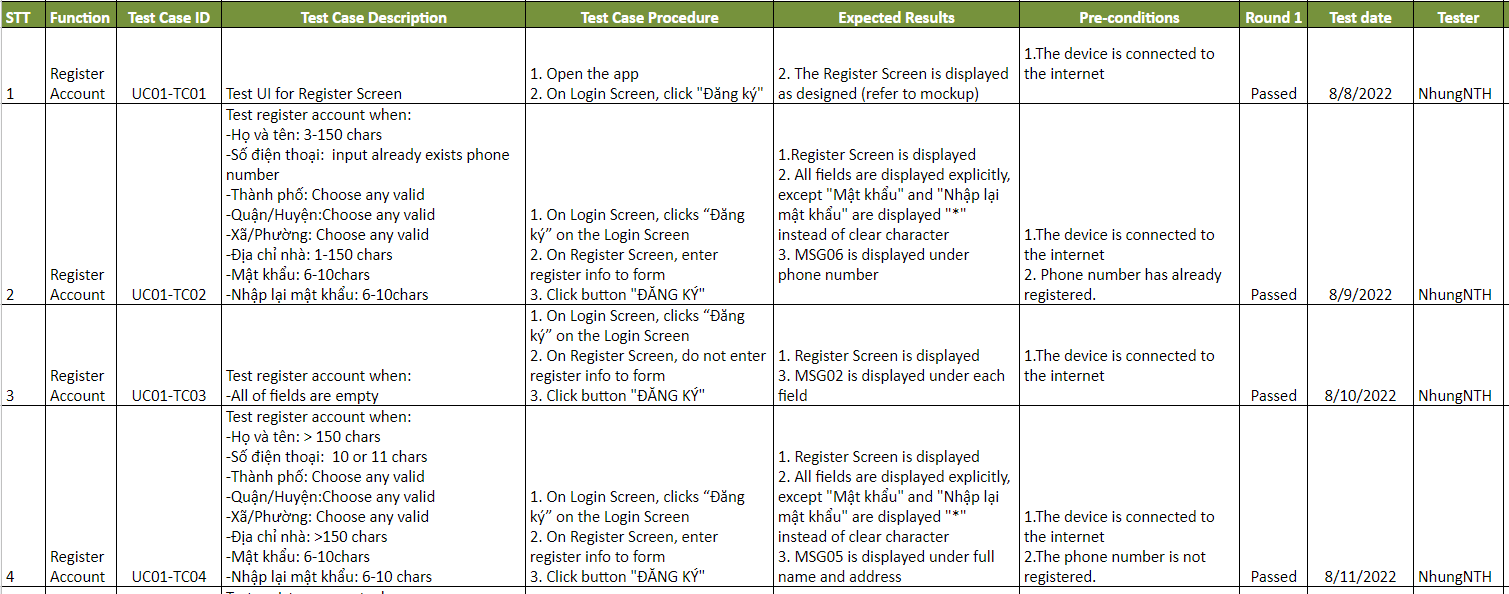
### 4.2 Integration Testing & System Testing

Detailed test cases of these types of testing are described in 2 files:

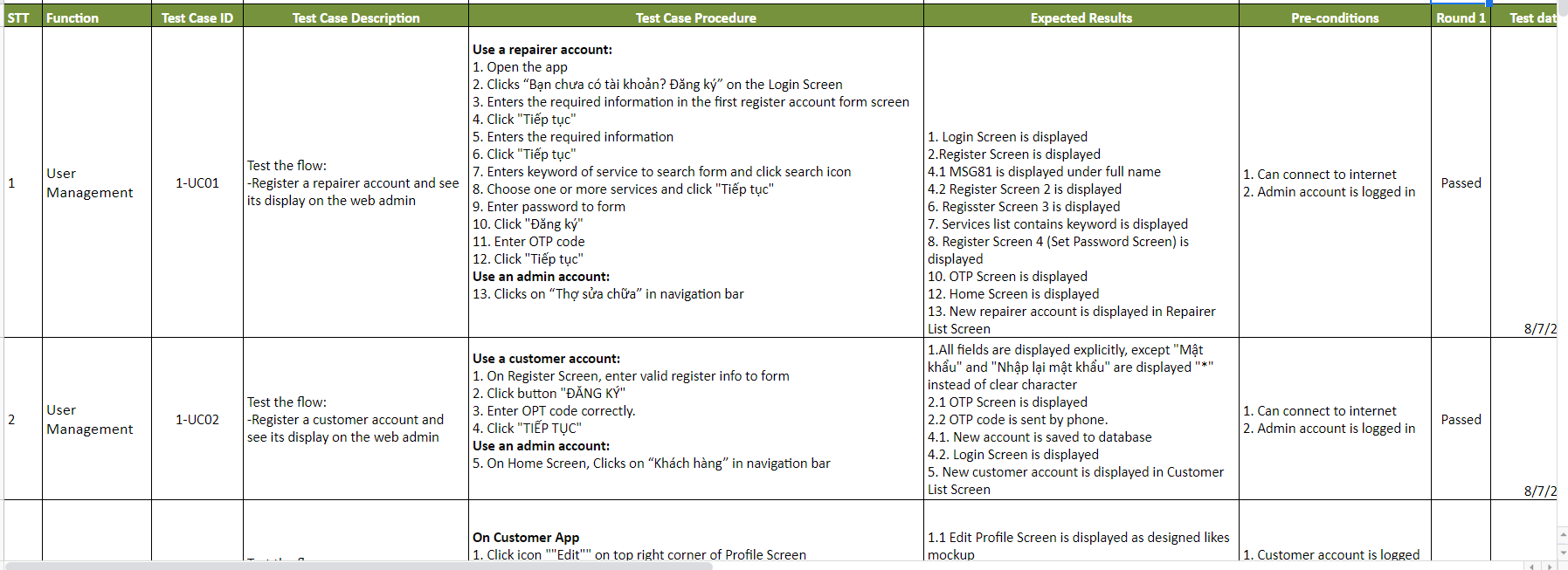
**Flix\_IntegrationTest.xlsx (for Integration Testing)**

**Flix\_SystemTest.xlsx (for System Testing)**

Below is an example of Integration and System test case.



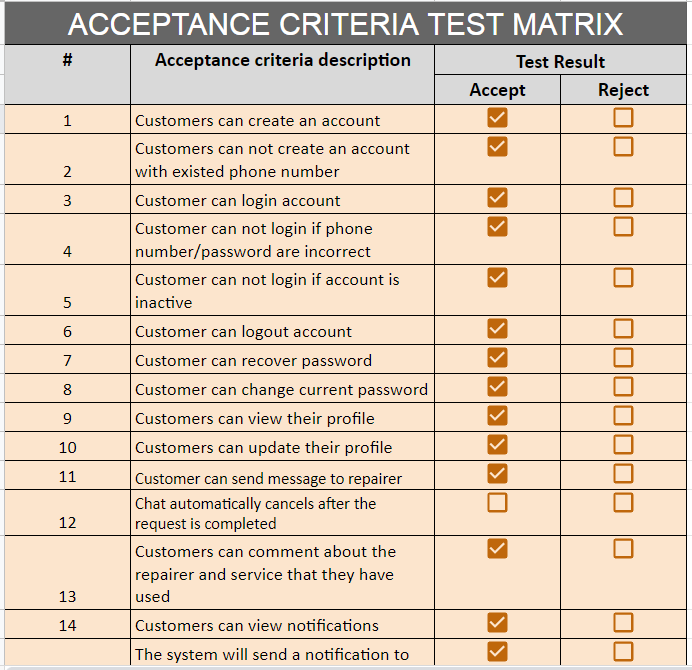
*Figure 4.2.1 Integration Test example*

**

*Figure 4.2.2 System Test example*

### 4.3 Acceptance Testing

Detailed checklists are described in **Flix\_AcceptanceTest\_Result.xlsx** file, below is a sample of the test checklist which is used for acceptance test.



*Figure 4.3.1 Acceptance Test example*

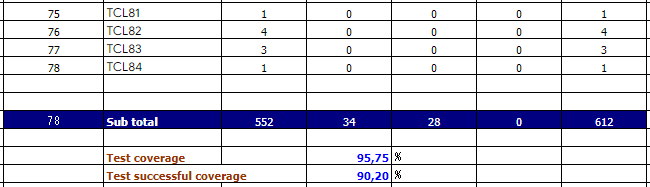
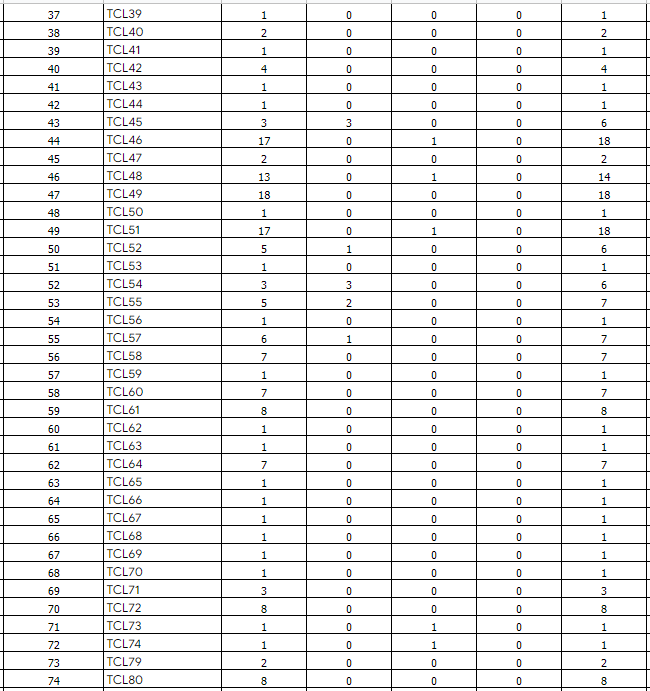
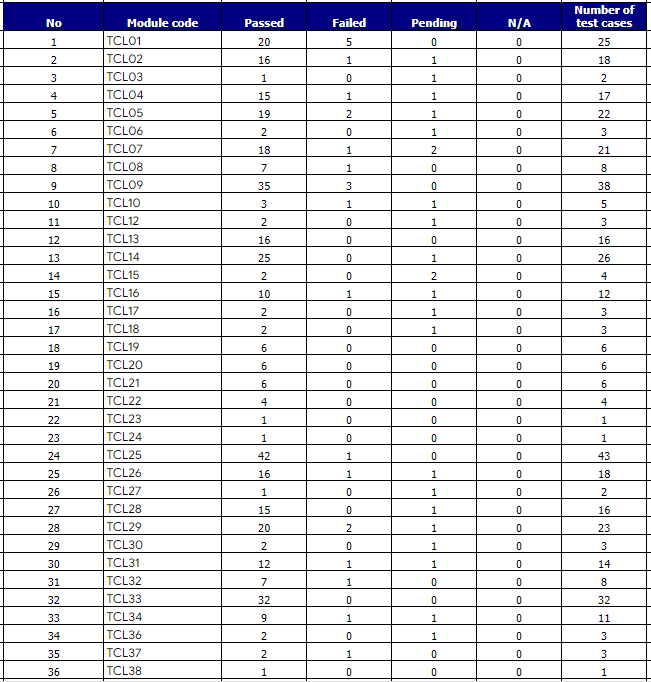
## 5. Test Reports

### 5.1 Unit Test

## 

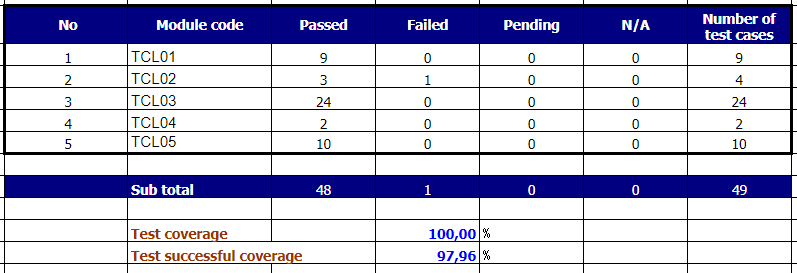
*Figure 5.1.1 Unit Test Report*

### 5.2 Integration Test

**

*Figure 5.2.1 Integration Test Report*

### 5.3 System Test

****

*Figure 5.3.1 System Test Report*