

**Capstone Project Report**

**Report 5 – Software Test Documentation**

– Hanoi, August 2019 –

**Table of Contents**

[I Project Report 3](#_Toc77682613)

[1 Status Report 3](#_Toc77682614)

[2 Team Involvements 3](#_Toc77682615)

[3 Issues/Suggestions 3](#_Toc77682616)

[II. Test Plan 4](#_Toc77682617)

[1 Introduction 4](#_Toc77682618)

[1.1 Purpose 4](#_Toc77682619)

[1.2 Definitions, Acronyms, and Abbreviations 4](#_Toc77682620)

[1.3 References 4](#_Toc77682621)

[2 Scope of Testing 5](#_Toc77682622)

[2.1 Target of Testing 5](#_Toc77682623)

[2.2 Test Stages 5](#_Toc77682624)

[2.3 Test Assumptions 6](#_Toc77682625)

[3 Requirements for Test 6](#_Toc77682626)

[3.1 Test Items 6](#_Toc77682627)

[3.2 Acceptance Criteria 6](#_Toc77682628)

[4 Test Strategy 6](#_Toc77682629)

[4.1 Test Model 6](#_Toc77682630)

[4.2 Type of Testing 7](#_Toc77682631)

[4.3 Test stage 12](#_Toc77682632)

[4.4 Tools 13](#_Toc77682633)

[5 Resources 13](#_Toc77682634)

[5.1 Human Resources 13](#_Toc77682635)

[5.2 Software and Hardware 13](#_Toc77682636)

[6 Test Schedule 13](#_Toc77682637)

[6.1 Test Milestones 13](#_Toc77682638)

[6.2 Deliverables 13](#_Toc77682639)

[II Test Cases 13](#_Toc77682640)

[III Test Reports 14](#_Toc77682641)

[IV Defect Reports 14](#_Toc77682642)

# I Project Report

## 1 Status Report

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Work Item** | **Status** | **Notes (Work Item in Details)** |
| 1 |  | Pending |  |
| 2 |  | In Progress |  |
| 3 |  | Completed |  |

## 2 Team Involvements

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task** | **Member** | **Notes (Task Details, etc.)** |
| 1 |  | KienNT |  |
| 2 |  | TuanTV |  |
| 3 |  | AnhLM |  |

## 3 Issues/Suggestions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Issue** | **Status** | **Notes (Solution, Suggestion, etc.)** |
| 1 |  | Pending |  |
| 2 |  | In Progress |  |
| 3 |  | Completed |  |

# II. Test Plan

## 1 Introduction

### 1.1 Purpose

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

### 1.2 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 |  |
| DMS | Defect Management System (Fsoft tool) |  |
| IT | Integration test |  |
| PM | Project Manager |  |
| PTL | Project Technical Leader |  |
| QA | Quality Assurance |  |
| SRS | Software Requirement Specification |  |
| ST | System test |  |
| TP | Test Plan |  |
| TC | Test Case |  |
| TR | Test Report |  |
| UAT | User Acceptance test |  |
| UT | Unit test |  |

### 1.3 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 |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 2 Scope of Testing

### 2.1 Target of Testing

*[List the main objectives of the testing activities]*

*<<Functional items and Non-functional items will be verified by development team, then be validated and approved by FU lecturer via project defence stages. The function items and none-functional items defined in the following section>>*

### 2.2 Test Stages

*[Clearly state the stage in which the test will be executed in the project and what stages project team will be responsible for and what stage customers/end user will be responsible for. Below is a sample]*

| No | Test Stages | Description |
| --- | --- | --- |
| 1 | Unit Test | Unit Test will be performed by *<<FPT development team>>.* |
| 2 | Integration Test | Integration Test will be performed by *<<FPT QA team>>.*  After the Unit Test is finished, testers will execute the UT Gate based on the UT Gate checklist for each function. Integration Test will only start if the result of UT Gate is Passed.  This test stage focuses on specific areas of use cases when all requirements are completed, integration test should be performed to ensure all components incorporate well. |
| 3 | System Test | System Test will be executed by *<<FPT QA team>>.*  Testers will perform complete, end-to-end system testing staged in pre-production environment to validate that functions and system interfaces perform properly in production environment. |
| 4 | Acceptance Test | **Acceptance Test will be conducted by <***Customer’s name or end user’s name >.*  *<<FPT’s >>* responsibilities during Acceptance Test phase are:   * Facilitate completion on the application deployment * Support fixing bugs * Support Final User Acceptance Test |

### 2.3 Test Assumptions

*[List out and describe all testing levels that would be performed in the project: functional testing, non-functional testing, structural testing, changes testing, etc.]*

## 3 Requirements for Test

### 3.1 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.]*

#### 3.1.1 Functional Items

*[Enter a high level list of features and functions to be tested/not tested]*

#### 3.1.2 Non-functional Items

*[List all non-functional item required for testing]*

### 3.2 Acceptance 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/Weighted defects/KLOC*

*- Statement coverage: 90%*

*- Branch coverage: 100%*

*- Path coverage: 100]*

## 4 Test Strategy

### 4.1 Test Model

*[Provide the test model for your project which fits to your selected software development process model, i.e V-Model, Iteractive, etc.]*

### 4.2 Type of Testing

#### 4.2.1 Functional 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.  - 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> |

#### 4.2.2 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. > |

#### 4.2.3 Data and Database Integrity Testing

[*The databases and the database processes should be tested as a subsystem within the Project. These subsystems should be tested without the target-of-test’s User Interface as the interface to the data.  Additional research into the Database Management System (DBMS) needs to be performed to identify the tools and techniques that may exist to support the testing identified below. ]*

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

#### 4.2.4 Performance testing

*[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.*

*Note:  Transactions below refer to “logical business transactions”.  These transactions are defined as specific use cases that an actor of the system is expected to perform using the target-of-test, such as add or modify a given contract. ]*

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

#### 4.2.5 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.   The goal of load testing is to determine and ensure that the system functions properly beyond the expected maximum workload.  Additionally, load testing evaluates the performance characteristics, such as response times, transaction rates, and other time sensitive issues).]*

*[Note:  Transactions below refer to “logical business transactions”.   These transactions are defined as specific functions that an end user of the system is expected to perform using the application, such as add or modify a given contract.]*

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

#### 4.2.6 Stress Testing

*[Stress testing is a type of performance test implemented and executed to find errors due to low resources or competition for resources. Low memory or disk space may reveal defects in the target-of-test that aren't apparent under normal conditions. Other defects might result from competition for shared resources like database locks or network bandwidth. Stress testing can also be used to identify the peak workload the target-of-test can handle. ]*

*<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. > |

#### 4.2.7 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.  Volume Testing also identifies the continuous maximum load or volume the target-of-test can handle for a given period.  For example, if the target-of-test is processing a set of database records to generate a report, a Volume Test would use a large test database and check that the software behaved normally and produced the correct report. ]*

|  |  |
| --- | --- |
| 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? > |

#### 4.2.8 Security and Access Control Testing

*[Security and Access Control Testing focus on two key areas of security:*

*   Application-level security, including access to the Data or Business Functions*

*    System-level Security, including logging into or remote access from the system.*

*Application-level security ensures that, based upon the desired security, actors are restricted to specific functions or use cases, or are limited in the data that is available to them.  For example, everyone may be permitted to enter data and create new accounts, but only managers can delete them. If there is security at the data level, testing ensures that” user type one” can see all customer information, including financial data, however,” user two” only sees the demographic data for the same client.*

*System-level security ensures that only those users granted access to the system are capable of accessing the applications and only through the appropriate gateways. ]*

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

#### 4.2.9 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>.*     <*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: |  |

#### 4.2.10 White-box Testing

*[White-box testing is a necessary aimed at showing that code has been covered, no redundant and valid coding logic.]*

|  |  |
| --- | --- |
| Test Objective: | White-box testing is to verify internal detailed structure of the software, to make sure that all the statement, paths or branches have been run at least one time and correct coding logic. This activity focus on discovering faults in source codes related to redundant statement or invalid coding logic issues. |
| Technique: |    <Desk debug>.     <Use state machine testing **>**     < Data flow testing>   < Control flow testing> |
| Completion Criteria: |    <All statement are performed and passed>     <All paths are executed at least one and valid coding logic>     <All branches are executed at least one and valid coding logic> |
| Special Considerations: |  |

### 4.3 Test stage

*[Clearly state the stage in which the test will be executed. Identified below are the stages in which common test are executed]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of Test** | **Stage of Test** | | | |
| ***Unit*** | ***Integration*** | ***System*** | ***Acceptance*** |
| <Function Test > | X | X | X | X |
| <White-box Test> | 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 |  |
| <Data integrity test> |  | X | X |  |

### 4.4 Tools

*[List all tools used for testing activities]*

## 5 Resources

### 5.1 Human Resources

*[List and provide the details on roles and responsibilities of the project members who would involve in testing works, in the table format as below]*

|  |  |  |
| --- | --- | --- |
| **Worker/Doer** | **Role** | **Specific Responsibilities/Comments** |
|  |  |  |
|  |  |  |

### 5.2 Software and Hardware

*[List and provide the details about the tools (software, hardware, infrastructure) which the project would use for testing. The information can be provided in the table format as below]*

|  |  |  |  |
| --- | --- | --- | --- |
| **Purpose** | **Tool** | **Provider** | **Version** |
|  |  |  |  |
|  |  |  |  |

## 6 Test Schedule

### 6.1 Test Milestones

*[Separate test milestones, which should be identified to communicate project status accomplishments. The information can be provided in the table format as below]*

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone Task** | **Effort (md)** | **Start Date** | **End Date** |
|  |  |  |  |
|  |  |  |  |

### 6.2 Deliverables

*[Define delivery of Testing. Refer to the Project Management Plan for more details]*

|  |  |  |
| --- | --- | --- |
| No | Deliverables | Due Date |
| 1 |  |  |
| 2 |  |  |

# II Test Cases

*[Prepare the details on the test cases following the provided template*

* *Unit Test Cases: Report5\_Unit Test Case.xls*
* *Other Test Cases: Report5\_ Test Case.xls]*

# III Test Reports

*[Provide the test result, statistics and the relevant test analysis for your testing in the project]*

# IV Defect Reports

*[Provide the defect report/defect log, statistics and the relevant defect analysis for your testing in the project]*