|  |
| --- |
| **FSOFT-new-horizontal** |

<Name of the project>

TEST PLAN

Project Code: <Code of the project>

Document Code: <Code of the document >– v<x.x>

**<Location, issued date of the Document>**

Record of change

\*A - Added M - Modified D - Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Effective Date | Changed Items | A\* M, D | Change Description | New Version |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

SIGNATURE PAGE

**ORIGINATOR:** <Name> <Date>

< Role>

**REVIEWERS:** <Name> <Date>

< Role >

<Name, if it’s needed> <Date>

< Role >

**APPROVAL:** <Name> <Date>

< Role >

TABLE OF CONTENTS

[1 INTRODUCTION 6](#_Toc361155594)

[1.1 Purpose 6](#_Toc361155595)

[1.2 Definitions, Acronyms, and Abbreviations 6](#_Toc361155596)

[1.3 References 7](#_Toc361155597)

[1.4 Background information 7](#_Toc361155598)

[1.5 Scope of testing 7](#_Toc361155599)

[1.6 Constraints 8](#_Toc361155600)

[1.7 Risk list 8](#_Toc361155601)

[1.8 Training needs 8](#_Toc361155602)

[2 Requirements for Test 9](#_Toc361155603)

[2.1 Test items 9](#_Toc361155604)

[2.2 Acceptance Test Criteria 9](#_Toc361155605)

[3 TEST STRATEGY 10](#_Toc361155606)

[3.1 Test types 10](#_Toc361155607)

[3.1.1 Function Testing 10](#_Toc361155608)

[3.1.1.1 Function Testing 10](#_Toc361155609)

[3.1.1.2 Business Cycle Testing 11](#_Toc361155610)

[3.1.2 User Interface Testing 11](#_Toc361155611)

[3.1.3 Data and Database Integrity Testing 12](#_Toc361155612)

[3.1.4 Performance testing 13](#_Toc361155613)

[3.1.4.1 Performance testing 13](#_Toc361155614)

[3.1.4.2 Load Testing 14](#_Toc361155615)

[3.1.4.3 Stress Testing 15](#_Toc361155616)

[3.1.4.4 Volume Testing 16](#_Toc361155617)

[3.1.5 Security and Access Control Testing 17](#_Toc361155618)

[3.1.6 Regression Testing 18](#_Toc361155619)

[3.2 Test stages 18](#_Toc361155620)

[4 RESOURCE 20](#_Toc361155621)

[4.1 Human Resource 20](#_Toc361155622)

[4.2 Test management 20](#_Toc361155623)

[5 Test environment 21](#_Toc361155624)

[5.1 Hardware 21](#_Toc361155625)

[5.2 Software 21](#_Toc361155626)

[5.3 Infrastructure 21](#_Toc361155627)

[6 TEST MILESTONES 22](#_Toc361155628)

[7 DELIVERABLES 23](#_Toc361155629)

# 

# INTRODUCTION

## Purpose

*<Describe briefly about the purpose and organization of the documents. How many sections? What each section describes about?*

*Refer to Guideline Test Plan>*

## Definitions, Acronyms, and Abbreviations

*<This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the Test Plan>*

| Abbreviations | Description | Note |
| --- | --- | --- |
| AT | Acceptance test |  |
| B Voucher | Bug voucher |  |
| DMS | Defect Management System (Fsoft tool) |  |
| ES | Enhance Specification |  |
| IT | Integration test |  |
| PM | Project Manager |  |
| PTL | Project Technical Leader |  |
| PT/TT | Program test/ Total test |  |
| P Voucher | Program voucher |  |
| QA | Quality Assurance |  |
| QUP | Quality up |  |
| SRS | Software Requirement Specification |  |
| ST | System test |  |
| TP | Test Plan |  |
| TC | Test Case |  |
| TR | Test Report |  |
| UAT | User Acceptance test |  |
| UT | Unit test |  |

## References

*<This subsection provides a complete list of all documents referenced elsewhere in the Test Plan. Identify each document by title, Author, Version and Effective date>*

| Title/File name | Author | Version | Effective Date |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Background information

*<Enter a brief description of the target-of-test (components, application, system, etc.) and its goals. Include the information such as major functions and features, its architecture, and a brief history of the project.*

*Refer to Guideline Test Plan for sample>*

## Scope of testing

Dựa vào mô hình và thị trường của Shopee . Shopee là nền tảng thị trường trao đổi hàng hoá lớn vì vậy để đảm bảo ứng dụng nền tảng giảm thiểu tối đa những rủi ro do thực hiện dự , nhằm tăng hiệu suất và chất lượng của sản phẩm , yêu cầu đội ngũ test phải đưa ra những phạm vi kiểm thử.

Phạm vi kiểm thử:

+ Test các giao diện của shopee( ở đây nhóm test toàn lỗi sẽ kiểm tra cụ thể 8 màn hình)

+ Test các chức năng (Ví dụ : thêm vào giỏ hàng, test các đường link, Test đăng nhập đăng ký,…)

\_Ngoài phạm vi kiểm thừ

+ Cơ sở dữ liệu

+ Giao diện ngoàI

+ Phần cứng

## Constraints

Sau đây là những ràng buộc có trong kiểm thử. Nó đảm bảo cho quá trình kiểm thử đi đúng quỹ đạo và đụng thời gian qui đính

+ Tiến độ cột mốc thời gian cho dự án ( Tiến hành quan sát và đánh giá các màn hình theo đúng thời hạn)

+ Chất lượng sản phẩm và chất lượng kiểm (Đưa ra đánh giá về sản phẩm và đánh giá về chất lượng của sản phẩm)

+ Nguồn lực tham gia vào kiểm thử dự án (Nhận và chia số lượng công việc phù hợp cho mỗi thành viên trong nhóm

+Phạm vi kiểm thử rộng, đa dạng (Thực hiện kiểm tra tất cả các tiêu chí trong phạm vi kiểm thử đưa ra)

+ Đảm bảo hạn chế rủi ro xảy ra ( Ngăn chặn và đưa ra một số biện pháp cho rủi ro nếu nó xảy ra)

+ Thực hiện đầy đủ các requirement của khách hàng ( Đánh giá và xem xét thưc hiện theo các yêu cầu mà khách hàng đưa ra)

## Risk list

Trong quá trình thực hiện kiểm tra sản phẩm. Một số rủi ro có thể xảy ra nhằm hạn chế quá trình kiểm tra của nhóm kiểm:

Các rủi ro gặp phải là do một số nguyên nhân như:

|  |  |
| --- | --- |
| **Risk** | **Giải pháp giảm tránh Risk** |
| Thành viên trong nhóm thiếu các kỹ năng cần thiết để kiểm thử trang. | Lập kế hoạch khóa training để nâng cao kỹ năng của các thành viên |
| Mô hình dự án lớn và nhiều | Đặt mức độ ưu tiên (Test Priority) cho từng màn hình hoạt động kiểm thử. |
| Thời gian kiểm tra bị giới hạn và hạn chế | Lập kế hoạch và phân chia thời gian hợp lý hoá |
| Thiếu hợp tác ảnh, ảnh hưởng tiêu cực đến năng suất của thành viên | Khuyến khích mỗi thành viên trong nhóm thực hiện nhiệm vụ của mình và truyền cảm hứng cho họ để họ nỗ lực nhiều hơn. |
|  |  |

## Training needs

Trong việc thực hiện kiểm thử mọt sản phẩm yêu cầu các thành viên phải có kiến thức về test và việc sử dụng các công cụ hỗ trợ kiểm thử. Để đảm bảo các thành viên đều có kiến thức kiểm yêu cầu phải có sự đào tạo cho các thành viên. Sau đây là danh sách những yêu cầu đào tạo:   
+ Đào tạo qui trình thiết kế và lập kế hoạch test (Đưa ra các test plan và các test case)

+ Đào tạo qui trình kiểm thử (Các bước kiểm thử vẫn chưa thực hiện)

+ Đào tạo cách sử dụng các công cụ hỗ trợ test( Hiện chưa thể thực hiện)

+ Đào tạo quá trình phát hiện lỗi và thực hiện xử lí

# Requirements for Test

## Test items

*<The listing below identifies those items (use cases, functional requirements, non-functional requirements) that have been identified as targets for testing. This list represents what will be tested and the number of test case estimated for testing each item.*

*Enter a high level list of features and functions to be tested/not tested. Refer to the sample in Guideline Test Plan>.*

*The table below can be used if suitable.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Name of features and functions | Outline of features and functions | Number of Test case(Estimate) | Note |
|  |  |  |  |  |
|  |  |  |  |  |

## Acceptance Test Criteria

*<List of criteria to define what levels of test quality are sufficient to move to the next testing phase, such as:*

* *Test coverage*
* *Successful Test coverage*
* *Number of Test cases (Unit/Integration/System Test cases)*
* *Number of defects/Weighted defects*

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

* *Number of UTC/KLOC: 80 UTC/KLOC*
* *Number or Weighted defects/KLOC*
* *Statement coverage: 90%*
* *Branch coverage: 100%*
* *Path coverage: 100%*
* *….*

*Refer to the sample in Guideline Test Plan>*

# TEST STRATEGY

*<The Test Strategy presents the recommended approach to the testing of the target-of-test. Outline for test strategy, refer to Guideline Test Plan. >*

## Test types

### Function Testing

#### 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. For more details, refer to the sample in Guideline Test Plan. >*

|  |  |
| --- | --- |
| 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> |

#### Business Cycle Testing

*<Outline for Business Cycle Testing. Refer to the sample in Guideline Test Plan>*

|  |  |
| --- | --- |
| Test Objective | <Ensure proper target-of-test and background processes function according to required business models and schedules. > |
| Technique: | <Testing will simulate several business cycles by performing the following:  The tests used for target-of-test’s function testing will be modified or enhanced to increase the number of times each function is executed to simulate several different users over a specified period.  All time or date-sensitive functions will be executed using valid and invalid dates or time periods.    All functions that occur on a periodic schedule will be executed or launched at the appropriate time.  Testing will include 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: | <System dates and events may require special support activities  Business model is required to identify appropriate test requirements and procedures. > |

### User Interface Testing

*< User Interface (UI) testing verifies a user’s interaction with the software. For more details, refer to the sample in Guideline Test Plan >*

|  |  |
| --- | --- |
| 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. > |

### Data and Database Integrity Testing

*< The databases and the database processes should be tested as a subsystem within the Project. For more details, refer to the sample in Guideline Test Plan >*

|  |  |
| --- | --- |
| 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. > |

### Performance testing

#### Performance testing

*<Performance test is to measure and evaluate response times, transaction rates, and other time-sensitive requirements. It includes Load test, Stress test, Volume test...*

*For more details, refer to the sample in Guideline Test Plan >*

|  |  |
| --- | --- |
| 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. > |

#### Load Testing

*<Load testing is a performance test which subjects the target-of-test to varying workloads to measure and evaluate the performance behaviors and ability of the target-of-test to continue to function properly under these different workloads. For more details, refer to the sample in Guideline Test Plan >*

|  |  |
| --- | --- |
| Test Objective: | <Verify performance behavior time for designated transactions or business cases under varying workload conditions. > |
| Technique: | <Use tests developed for Function or Business Cycle Testing.  Modify data files to increase the number of transactions or the tests to increase the number of times each transaction occurs. > |
| Completion Criteria: | <Multiple transactions or multiple users:  Successful completion of the tests without any failures and within acceptable time allocation. > |
| Special Considerations: | <Load testing should be performed on a dedicated machine or at a dedicated time.  This permits full control and accurate measurement. The databases used for load testing should be either actual size or scaled equally. > |

#### Stress Testing

*<Stress testing is a type of performance test implemented and executed to find errors due to low resources or competition for resources. For more details, refer to the sample in Guideline Test Plan >*

*<Note:  References to transactions below refer to logical business transactions. >*

|  |  |
| --- | --- |
| Test Objective: | <Verify that the target-of-test functions properly and without error under the following stress conditions:  little or no memory available on the server (RAM and DASD)  maximum actual or physically capable number of clients connected or simulated  multiple users performing the same transactions against the same data or accounts  worst-case transaction volume or mix (see Performance Testing above).  Notes: The goal of Stress Testing might also be stated as identify and document the conditions under which the system FAILS to continue functioning properly. > |
| Technique: | <Use tests developed for Performance Profiling or Load Testing.  To test limited resources, tests should be run on a single machine, and RAM and DASD on server should be reduced or limited.  For remaining stress tests, multiple clients should be used, either running the same tests or complementary tests to produce the worst-case transaction volume or mix. |
| Completion Criteria: | <All planned tests are executed and specified system limits are reached or exceeded without the software failing or conditions under which system failure occurs is outside of the specified conditions. > |
| Special Considerations: | <Stressing the network may require network tools to load the network with messages or packets.  The DASD used for the system should temporarily be reduced to restrict the available space for the database to grow.  Synchronization of the simultaneous clients accessing of the same records or data accounts. > |

#### Volume Testing

*< Volume Testing subjects the target-of-test to large amounts of data to determine if limits are reached that cause the software to fail. For more details, refer to the sample in Guideline Test Plan >*

|  |  |
| --- | --- |
| Test Objective: | <Verify that the target-of-test successfully functions under the following high volume scenarios:  Maximum (actual or physically- capable) number of clients connected, or simulated, all performing the same, worst case (performance) business function for an extended period.  Maximum database size has been reached (actual or scaled) and multiple queries or report transactions are executed simultaneously. > |
| Technique: | <Use tests developed for Performance Profiling or Load Testing.  Multiple clients should be used, either running the same tests or complementary tests to produce the worst case transaction volume or mix (see Stress Testing above) for an extended period.  Maximum database size is created (actual, scaled, or filled with representative data) and multiple clients used to run queries and report transactions simultaneously for extended periods. > |
| Completion Criteria: | <All planned tests have been executed and specified system limits are reached or exceeded without the software or software failing. > |
| Special Considerations: | <What period of time would be considered an acceptable time for high volume conditions, as noted above? > |

### Security and Access Control Testing

*< Outline for Security and Access Control Testing, refer to the sample in Guideline Test Plan >*

|  |  |
| --- | --- |
| Test Objective: | Application-level Security:  <Verify that an actor can access only those functions or data for which their user type is provided permissions. >  System-level Security:  Verify that only those actors with access to the system and applications are permitted to access them. > |
| Technique: | Application-level Security:  <Identify and list each user type and the functions or data each type has permissions for. >  <Create tests for each user type and verify each permission by creating transactions specific to each user type. >  Modify user type and re-run tests for same users.  In each case, verify those additional functions or data are correctly available or denied.  System-level Access: <See Special Considerations below> |
| Completion Criteria: | <For each known actor type the appropriate function or data are available, and all transactions function as expected and run in prior Application Function tests. > |
| Special Considerations: | <Access to the system must be reviewed or discussed with the appropriate network or systems administrator.  This testing may not be required as it may be a function of network or systems administration. > |

### Regression Testing

*<Regression testing is a necessary maintenance activity aimed at showing that code has not been adversely affected by changes. For more details, refer to the sample in Guideline Test Plan >*

|  |  |
| --- | --- |
| 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>.  <Use Rational Robot tool: Creating some functional test scripts. Define automated test execution schedule here **>**  < 80% Test cases is randomly select from existing Test cases >  < 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: |  |

## 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 | Integration | System | Acceptance |
| <Function Test > | X | X | X | X |
| <User Interface test> | X |  | X |  |
| <Performance Tests  (Performance profiles of individual components)> | X | X |  |  |
| <Load, Stress, Volume test> |  |  | X | X |
| <Security test> | X |  | X |  |
| <Date integrity test> |  | X | X |  |

# 

# RESOURCE

## Human Resource

*This table shows the staffing assumptions for the project.*

|  |  |  |  |
| --- | --- | --- | --- |
| Worker/Doer | Role | Specific Responsibilities/Comments | Location |
| <Test Leader ‘s Name>  <Mail account> | <Test Leader> | <Manage Test resource and assign test tasks>  <Create TP>  <Review TC>  <Create Test report> | < FPT, Vietnam> |
| <Tester 1 ‘s Name>  <Mail account> | <Tester > | <Create TC for modules …>  <Execute test>  <Report test result> | < FPT, Vietnam> |
| <Tester 2’s Name>  <Mail account> | <Tester > | <Create TC for modules … >  <Execute test>  <Report test result> | < FPT, Vietnam> |

## Test management

*<Define about the following items’ method:*

Test management (Test planning and tracking, Communication)

Defect management (Defect management tool and defect process follow……)

*Refer to the sample in Guideline Test Plan>*

# Test environment

*<List of required hardware and software resources. Describe equipments which needs for testing such as USB Memory, Memory card…etc. Describe the number of equipments need and the duration using these equipments. >*

## Hardware

*<Describe the hardware which projects will use for testing*

*Refer to the sample in Guideline Test Plan>*

## Software

*<Describe the software which projects will use for testing*

*Refer to the sample in Guideline Test Plan>*

## Infrastructure

*<List tools will be employed for this project.*

*Refer to the sample in Guideline Test Plan>*

|  |  |  |  |
| --- | --- | --- | --- |
| Purpose | Tool | Vendor/In-house | Version |
| <Defect log> | <DMS2> | <Fsoft tool> | <1.4> |
| <Test effort> | <Timesheet> | <Fsoft tool> | <3.8.4> |

# 

# TEST MILESTONES

*<Separate test milestones, which should be identified to communicate project status accomplishments>*

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone Task | Effort (pd) | Start Date | End Date |
| <Create Unit Test Plan> |  | <YY-MM-DD> | <YY-MM-DD> |
| <Review & update UTP> |  |  |  |
| <Create Unit Test case> |  |  |  |
| <Review & update UTC> |  |  |  |
| <Create Test Plan> |  |  |  |
| <Review & update TP> |  |  |  |
| <Create Integration Test case> |  |  |  |
| Review & Update Integration TC> |  |  |  |
| <Create System Test case> |  |  |  |
| <Review & Update System TC> |  |  |  |
| <Execute Unit Test> |  |  |  |
| <Execute Integration test> |  |  |  |
| <Execute System test> |  |  |  |

# 

# DELIVERABLES

*<Define delivery of Testing. Refer to Project Plan for more details.>.*

| No | Deliverables | Language | Delivered Date |
| --- | --- | --- | --- |
|  | <Unit Test Plan> | <Japanese> | <YY-MM-DD>\* |
|  | <Test Plan> |  |  |
|  | <Unit Test cases> |  |  |
|  | <Integration Test Cases> |  |  |
|  | <System Test cases> |  |  |
|  | <Defect log> |  |  |
|  | <Test reports> |  |  |

*Note: \* <YY-MM-DD> is only used when customer is Japanese.*

*Content guideline in <> must be deleted when finish test plan.*