|  |  |
| --- | --- |
|  | **FPT SOFTWARE** |

**CANTEEN MANAGEMENT SYSTEM**

**Test Plan**

**TABLE OF CONTENTS**

[**1**](#_heading=h.49x2ik5) **INTRODUCTION 5**

[1.1](#_heading=h.2p2csry) Purpose 5

[1.2](#_heading=h.147n2zr) Background information 5

[1.3](#_heading=h.3o7alnk) Scope of testing 5

[1.4](#_heading=h.3dy6vkm) Constraints 5

[1.5](#_heading=h.23ckvvd) Risk list 6

[**2**](#_heading=h.ihv636) **REQUIREMENTS FOR TEST 6**

[**3**](#_heading=h.32hioqz) **TEST STRATEGY 6**

[3.1](#_heading=h.1hmsyys) Test types 9

[*3.1.1*](#_heading=h.41mghml) *Functional Testing 9*

[*3.1.2*](#_heading=h.44sinio) *Performance testing 12*

[*3.1.3*](#_heading=h.2grqrue) *Security and Access Control Testing 15*

[*3.1.4*](#_heading=h.vx1227) *Regression Testing 16*

[3.2](#_heading=h.3fwokq0) Test stage 16

[3.3](#_heading=h.1v1yuxt) Tools 17

[**4**](#_heading=h.4f1mdlm) **RESOURCE 17**

[4.1](#_heading=h.2u6wntf) Human Resource 17

[4.2](#_heading=h.19c6y18) System 17

[**5**](#_heading=h.3tbugp1) **TEST MILESTONES 17**

[**6**](#_heading=h.28h4qwu) **DELIVERABLES 18**

1. **INTRODUCTION**
   1. ***Purpose***

This is the Canteen Management System project's detailed test plan. The goal of the document is to outline the kind of tests that will be conducted and the steps that must be done throughout project testing. It talks about the following things:

|  |  |
| --- | --- |
| · Scopes of Testing  · Requirements for Testing  · Test Strategy | · Test Resources  · Test Milestones  · Test Deliverables |

* 1. ***Background information***

- School Facilities: Nowadays, FPT has become one of the most popular universities in Vietnam. Not only does it have many cutting-edge technologies in teaching, but it also includes an enormous amount of wonderful infrastructure like buildings, libraries, canteens, and so on.

- Canteen Crisis: However, the canteen, the main location where students could gain back energy, has received a lot of bad feedback. In spite of having multiple options to choose from for meals, many students are concerned about food quality. It is becoming usual to see bad responses about cooking services: from poor quality food origin to bread being spoiled. People are also complaining about the time it takes to order food.

- Solution: In order to solve this crisis, our capstone project will provide a canteen management system that will help students and school staff to easily spectate and evaluate food quality from those restaurants.

- Students: It will provide a detailed summary of the od origin and daily restaurant menu, which is believed to help students to make healthy decisions. By looking at the ingredient source, students could have a better understanding of the cooking process before making any purchase. It could also solve the amount of time wasted waiting for food by online ordering. In the future, we expect to add a Smart Calorie Evaluator, which could help users to control the number of calories received and plan for a suitable diet.

- School: School censors can weekly check the food origin of the restaurant published by the owner of the canteen using our system: The owner has to send their ingredient source for next week's menu to the school’s censor to evaluate before getting permission to publish it to the system.

* 1. ***Scope of testing***

The scope of the test will be limited to testing three applications: the HTML 5 applications and two hybrid applications (on iPhone and Android).

1. Target of Test

Functional items and Non-functional items will be verified and passed by FPT development team, then be validated and approved by CAMS via test stages

1. Test Stage

| **No** | **Test Stages** | **Description** |
| --- | --- | --- |
| 1 | Unit Test | Unit Test will be performed by team 5 |
| 2 | Integration Test | Integration Test will be performed by team 5  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 team 5  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 | FPT’s responsibilities during Acceptance Test phase are:   * Facilitate completion on the application deployment * Support fixing bugs * Support Final User Acceptance Test |

* 1. ***Constraints***

They may be:

* Testing deadlines can only be reached if development progress is on schedule.
* When the system passes the unit test inspection, testing can begin.
* Requirements testing must go through at least one round.
* Canteen Management assistance is required to set up the production setup for all testing.
  1. ***Risk list***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| 1 | Member illness | High | Medium | Other members ought to cooperate to meet the work of the absent member. |
| 2 | Disagreements among members | Medium | Medium | All members must meet immediately to solve the problems. |
| 3 | Lack of skills | High | Medium | All members have to train each other. Besides, everyone must learn by themself every day. |
| 4 | Unclear business requirement | High | Medium | Must do actual survey, study business requirement carefully |
| 5 | Having a difficult problem | High | Medium | All members have to discuss the issues to decide the simplest solution.  Raise the supervisor and people with experience regarding the problems. |
| 6 | Requirements change | High | Medium | All members have to discuss carefully  at the start of every iteration to outline scope and requirement. |
| 7 | Meeting with the mentor | High | Medium | Break the 3 sessions in a week: Monday, Wednesday, Friday. |

1. **REQUIREMENTS FOR TEST**

*2.1. Acceptance Test Criteria*

* Functional Item

***System Admin***

* Authentication account
* Manage users’ accounts
* Manage data system

***Censor***

* Authentication account
* Manage owner
* Manage canteen
* Manage request’s the menu

***Owner***

* Authentication account
* Manage staff account
* Manage dishes
* Manage revenue
* Manage feedback of user
* Manage order

***Staff***

* Authentication account
* Manage meals
* Manage to order of customer

***Customer***

* Authentication account
* Manage user account
* Ordering food
* Feedback and Rating

* Non-functional Items

● The language is English.

● Links and buttons are designed to easily click.

● User interface should be clear and easy to use.

● The images have a maximum size of 3MB.

● The system is designed so that users will be easy to get used to themselves and use functions in an exceedingly short time. Main functions are unionized into tabs for easier access.

● The maximum number of steps to complete 1 activity from seeing the screen is 5.

● The speed of the connection between the client and server is 12MB/s **4.2. Security**

*2.2. Acceptance Test Criteria*

|  |  |  |
| --- | --- | --- |
| No | Test Stages | Qualified ratios |
| 1 | Unit Test | All unit test cases must be run and receive a passing score of 80% in order to move forward. Retests and fixes for all flaws are required. 8 bugs on average per KLOC. |
| 2 | Integration Test | All test cases must be run in order for this stage to be passed. Retests and fixes for all flaws are required. 3 bugs on average per KLOC. |
| 3 | System Test | All test cases must be run and have a 91% passing grade in order to move forward. Retests and fixes for all flaws are required. 0.7 bugs on average per KLOC. |
| 4 | Acceptance Test | Canteen Management will carry out and approve the acceptance test. |

1. **TEST STRATEGY**
   1. ***Test types***

### Function Testing

|  |  |
| --- | --- |
| Test Objective: | Verify the application and its internal processes by interacting with the application via the Graphical User Interface (GUI) and analyzing the outputs or results |
| Technique: | - Testers will create test scenarios against the requirements provided by customer. Test scenarios will be created based on black box test technique.  - Testers execute test based on test scenarios and create report. Common defects will be collected for improved checklist.  - Execute each case, using valid and invalid data, to verify the following:  Get the expected results when valid and invalid data is used  Valid input data is updated correctly into database  The appropriate errors or warning messages are displayed when invalid data is used  - Execute each case, using boundary data, to verify the following:  Get the expected results when boundary data is used  Data is updated correctly to database  The appropriate errors or warning messages are displayed when invalid data is used  - Each valid data input is updated correctly into database.  - Each business rule is properly applied. |
| Completion Criteria: | All functional test cases have been executed to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules, and passed  The appropriate activities will be performed when valid data is used  The corresponding error/warning message mechanism is applied for each specific case  All bugs found must be fixed |
| Special Considerations: | Functional testing will NOT be started in case of developers have not executed unit test before passing application to testers |

### User Interface Testing

|  |  |
| --- | --- |
| Test Objective: | Navigation through the target-of-test properly reflects business functions and requirements, including screen to screen, field-to-field, and use of access methods. Objects and characteristics, such as menus, size, position, state, and focus conform to standards. |
| Technique: | - Testers will create specific test scenarios against UI prototypes approved by customer.  - Testers execute test based on test scenarios and create report. Common defects will be collected for improved checklists.  - Execute each case, using valid, invalid and boundary data, to verify the expected results display when valid, invalid and boundary data is used. |
| Completion Criteria: | - All GUI test cases have been executed and passed  - Any defects related to GUI are fixed, except those accepted by customer  - All defects found by tester are addressed in DMS |
| Special Considerations: | N/A |

### Load Testing

|  |  |
| --- | --- |
| Test Objective: | Verify the application and its internal processes should be met:  In the normal condition could be 100 concurrent users or less, each page should load in four seconds or less  In the stress condition could be more than 100 concurrent users, each page should load in 12 seconds or less |
| Technique: | - Testers will create test scenarios, test scripts against the requirements provided by customer. Test scenarios will be created based on black box test technique, and be supported by one of the following tools: IBM Rational Robot & Manager.  - Testers execute test based on test scenarios and create report. Common defects will be collected for improved checklists.  - Execute each case, using valid and invalid data, to verify the following on a random device:  In the normal condition could be 100 concurrent users or less, each page should load in four seconds or less  In the stress condition could be more than 100 concurrent users, each page should load in 12 seconds or less |
| Completion Criteria: | In the normal condition could be 100 concurrent users or less, each page should load in four seconds or less.  In the stress condition could be more than 100 concurrent users, each page should load in 12 seconds or less  All performance requirements must be met |
| Special Considerations: | Numbers of 100 virtual users should be available for Microsoft Visual Studio |

### Security Testing

|  |  |
| --- | --- |
| Test Objective: | Verify that the application is HIPAA & TRUSTe compliance |
| Technique: | - Testers will create test scenarios against the requirements which are based on HIPAA & TRUSTe compliance. Test scenarios will be created based on black box test technique. Refer to:  [https://www.owasp.org](https://www.owasp.org/); <http://www.wedi.org/snip/public/articles/testing_whitepaper082602.pdf>;  <http://www.macadamian.com/images/uploads/whitepapers/HIPAA_TestStrategies.pdf> for more detail.  - Testers execute test based on test scenarios and create report. Use WireShark tool to validate transaction encrypted or not. Common defects will be collected for improved checklists.  - Execute each case, using valid and invalid data, to verify the following: The expected results occur when valid HIPAA & TRUSTe compliance. |
| Completion Criteria: | All test cases have been executed to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the HIPAA & TRUSTe compliance rules, and passed |
| Special Considerations: | HIPAA & TRUSTe compliance requirement |

### Regression Testing

|  |  |
| --- | --- |
| Test Objective: | Verify the application on new build/ after bug fixing, and be sure that other functions is not affected by fixed parts each iterations |
| Technique: | - Testers will select some test scenarios against functional test  - Testers execute test based on test scenarios and create report. Common defects will be collected for improved checklists.  - Execute each case, using valid and invalid data, to verify the following:  Get the expected results when valid and invalid data is used  Valid input data is updated correctly into database  The appropriate errors or warning messages are displayed when invalid data is used  - Execute each case, using boundary data, to verify the following:  Get the expected results when boundary data is used  Data is updated correctly into database  The appropriate errors or warning messages are displayed when invalid data is used  - Each valid data input is updated correctly into database.  - Each business rule is properly applied |
| Completion Criteria: | Specified function test cases have been executed to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules, and passed  The appropriate activities will be performed when valid data is used.  The corresponding error/warning message mechanism is applied for each specific case.  All bugs found must be fixed.  Some of other functions is not affected by fixed code |
| Special Considerations: | N/A |

* 1. ***Test stage***

| **Type of Tests** | **Stage of Test** | | | |
| --- | --- | --- | --- | --- |
| **Unit** | **Integration** | **System** | **Acceptance** |
| <Functional Tests  (Function, User Interface)> | X | X | X | X |
| <Performance Tests  (Performance profiles of individual components)> | X | X |  |  |
| <Performance Tests  (Load, Stress, Contention)> |  |  | X | X |
| <Reliability  (Integrity, Structure)> | X | X |  |  |

* 1. ***Tools***

|  |  |  |  |
| --- | --- | --- | --- |
| ***Purpose*** | ***Tool*** | ***Vendor/In-house*** | ***Version*** |
| ***Documenting*** | ***MicrosoftWord,Google Drive/Docs/Sheets*** | ***Microsoft*** | ***2010*** |
| ***Unit Testing*** | ***IntelliJ*** | ***JetBrains*** |  |
| ***API Testing*** | ***Postman*** | ***Postman Inc.*** |  |
| ***Database*** | ***MySQL*** | ***Oracle Corporation*** | ***2019*** |

1. **RESOURCE**
   1. ***Human Resource***

|  |  |  |
| --- | --- | --- |
| Worker/Doer | Role | Specific Responsibilities/Comments |
| Nguyễn Đức Anh | Test Leader | Manage test resources and assign test tasks  Create Test Plan, Test Cases (IT, ST), Test Scripts (IT, ST)  Review Test Data  Create Test Reports |
| Nguyễn Bá Hoàn | Tester | Review Test Cases (IT, ST)  Create Test Data and Execute Test (IT, ST)  Report Test Results |
| Nguyễn Minh Hiếu | Quality Assurance | Final Inspection Test Cases, Test Plan, Test Reports |
| Mạc Huyền | Project Manager | Approve Test Cases (UT, IT, ST), Test Plan, Test Results, Test Reports |
| Phi Hùng | Project Technical Leader | Review Test Cases (UT, IT, ST) |
| Tào Vũ Anh | Developer | Create UT Cases, UT Reports |

* 1. ***System***

The FMO HTML5 web-based application and iPhone, Android Hybrid applications will require testing on the following iPhone and Android devices:

|  |  |
| --- | --- |
| Hardware | OS Version |
| **iPhone 10-11-12-14** | **iOS 15** |
| **Samsung Galaxy** | **Android OS, Fuchsia OS** |

|  |  |
| --- | --- |
| Software | Version |
| **Microsoft Windows Server** | 2019 |
| **Microsoft SQL Server** | 2019 |
| **Microsoft .NET Framework** | 4.5, 4.8 |

1. **TEST MILESTONES**

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone Task | Effort (pd) | Start Date | End Date |
| Create Test Plan | 5 | 9/2 | 15/2 |
| Review & Update Test Plan | 2 | 16/2 | 17/2 |
| Create IT&ST Test Cases | 13 | 17/2 | 21/2 |
| Review & Update IT&ST Test Cases | 2 | 21/2 | 27/2 |
| Create & Execute & Report UT for Software Package | 15 | 27/2 | 9/3 |
| Create test data for Software Package | 1 | 10/3 | 17/3 |
| UT Gate for Software Package | 0.5 | 18/3 | 19/3 |
| Execute IT for Software Package | 16 | 20/3 | 1/4 |
| Execute ST for Software Package | 4 | 2/4 | 4/4 |
| Create IT&ST Test Report for Software Package | 1 | 5/4 | 5/4 |
| Create & Execute & Report UT for Software Package version 1.1 | 22 | 6/4 | 30/4 |
| UT Gate for Software Package version 1.1 | 0.5 | 1/5 | 2/5 |
| Create test data for Software Package version 1.1 | 1 | 3/5 | 5/5 |
| Execute IT for Software Package version 1.1 | 17 | 6/5 | 16/5 |
| Execute ST for Software Package version 1.1 | 5 | 17/5 | 19/5 |
| Create IT&ST Test Report for Software Package version 1.1 | 1 | 20/5 | 20/5 |
| Create & Execute & Report UT for Software Package version 1.2 | 16 | 21/5 | 10/6 |
| UT Gate for Software Package version 1.2 | 0.5 | 11/6 | 11/6 |
| Create test data for Software Package version 1.2 | 1 | 12/6 | 14/6 |
| Execute IT for Software Package version 1.2 | 20 | 15/6 | 25/6 |
| Execute ST for Software Package version 1.2 | 3 | 25/6 | 27/6 |
| Create IT&ST Test Report for Software Package version 1.2 | 0.5 | 28/6 | 28/6 |
| Execute System Test for Software Package final version | 8 | 29/6 | 3/7 |
| Create Test Report for Software Package final version | 1 | 4/7 | 4/7 |

# 6 DELIVERABLES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Deliverables | Delivered Date | Delivered by | Delivered to |
| 1 | Test Plan | 15/2 | FPT |  |
| 2 | Test Reports for Software Package version 1.0 | 20/5 | FPT |  |
| 3 | Test Reports for Software Package version 1.1 | 25/2 | FPT |  |
| 4 | Test Reports for Software Package version 1.2 | 28/6 | FPT |  |
| 5 | Test Reports for Software Package final version | 4/7 | FPT |  |