**<<ONLINE LEARNING SYSTEM>>**

**Software Testing Documentation**

## **Overall Description**

### **Test Model**

We choose testing by V-Model.

The V-model is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also aka the Verification **and Validation model**.

The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.

Advantages of V-Model:

* Simple and easy to use
* Testing activities like planning, test designing happens well before coding. This saves a lot of time
* Proactive defect tracking. That is defects are found at an early stage
* Avoids the downward flow of the defects
* Works well for small projects where requirements are easily understood

##### **Validation Phases**

The different Validation Phases in a V-Model are explained in detail below.

* *Unit Testing*

Unit tests designed in the module design phase are executed on the code during this validation phase. Unit testing is the testing at code level and helps eliminate bugs at an early stage, though all defects cannot be uncovered by unit testing.

* *Integration Testing*

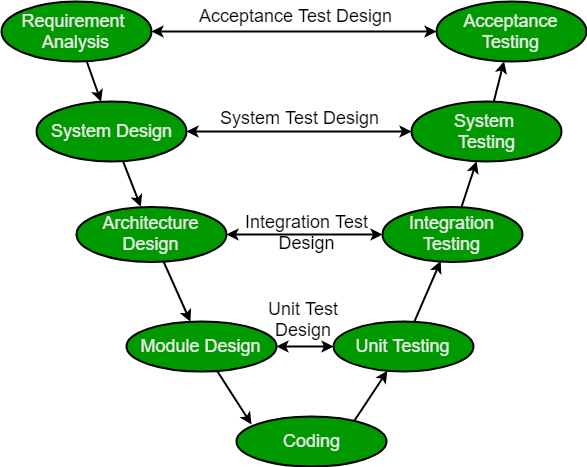
Integration testing is associated with the architectural design phase. Integration tests are performed to test the coexistence and communication of the internal modules within the system.

* *System Testing*

System testing is directly associated with the system design phase. System tests check the entire system functionality and the communication of the system under development with external systems. Most of the software and hardware compatibility issues can be uncovered during this system test execution.

* *Acceptance Testing*

Acceptance testing is associated with the business requirement analysis phase and involves testing the product in user environment. Acceptance tests uncover the compatibility issues with the other systems available in the user environment. It also discovers the non-functional issues such as load and performance defects in the actual user environment.

****

##### **Verification Phases**

There are several Verification phases in the V-Model, each of these are explained in detail

below.

* *User Requirement*

This is the first phase in the development cycle where the product requirements are understood from the customer’s perspective. This phase involves detailed communication with the customer to understand his expectations and exact requirement. This is a very important activity and needs to be managed well, as most of the customers are not sure about what exactly they need. The **acceptance test design planning** is done at this stage as business requirements can be used as an input for acceptance testing.

* *System Requirement Specification*

Once you have the clear and detailed product requirements, it is time to design the complete system. The system requirement specification will have the understanding and detailing the complete hardware and communication setup for the product under development. The system test plan is developed based on the system design. Doing this at an earlier stage leaves more time for the actual test execution later.

* *Architecture Design*

Architecture specifications are understood and designed in this phase. Usually more than one technical approach is proposed and based on the technical and financial feasibility the final decision is taken. The system design is broken down further into modules taking up different functionality. This is also referred to as **High Level Design (HLD)**.

The data transfer and communication between the internal modules and with the outside world (other systems) is clearly understood and defined in this stage. With this information, integration tests can be designed and documented during this stage.

* *Detailed Design*

In this phase, the detailed internal design for all the system modules is specified, referred to as **Low Level Design (LLD)**. It is important that the design is compatible with the other modules in the system architecture and the other external systems. The unit tests are an essential part of any development process and helps eliminate the maximum faults and errors at a very early stage. These unit tests can be designed at this stage based on the internal module designs.

##### **Coding Phase**

The actual coding of the system modules designed in the design phase is taken up in the Coding phase. The best suitable programming language is decided based on the system and architectural requirements.

The coding is performed based on the coding guidelines and standards. The code goes through numerous code reviews and is optimized for best performance before the final build is checked into the repository.

### **Testing Levels**

The project is using V-model, with daily iterations. At the end of the day the requirements identified for that iteration will be delivered to the team and will be tested.

Exploratory testing will play a large part of the testing as the team has never used this type of tool and will be learning as they go.

The Test Approach document is derived from the Project Plan, Requirements and Functional Specification documents. This document defines the overall test approach to be taken for the project. The Standard Test Approach document that you are currently reading is a boilerplate from which the more specific project Test Approach document can be extracted.

When this document is completed, the Test Lead will distribute it to the Product Manager, Development Lead, User Representative, Program Manager, and others as needed for review.

There are 4 levels: Unit testing, Integration testing, System testing and Acceptance testing.

| **No** | **Test Stages** | **Description** |
| --- | --- | --- |
| 1 | **Unit testing** | The main aim of this endeavor is to determine whether the |
| application functions as designed. In this phase, a unit can refer |
| to a function, individual program or even a procedure, and a |
| White-box Testing method is usually used to get the job done.t |
| 2 | **Integration testing** | Integration testing allows individuals the opportunity to |
| combine all of the units within a program and test them as a |
| group. This testing level is designed to find interface defects |
| between the modules/functions. This is particularly beneficial |
| because it determines how efficiently the units are running |
| together. |
| 3 | **System testing** | System testing is the first level in which the complete |
| application is tested as a whole. The goal at this level is to |
| evaluate whether the system has complied with all of the |
| outlined requirements and to see that it meets Quality |
| Standards. |
| 4 | **Acceptance testing** | Acceptance testing (or User Acceptance Testing), is conducted |
| to determine whether the system is ready for release. During |
| this phase, the tester and some alpha test user will test the |
| system to find out whether the application meets their business’ |
| needs. |

## **Test Plan**

### **Test Stages**

The following table show test stages in our project:

| **Type of test** |  | **Stage of Test** | | |
| --- | --- | --- | --- | --- |
| ***Unit*** | ***Integration*** | ***System*** | ***Acceptance*** |
| Unit test | X | X | X |  |
| GUI test |  |  | X |  |
| Functional test | X | X | X | X |
| Acceptance test |  |  |  | X |

##### **Human Resources**

The following table show human resources in our project:

| **In charge of** |  | **Role** | **Tasks** |
| --- | --- | --- | --- |
| Nguyen Tuan Duc |  | Project Manager | * Responsible for project schedule and overall |
|  |  | success of the project. |
|  |  | * Review test cases and reports. |
| Thai |  |  | * Preforming the actual system testing. |
|  |  | * Manage test resource and assign test tasks. |
| Viet Anh |
|  |  | * Create test plan. |
| Tuan Duc |  |  | * Create test cases. |
| Le Duc |  |  | * Create test report. |
|  |  | * Execute test. |
|  |  | * Test log report. |
| Nguyen Huy |
| Van Tuan |  |  | * Create unit test and integration test scripts. |
| Nguyen Huy |  |  | * Fix bugs. |

##### **Environment**

The following table show test environments in our project:

| **Purpose** |  | **Tools** | **Provider** | **Version** |
| --- | --- | --- | --- | --- |
| Software | JDK IDE | |  | 2022 |
| Google Chrome | |  | 111.0.5563.65 (Phiên bản Chính thức) (64 bit) |
| Microsoft Excel | |  | Office 365 |
| Microsoft Word | |  | Office 365 |
| Hardware | Hp Zbook G1 | |  | Windows 10 |
| Intel® Core™ i7 |
| Installed RAM 32.00GB |
| Acer Nitro 5 | |  | Windows 10 |
| Intel® Core™ i7 |
| Installed RAM 16.00GB |
| Asus Tuf A15 | |  | Windows 10 |
| Intel® Core™ i7 |
| Installed RAM 16.00GB |
| MSI Gf63 | |  | Windows 10 |
| Intel® Core™ i7 |
| Installed RAM 16.00GB |

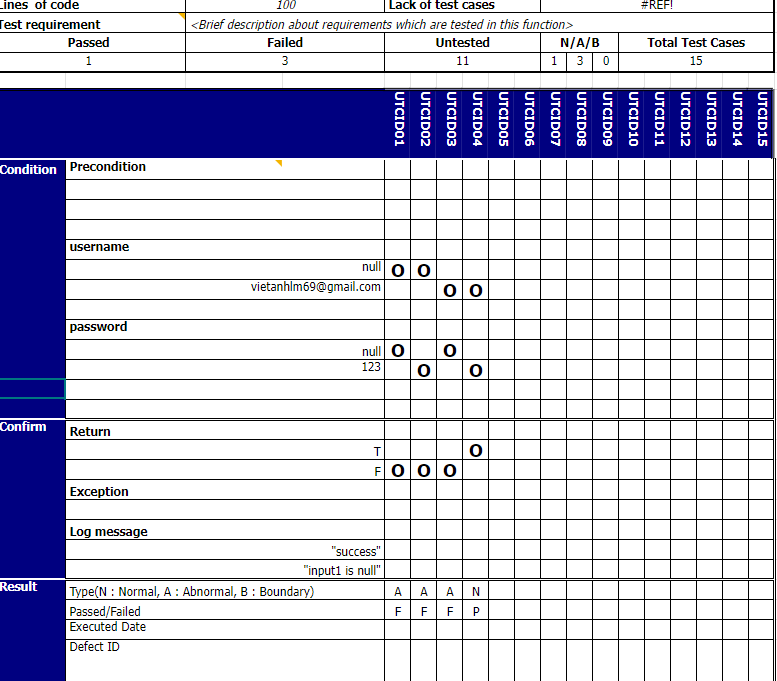
### **Test Cases**

### **Unit Test**

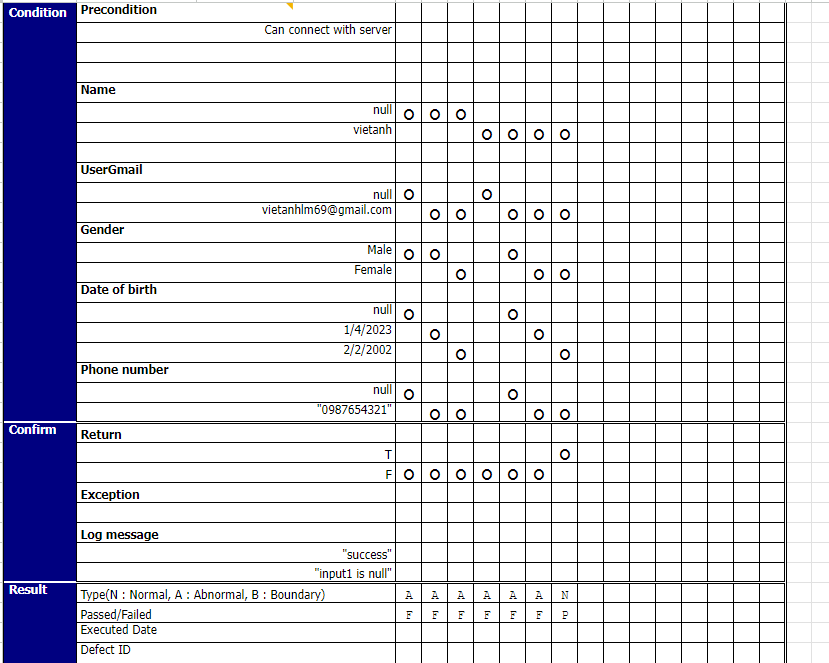
##### **Unit Test Case**

The following figure show some examples of unit test cases

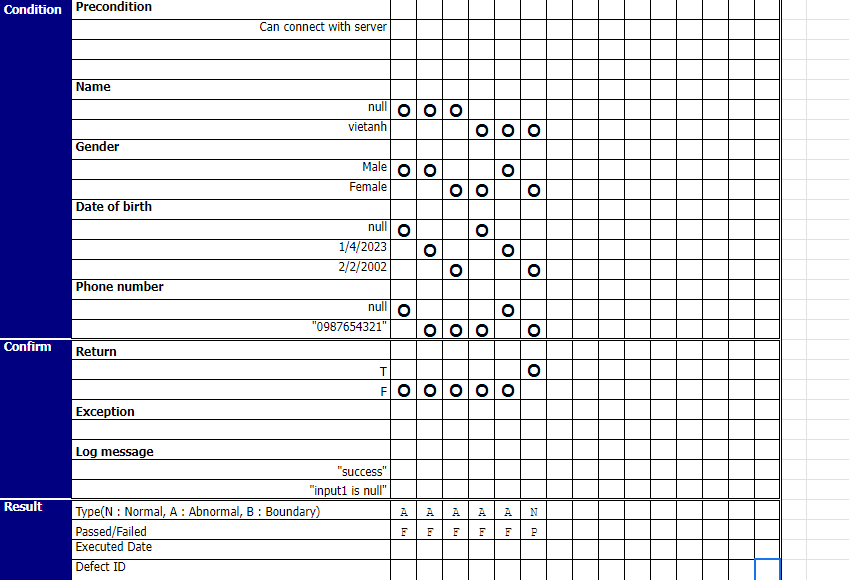
5.3.1.1.1 Login test



5.3.1.1.2 Register test

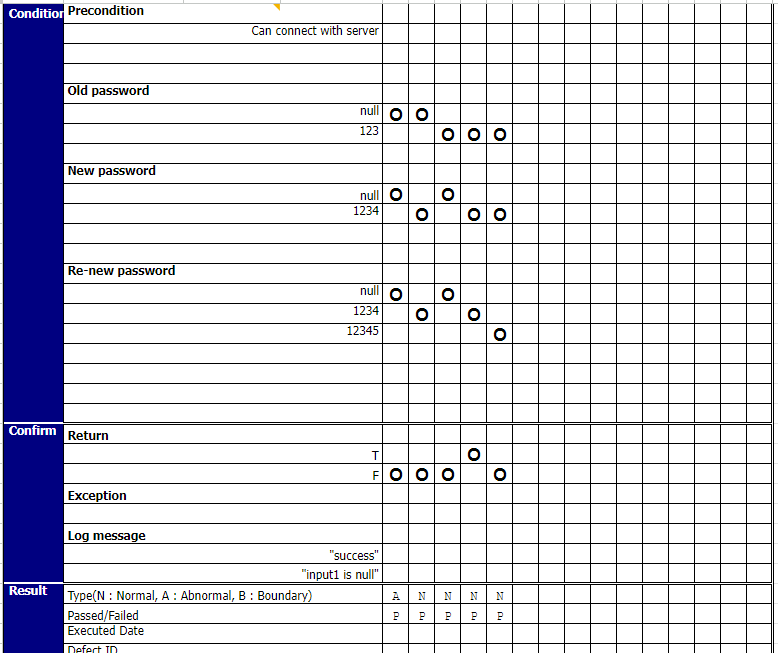


5.3.1.1.3 Change profile test

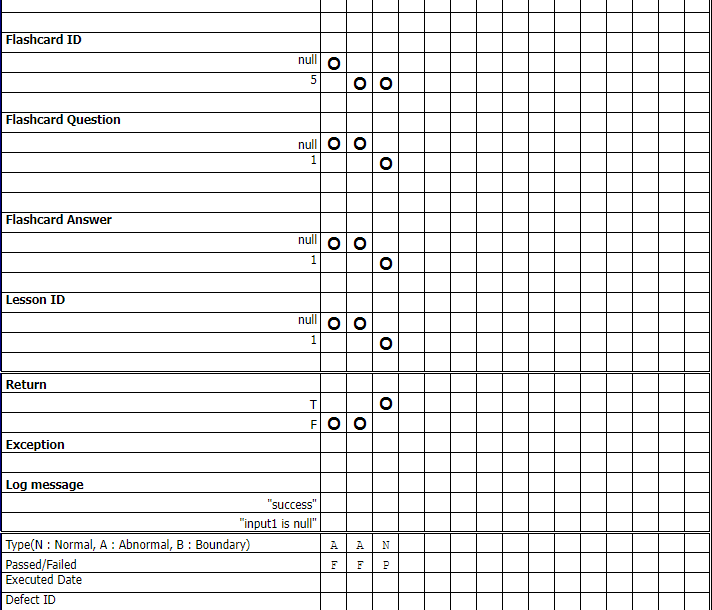


5.3.1.1.4 Change password test

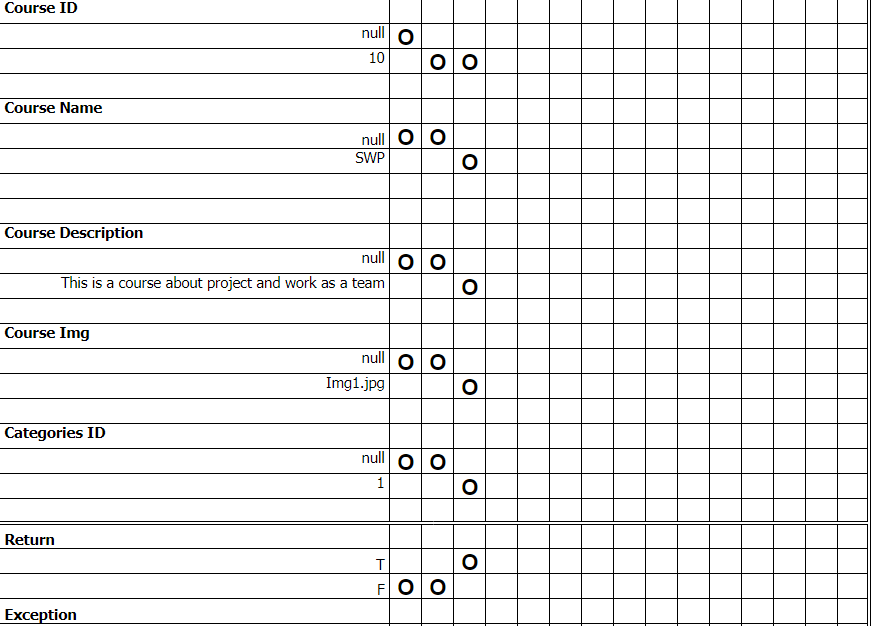
##### 



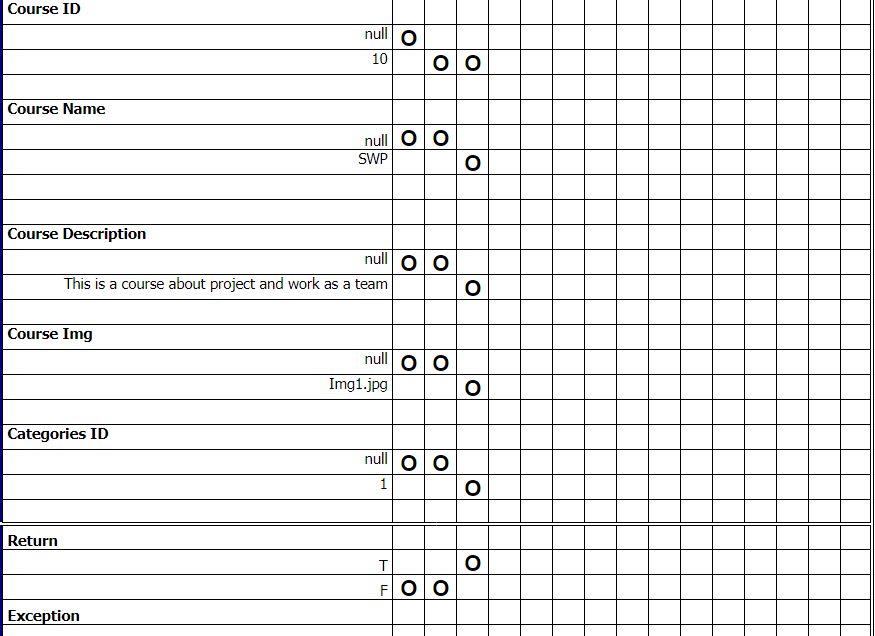
5.3.1.1.5 Add flashcard test

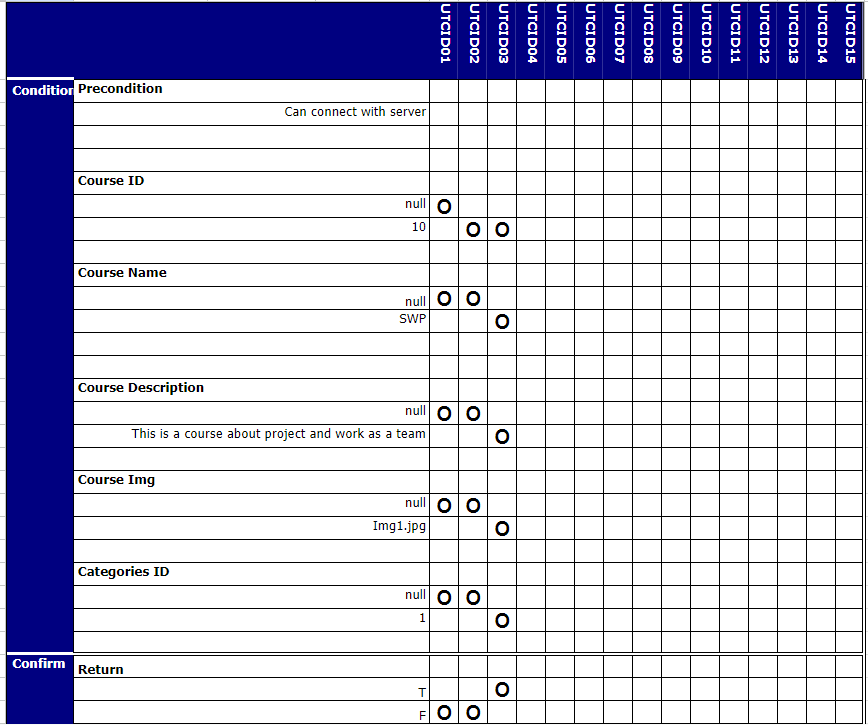


5.3.1.1.6 Add course test

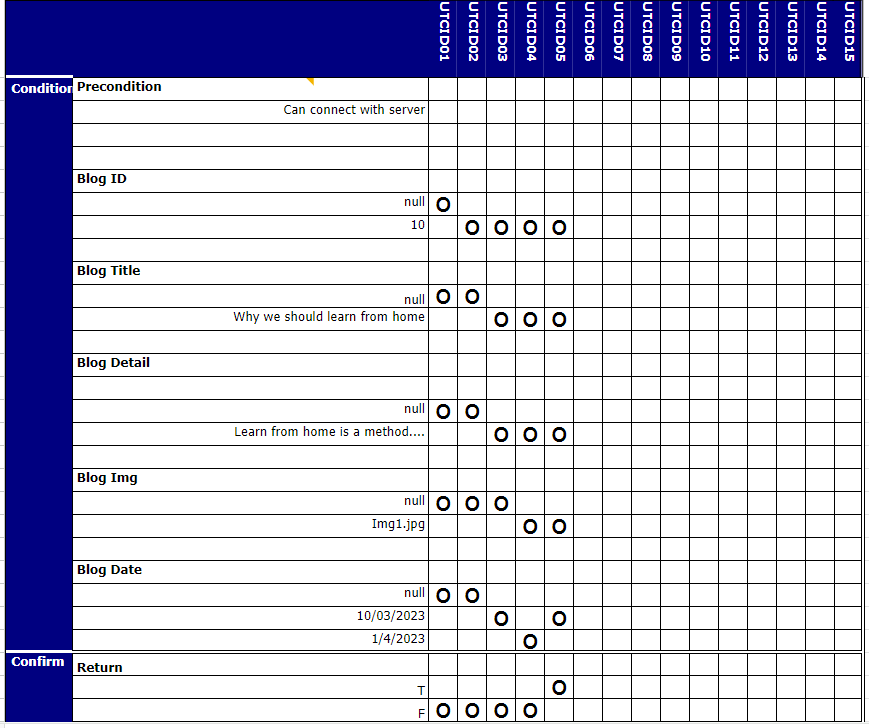


5.3.1.1.7 Update course test

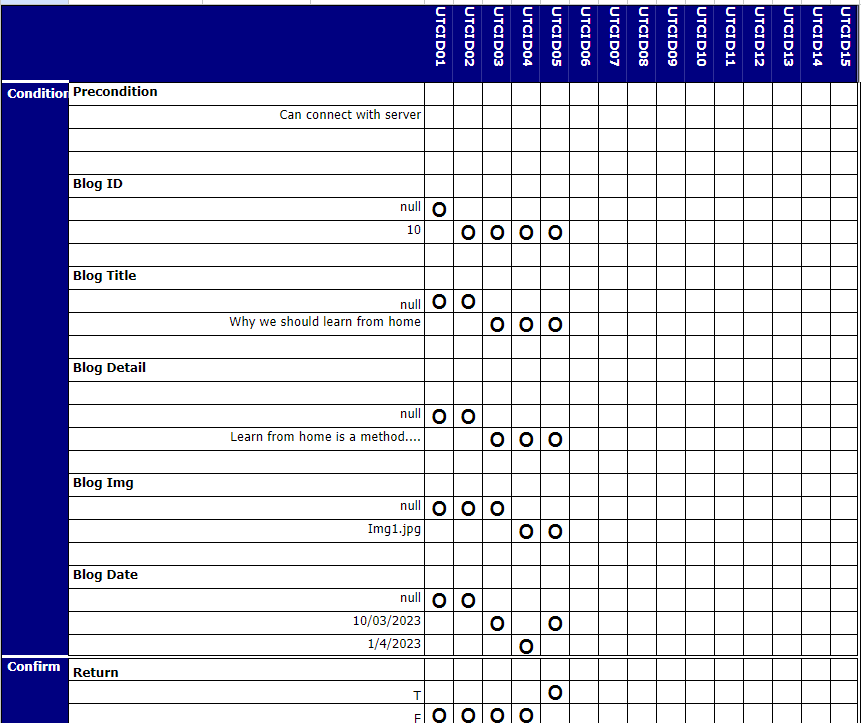




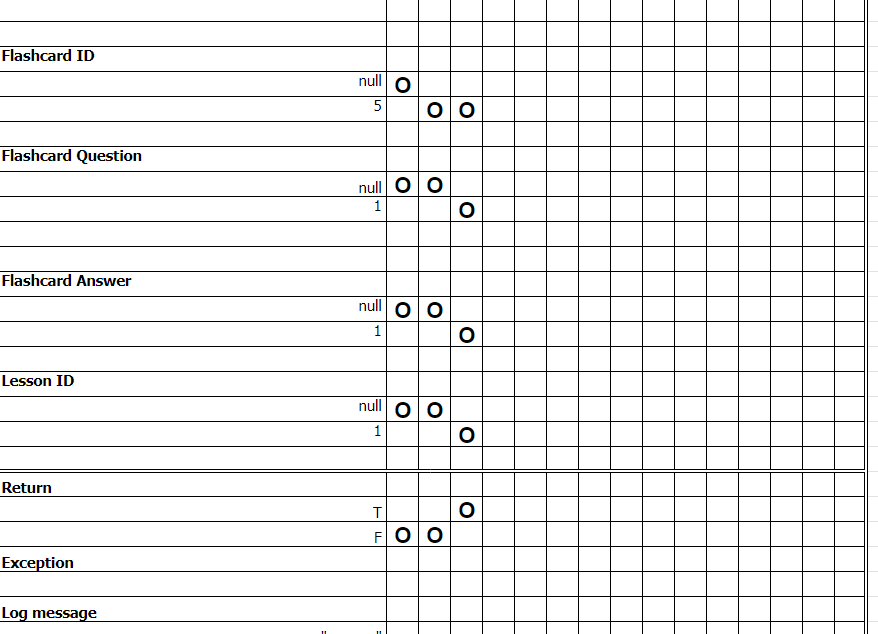
5.3.1.1.8 Add blog test



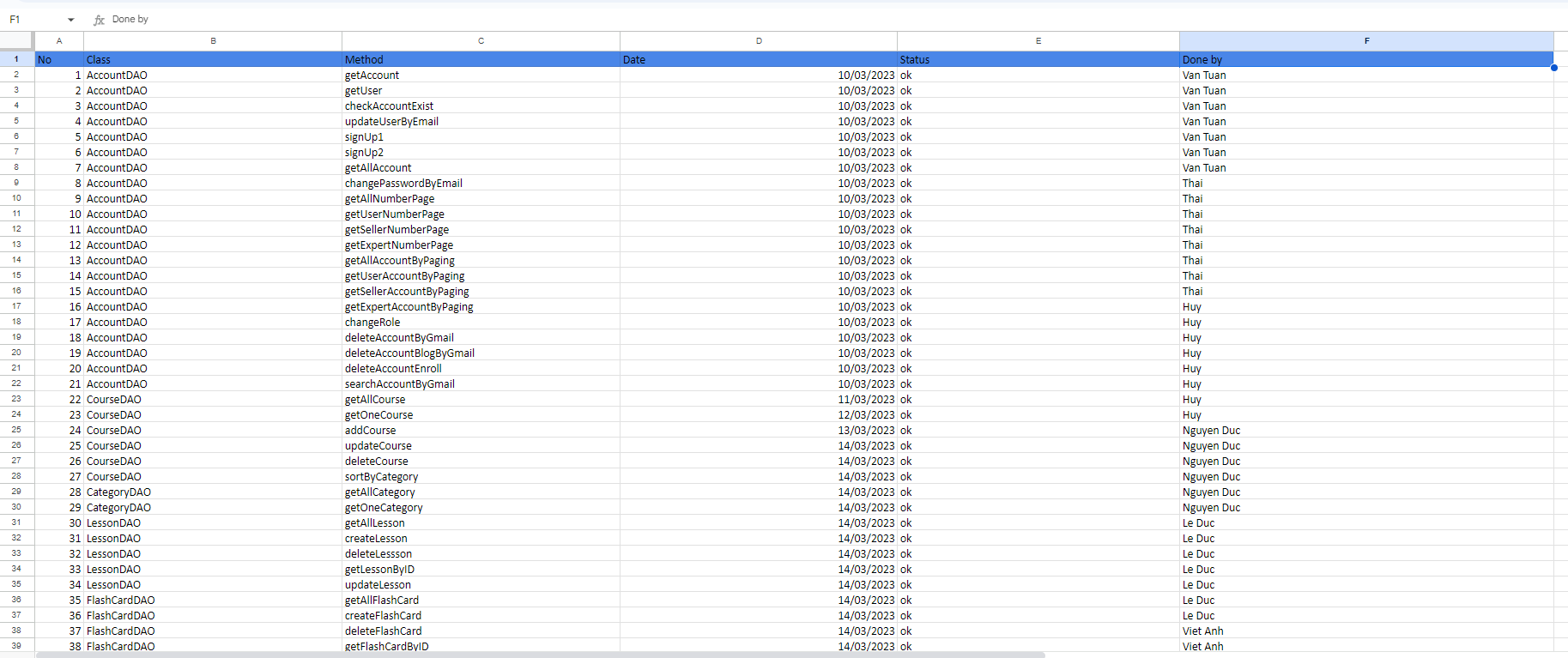
5.3.1.1.8 Update blog test



5.3.1.1.9 Update flashcard test



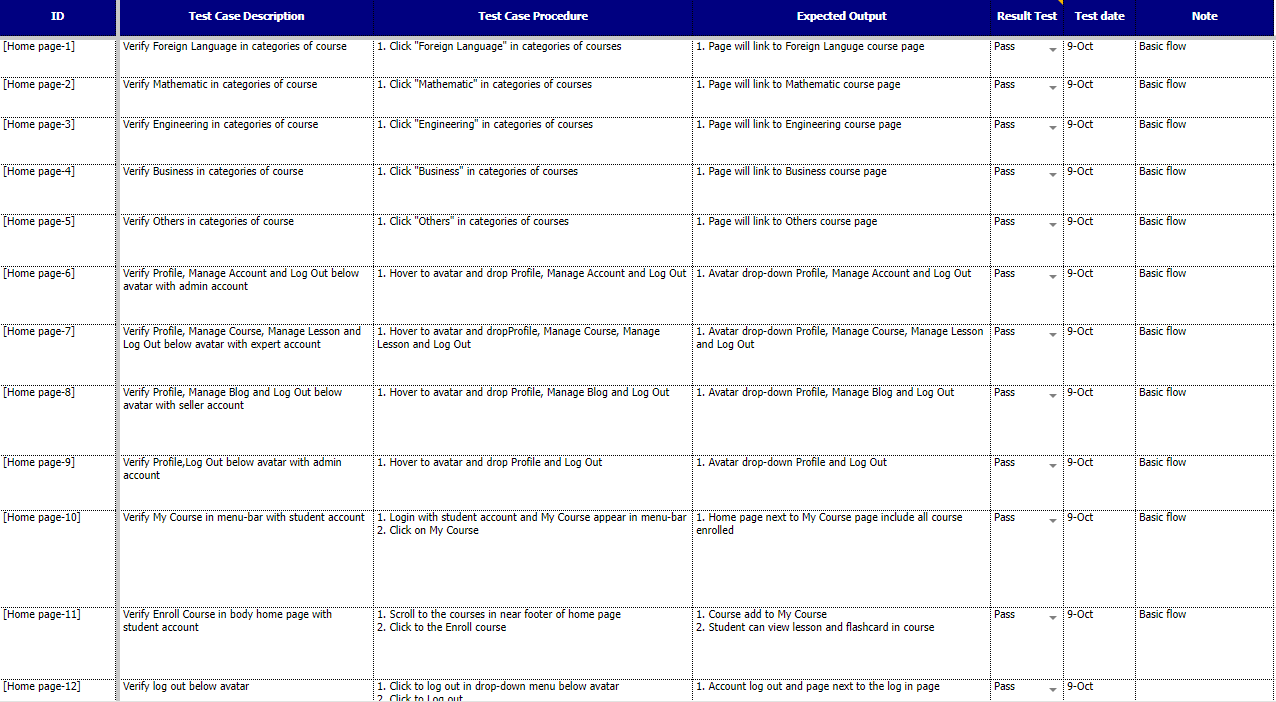
##### **5.2Unit Test Log**



##### **5.3.1 Integration test**

The term “integration” means a process of combining into an integral whole. Similarly, integration testing of a build indicates separate testing units of a system as one.

Detailed test cases will be described in the “[Integration Test.xlsx](https://docs.google.com/spreadsheets/d/1xTHS7yiB2f4QjKuv0QiqCs3QQBAFXRMO/edit#gid=1471854318)” file.

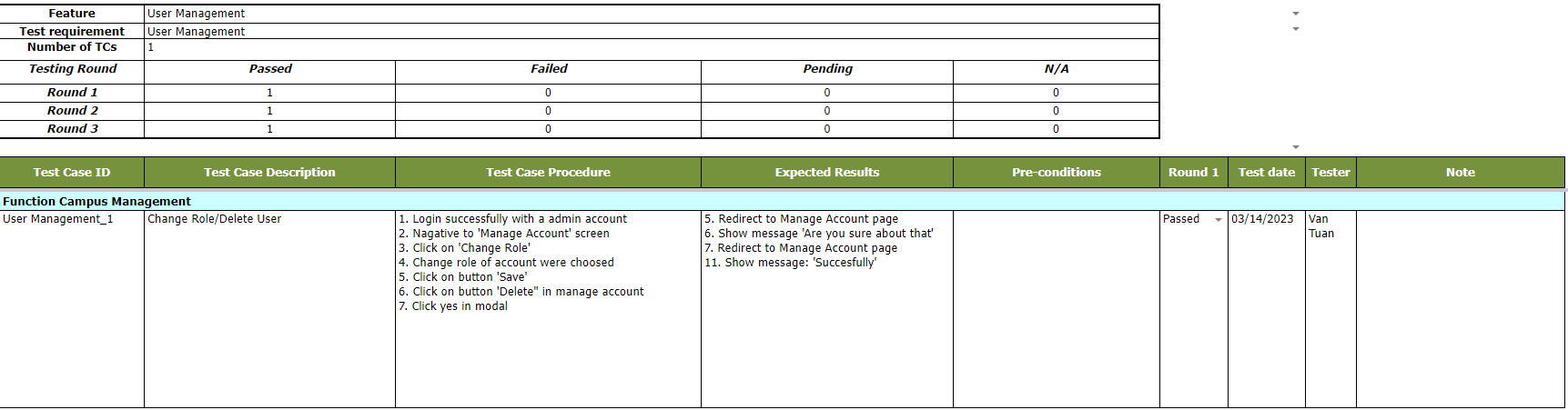


***Integration Test case sample***

##### **5.3.2 Acceptance test**

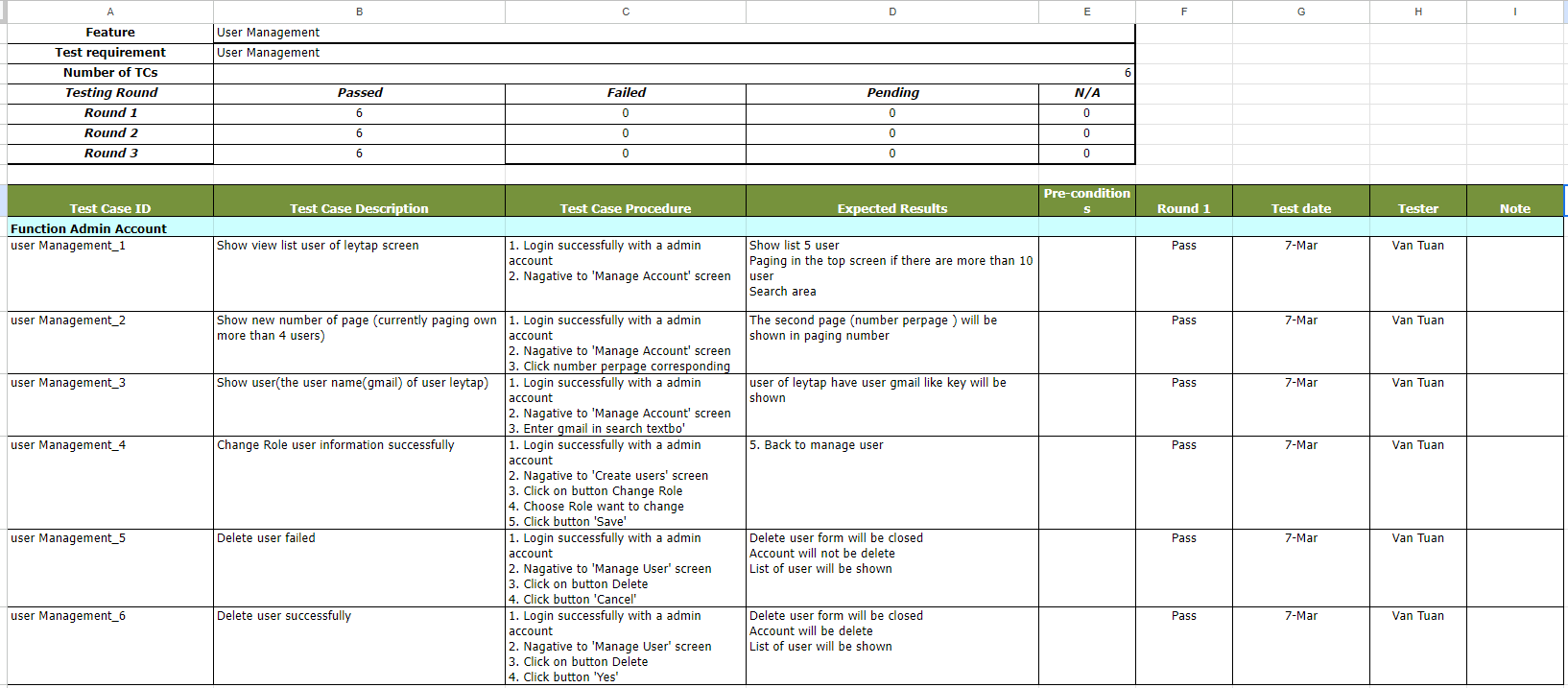
Acceptance Testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery. But our project will use the checklist as a substitute for acceptance testing.

Detailed test cases will be described in the “[Acceptance-Test.xlsx](https://docs.google.com/spreadsheets/d/1Z528J3OcmS4LEdjYtxw1VaHgeA4-qeUq/edit#gid=1339504847)” file.



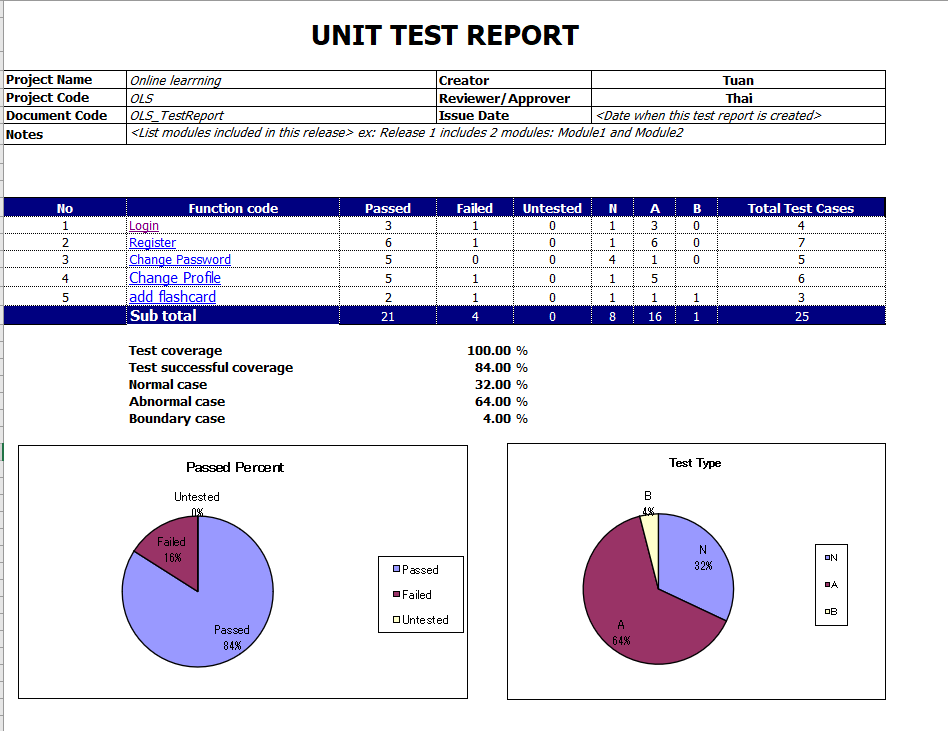
##### **5.3.3 System Test**

Detailed Test cases will be described in the “[System-Test.xlsx](https://docs.google.com/spreadsheets/d/10f8qbAdXNsLUXukH63fWmMoprSC150C0/edit#gid=2030525257)” file.

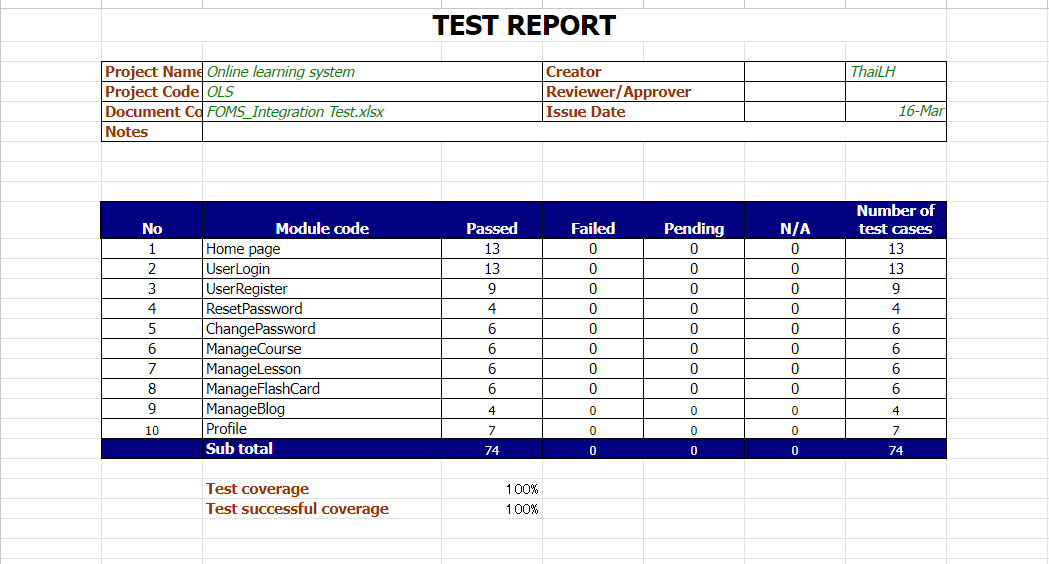


## **5.4 Test Report**

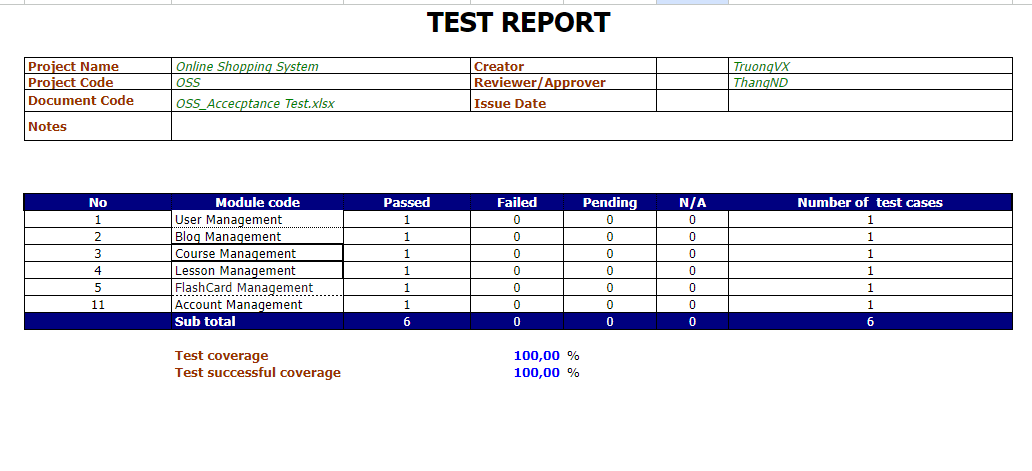
5.4.1.Unit test



5.4.2.Integration test



5.4.3.Acceptance test



5.4.4 System test

