|  |
| --- |
| C:\Users\Tsubaki Yukino\Desktop\FlyAwayPlus Project\Capture.JPG |

**BAKERY STORE ONLINE**

Report #2 – Test Plan

|  |  |  |
| --- | --- | --- |
| **Bakery Store Online** | | |
| **Group member** | Phạm Thị Minh Phương | SE03773 |
| Cao Anh Tuấn | SE03737 |
| Hoàng Anh Phương | SE03715 |
| Ngô Trung Kiên | SE03736 |
| Nguyễn Xuân Hà | SE03484 |
| **Supervisor** | Mr. Nguyễn Văn Sang | |
| **Project code** | BSO | |

**Hà Nội, 10/2017**

Record of change

\*A - Added M - Modified D - Deleted

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Effective Date | Changed Items | A\* M, D | Change Description | Reason for change | New Version |
| 19/10/2017 | Create test plan | A | First Version | Create test plan | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

SIGNATURE PAGE

AUTHOR: Phạm Thị Minh Phương 06/10/2017

Project Manager

REVIEWERS: Cao Anh Tuấn /10/2017

Team member

APPROVAL: Nguyễn Văn Sang --/--/--17

Supervisor

TABLE OF CONTENTS

[1 introduction 5](#_Toc9967)

[*1.1* Purpose 5](#_Toc10454)

[*1.2* Definitions, Acronyms, and Abbreviations 5](#_Toc12650)

[*1.3* References 6](#_Toc32710)

[*1.4* Background information 6](#_Toc25499)

[*1.5* Scope of testing 6](#_Toc30163)

[*1.6* Constraints 8](#_Toc18097)

[*1.7* Risk list 8](#_Toc15220)

[*1.8* Training needs 8](#_Toc27976)

[2 Requirements for Test 11](#_Toc5349)

[*2.1* Test items 11](#_Toc24175)

[*2.2* Acceptance Test Criteria 13](#_Toc6558)

[3 TEST STRATEGY 14](#_Toc21982)

[*3.1* Test types 14](#_Toc30757)

[3.1.1 Function Testing 14](#_Toc27369)

[3.1.2 User Interface Testing 15](#_Toc10692)

[3.1.3 Data and Database Integrity Testing 15](#_Toc31726)

[*3.2* Test stages 16](#_Toc7346)

[4 RESOURCE 16](#_Toc27215)

[*4.1* Human Resource 16](#_Toc20647)

[5 Test environment 18](#_Toc249)

[*5.1* Hardware 18](#_Toc5103)

[*5.2* Software 18](#_Toc4270)

[*5.3* Infrastructure 18](#_Toc7097)

[6 TEST MILESTONES 19](#_Toc20531)

[7 DELIVERABLES 20](#_Toc12681)

# introduction

## Purpose

This is the comprehensive test plan of the BSO project. The purpose of the document describes scopes of test and activities which need to be taken during test process of project. It addresses the following items: Scopes of Testing, Requirements for Testing, Test Strategy, Test Resources, Test Environment, Test Milestones and Deliverable.

## Definitions, Acronyms, and Abbreviations

| Abbreviations | Description | Note |
| --- | --- | --- |
| BSO | Bakery Store Online |  |
| GUI | Graphic User Interface |  |
| IT | Integration Test |  |
| KLOC | 1000 line of code |  |
| PM | Project Manager |  |
| QA | Quality Assurance |  |
| SRS | Software Requirement Specification |  |
| ST | System Test |  |
| TC | Test Case |  |
| TL | Test Leader |  |
| TP | Test Plan |  |
| TR | Test Report |  |
| UT | Unit Test |  |

## Background information

The target of testing is ensured all functions will be run correctly as SRS description. In addition, restrict maximum of defect during the user access in the application. To do this target, website will have to:

* Passed the stages of testing: Integration Testing, System Testing, Acceptance Testing
* Passed the types of testing: Function Testing, User Interface Testing , Data and Data Integrity Testing
* Run normally in required devices/browsers.

## Scope of testing

Bakery Store will be tested by 3 phases:

**Phase 1: Integration testing**

* After finishing component testing, integration testing will be performed by testers.
* Material are integration test cases, high- level design and test tools.
* Integration test focuses on specific areas of use cases when all requirements are completed.
* Integration test should be performed to ensure all components incorporate well.
* When executing integration testing, if any bugs are found, testers have to log on “Defect Log Management” file and assign to developer fix it and redo this process until it is correct.

*Rule for filling test result:*

|  |  |
| --- | --- |
| Test result pass | Pass |
| Test result fail | Fail |
| Do not test | Untested |
| Cannot test | N/A (Not available) |

**Phase 2: System testing**

* After finishing integration testing and developers collect all functions and items, testers will be performed system testing, it means doing test whole system.
* Material area system test case, SRS
* If any bugs are found, developers have to fix and testers will verify them. System test is ended only when test cases are passed and no bug is found.

*Rule for filling test result:*

|  |  |
| --- | --- |
| Test result pass | Pass |
| Test result fail | Fail |
| Do not test | Untested |
| Cannot test | N/A (Not available) |

**Phase 3: Acceptance testing**

* Base on customer/user requirement specification, system is tested again, for ensure there is not lacking or mistake any requirement.
* If there is any problem, developers have to fix/update and tester will verify them.
* Acceptance testing is ended only when whole system met requirement specification.

## Constraints

* Deadline for testing only can be met if development progress is on time.
* At least one round of testing must be performed for requirements.
* Have more environments should be tested: Window XP, Window 8… and more browsers: Firefox 30 and Google Chrome 40…

## Risk list

* Depends on the UI design and develop progress.
* Not enough time to write enough test cases, execute test or re-test for fixed bug.
* Tester can be ill during the testing phase.

## Training needs

Bakery Store follows V-Model process:



**Figure 2-1**: V-Model

Testing progress is divided to 5 phases include: Unit test, Component test, Integration test, System test and Acceptance test

* Integration test:
  + Integration testing is a level of the software testing process where individual units or component are combined and tested as a group.
  + The purpose is to expose faults in the interaction between integrated units.
  + Integration testing is performed after component testing
  + Integration testing will be done by tester
  + There are two methods of doing integration testing: Bottom-up Integration testing and Top Down Integration testing:

|  |  |
| --- | --- |
| No | Integration Testing Method |
| 1 | Bottom-up integration  This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules. |
| 2 | Top- Down integration  This testing, the highest-level modules are tested first and progressively lower-level modules are tested after that |

**Table 1-3:** Integration test

* System test:
  + System Testing is a level of the software testing process where a complete, integrated system is tested
  + The purpose is to evaluate the system’s compliance with the specified requirements
  + System testing is performed after integration testing
  + System testing will be done by tester
* Acceptance test:
  + Acceptance testing is performed after system testing
  + Acceptance testing will be performed by the test leader and team leader.
  + The acceptance test will be done for a period of 1 weeks after completion of the system test process

# Requirements for Test

## Test items

|  |  |  |
| --- | --- | --- |
| UC No. | **Group Of Functions** | **Function** |
| **Administrator** | | |
| UC001 | Common | Admin Login |
| UC002 | Admin Logout |
| UC003 | Admin Forgot Password |
| UC004 | Manage Staff | View list staff |
| UC005 | View staff profile |
| UC006 | Activate staff account |
| UC007 | Deactivate staff account |
| UC008 | Search staff account |
| UC009 | Update staff account |
| UC010 | Insert a new staff account |
| UC011 | Manage Customer | View list customer |
| UC012 | Activate customer account |
| UC013 | Deactivate customer account |
| UC014 | View customer profile |
| UC015 | Search customer account |
| UC016 | Manage Benefit | Search benefit |
| UC017 | View benefit |
| UC018 | Manage Cake | Search cake |
| UC019 | Add cake |
| UC020 | Update cake |
| UC021 | Block Cake |
| UC022 | Unblock Cake |
| UC023 | View information cake |
| UC024 | Manage blog | View list blog |
| UC025 | Search blog |
| UC026 | Add blog |
| UC027 | Block blog |
| UC028 | Unblock blog |
| UC029 | Manage Event | Search event |
| UC030 | Add event |
| UC031 | Block event |
| UC032 | Unblock event |
| UC033 | Manage Category | Add Category |
| UC034 | Block Category |
| UC035 | Unblock Category |
| **Staff** | | |
| UC036 | Common | Log in |
| UC037 | Log out |
| UC038 | Forgot Password |
| UC039 | Manage Bill | Search order |
| UC040 | Confirm Order |
| UC041 | Unconfirm Order |
| UC042 | Manage Cake | View Cake |
| UC043 | Search Cake |
| UC044 | Personal information | Update Information |
| UC045 | Change Password |
| UC046 | Manage Introduction | Create Introduction |
| UC047 | Update Introduction |
| **Customer** | | |
| UC048 | Common Module | Register |
| UC049 | Login |
| UC050 | Logout |
| UC051 | Forgot Password |
| UC052 | View About |
| UC053 | View Contact |
| UC054 | View Blog |
| UC055 | View Catgory |
| UC056 | Order | Order Cake |
| UC057 | Check Out |
| UC058 | Personal Information | Edit Profile |
| UC059 | Change Password |
| UC060 | Purchase History |

## Acceptance Test Criteria

* + Criteria for Unit test of Development team, for Test team accepts to start testing:
* Number of TC/KLOC: 40TC/KLOC
* Number defects/KLOC: 3-4 defects/KLOC
* Statement coverage: 100%
* Branch coverage: 100%
* Path coverage: 100%
  + Criteria for Integration test:
* Number of TC/KLOC: 30 TC/KLOC
* Number defects/KLOC: 2-3 defects/KLOC
  + Criteria for System test:
* Number of TC/KLOC: 20 TC/KLOC
* Number defects/KLOC: 4-6 defects/KLOC
  + Criteria for Acceptance test:
* Number defects/KLOC: 1-2 defects/KLOC

# TEST STRATEGY

## Test types

### Function Testing

* + Functional testing is a type of software testing whereby the system is tested against the functional requirements/specifications.
  + Functions are tested by feeding them input and examining the output. Functional testing ensure that the requirements are properly satisfied by the website. This type of testing is not concerned with how processing occurs, but rather, with the results of processing.
  + During functional testing, [Black Box Testing](http://softwaretestingfundamentals.com/black-box-testing/) technique is used in which the internal logic of the system being tested is not known to the tester.

|  |  |
| --- | --- |
| **Test Objective:** | The type of this test is to ensure proper target-of-test functionality, including user interaction, all function defined in specification document implemented correctly. |
| **Technique:** | Executing 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. |
| **Completion Criteria:** | - All planned tests have been executed.  - All identified defects have been addressed and closed. |
| **Special Considerations:** | Testing may be stopped when  Time runs out  A certain number of defects found  Test coverage > 97%  Stop when testing becomes unproductive |

**Table 3-1:** Function Testing

### User Interface Testing

* GUI testing is the process of ensuring proper functionality of the GUI for a given web and making sure it conforms to its written specifications.
* GUI testing evaluates design elements such as layout, colors, [fonts](http://whatis.techtarget.com/definition/font), font sizes, labels, text boxes, text formatting, captions, buttons, lists, icons, links, content and more.

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

**Table 3-2:** GUI Testing

### 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. |

**Table 3-3:** Data and Data Integrity Testing

## 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 Testing | X | X | X | X |
| User Interface Testing | X |  | X | X |

# RESOURCE

## Human Resource

|  |  |  |
| --- | --- | --- |
| **Worker/Doer** | **Role** | **Specific Responsibilities/Comments** |
| PhuongPTM | Test Leader | Manage Test Resource and assign test tasks.  Create and review Test Plan.  Create and review Test Case.  Execute test. |
| PhuongHA | Tester | Create and review Test Case.  Execute test.  Create Test view points |

# Test environment

## Hardware

## Software

|  |  |  |
| --- | --- | --- |
| Name | Purpose | Detail |
| Test Plan | Managing test | Microsoft Word 2010, 2013 |
| Test case | Executing test | Microsoft Excel 2010, 2013 |
| Test report, Test checklist | Tracking test | Microsoft Excel 2010, 2013 |
| Chrome | Executing test | Chrome 40.0 |
| Firefox | Executing test | Firefox |

**Table 5-2:** Software

## Infrastructure

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Purpose** | **Detail** | **Vendor/In-house** | **Version** |
| DDL\_Defect Log Management\_v1.0\_EN | Tracking bug during testing time | Microsoft Excel 2013, 2010 | FPT-University | v1.0 |
| Test Effort | Effort execute test | Microsoft Excel 2013, 2010 | FPT-University | v1.0 |

**Table 5-3:** Infrastructure

# TEST MILESTONES

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone Task** | **Effort (pd)** | **Start Date** | **End Date** |
| Create Test Plan | 6 |  |  |
| Review & Update Test Plan | 2 |  |  |
| Create Component Test case | 12 |  |  |
| Review & Update Component Test Case | 2 |  |  |
| Create Integration Test case | 2 |  |  |
| Review & Update Integration Test Case | 2 |  |  |
| Create System Test case | 6 |  |  |
| Review & Update System Test Case | 2 |  |  |
| Execute Component test phase 1 | 1 |  |  |
| Execute Component test phase 2 | 1 |  |  |
| Execute Integration test phase 1 | 2 |  |  |
| Execute Integration test phase 2 | 2 |  |  |
| Execute System test phase 1 | 4 |  |  |
| Execute System test phase 2 | 4 |  |  |

**Table 6.1:** Test milestones

# DELIVERABLES

| **No** | **Deliverables** | **Responsibilities** | **Delivered Date** |
| --- | --- | --- | --- |
| 1 | Test Plan | Tester |  |
| 2 | Component Test case | Tester |  |
| 3 | Integration Test case | Tester |  |
| 4 | System Test case | Tester |  |
| 5 | Defect Log Management | All members |  |
| 6 | Test report | PM |  |

**Table 7-1:** Deliverables