

Test Plan

General Test Plan for ACMS



2017-10-15

ACMS\_e\_J

Historical Versions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ver Num. | Updates | Chapters | Update Date | Author |
| 1.0.0 | Create the test plan | all | 2017-10-15 | CAO |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Menu

[1. Document Information 1](#_Toc495978273)

[1.1. Overview 1](#_Toc495978274)

[1.2. Test Platform 1](#_Toc495978275)

[2. Test Protocol 1](#_Toc495978276)

[2.1. Test Process 1](#_Toc495978277)

[2.2. Test Method 2](#_Toc495978278)

[3. Test Case 2](#_Toc495978279)

[3.1. Rules for Test Cases 2](#_Toc495978280)

[3.1.1. Serial Number 2](#_Toc495978281)

[3.1.2. Equivalence Class 2](#_Toc495978282)

[3.2. List of Test Cases 3](#_Toc495978283)

[3.2.1. FRONT 3](#_Toc495978284)

[3.2.2. BACK 4](#_Toc495978285)

[4. Summary 4](#_Toc495978286)

[4.1. Testing Result 4](#_Toc495978287)

[4.2. Analysis 4](#_Toc495978288)

# Document Information

## Overview

This document is used to guide the project activity of testing, mainly including methods for testing, test cases and environment for testing (both software and hardware). Test cases would be executed after the complement of the whole system and results should be recorded in this document. As nearly all test cases written in this document are for system testing and acceptance testing, there would be several other documents which would be used for unit testing and integration testing.

This document will be followed in principle and should not be broken unless requirement change occurs or other unexpected accident happens. Updates for this document should be released after the publish of the new document of requirement analyses without dely. Both contents and the Historical Versions Table are required to be updated.

## Test Platform

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Frontend | Backend |
| Hardware | CPU | Intel(R) Core(TM) i7 4720HQ |  |
| Memory | DDR3 1600MHz 8G |  |
| Hard Disk | CHA-M2B7-M256 |  |
| Software | Operating System | Windows 10 Enterprise 15063 |  |
| Browser/Server | Chrome 61.0.3163.100 | Apache |
| Others |  | PHP7 |

# Test Protocol

## Test Process

The whole project activity of testing will be divided into three parts, which is unit testing, integration testing and system testing respectively. A unit testing will be executed after the complement of a single class by any other two coders or testers, except the author. The result will be delivered to the author and be used as one of the references during the debug procedure. Only if there’s no bug detected in the latest testing, the coder will be allowed to start a new class.

An integration testing will be executed after the complement of an individual module under the help from coders by testers. After passing the integration testing, a milestone can be settled in the schedule.

The system testing will be executed after the complement of the whole system. Testers should follow the instructions of test cases written in this document strictly. Before finish testing, each statement written in the requirement analysis document should be check again if it is completely achieved.

## Test Method

|  |  |  |
| --- | --- | --- |
| Unit Testing | Integration Testing | System Testing |
| White Box Test | Gray Box Test | Black Box Test |

# Test Case

## Rules for Test Cases

### Serial Number

A serial number for a test case is divided into five parts, which is connected by an underline respectively. The first part shows the global attribution of the test case, which has a binary value of FRONT and BACK. The second part shows that which module this test case will execute on. The value should be the name of the module. The third part shows the name of the secondary classification in the module. If there’s no secondary classification, the value should be ALL. The forth part shows the name of the class, which suggests that all classes will belong to a module. The fifth part is a three-digit number starting from 001, which differs multiple test cases executed on a same class.

Test cases for modules only has a three-part serial number, which is global attribution, module name and three-digit number.

### Equivalence Class

Inputs to a module or a class should be divided into different equivalence classes. The division mainly relays on experiences of testers. The amount of test cases to a single module or a class should be the product of equivalence classes on each parameter. That means both valid and invalid values should be tested.

Test cases for classes will be written in other documents. This document only includes test cases using in system testing. For convenience, test cases execute on a same module are allowed to be merged.

# Test Plan

## StaffSys\_UC001

1. Check whether “create a note” function works.
2. Check whether a creative staff can read notes.
3. Check whether a purchasing assistant can read notes.
4. Check whether an account manager can read notes.
5. Check whether a senior manager can read notes.
6. Check whether a javascript can be inject into a note.
7. Check whether a user can read notes without login.

## StaffSys\_UC003

1. Check whether a purchasing assistant can manage a campaign cost.
2. Check whether an account manager can manage a campaign cost.
3. Check whether a senior manager can manage a campaign cost.
4. Check whether a creative staff can manage a campaign cost.
5. Check whether a non-number character can be insert into the cost field.

## StaffSys\_UC004

1. Check whether a purchasing assistant can manage a staff pay rate.
2. Check whether an account manager can manage a staff pay rate.
3. Check whether a senior manager can manage a staff pay rate.
4. Check whether a creative staff can manage a staff pay rate.
5. Check whether a non-number character can be insert into the pay rate field.

## ManagerSys\_UC001

1. Check whether an account manager can update a campaign.
2. Check whether a creative staff can update a campaign.
3. Check whether a senior manager can update a campaign.
4. Check whether a purchasing assistant can update a campaign.
5. Check whether a non-number character can be insert into the cost field.
6. Check whether a blank title can be insert into the title field.

## ManagerSys\_UC002

1. Check whether an account manager can update the staff list of a campaign.
2. Check whether a creative staff can update the staff list of a campaign.
3. Check whether a purchasing assistant can update the staff list of a campaign.
4. Check whether a senior manager can update the staff list of a campaign.
5. Check whether a blank staff can be selected.
6. Check whether a staff’s workhour can be changed.
7. Check whether a non-number character can be insert into the workhour field.
8. Check whether a unauthorized user can get the staff list.

## ManagerSys\_UC003

1. Check whether an account manager can view all campaigns which he manages.
2. Check whether an account manager can view campaigns which he doesn’t manage.
3. Check whether a senior manager can view all campaigns.
4. Check whether a purchasing assistant can view campaigns where he is in.
5. Check whether a purchasing assistant can view campaigns where he is not in.
6. Check whether a creative staff can view campaigns where he is in.
7. Check whether a creative staff can view campaigns where he is not in.

# Summary

## Testing Result

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Module | | Amount | Pass | Fail | Ratio |
| FRONT |  |  |  |  |  |
|  |  |  |  |  |
| BACK |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Analysis

\*TODO after system testing