|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document History** | |  |  |  |
| **Version Number** | **Record Date** | **Prepared/**  **Modified By** | **Reviewed By** | **Change Details** |
| 0.1 | 14/04/2014 | Tanawat, Nattakit | PF,NC | * Draft |
| 0.8 | 11/05/2014 | Tanawat, Nattakit | PF,NC | * Draft |
| 0.9 | 28/05/2014 | Tanawat, Nattakit | PF,NC | * Draft |
| 1.0 | 15/07/2014 | Tanawat, Nattakit | PF,NC | * Draft |

Document Designed by

Tanawat Sitthitan  
Nattakit Chaiwongsri  
July 31, 2014   
  
Document Approved by

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Mr. Phudinan Singkhamfu

August 27, 2014

Content

[Chapter 1 | Introduction 3](#_Toc394407584)

[1.1 Purpose 4](#_Toc394407585)

[1.2 Scope 4](#_Toc394407586)

[1.3 Objective 4](#_Toc394407587)

[1.4 Dependency Document 4](#_Toc394407588)

[1.5 Environment Test 4](#_Toc394407589)

[1.5.1 Notebook 4](#_Toc394407590)

[**Test Procedure** 5](#_Toc394407591)

[Chapter 2 | Test Deliverable 5](#_Toc394407592)

# Chapter 1 | Introduction

This test approach document describes the appropriate strategies, process, workflows and methodologies used to plan, organize, execute and manage testing of promotion provider.

## 1.1 Purpose

The purpose of Test Plan document describes the test plan for the promotion provider. In this document consists of the test cases which have to test in each progress period. The system testing covers the URS to ensure that the failures are discovered and fixed in the system.

## 1.2 Scope

Test plan explains the unit testing activities to detect the failures of implemented methods in system. Also explains the system testing activities to verify consistency between test and URS.

## 1.3 Objective

The objective of test plan of Promotion Provider is to ensure all failures are discovered and fixed. Project can satisfy all of user’s requirements and all of functions are consistently and reliability

## 1.4 Key Definition

* IEEE Institute for Electrical and Electronics Engineers. Biggest global   
   interest group for engineers of different branches and for computer   
   scientists. [IEEE90]
* Requirement (1) A condition or capability needed by a user to solve a problem or   
   achieve an objective. (2) A condition or capability that must be met   
   or possessed by a system or system component to satisfy a contract,   
   standard, specification, or other formally imposed document. (3) A   
   documented representation of a condition or capability as in   
   definition (1) or (2). [IEEE90]
* Feature Transformation of input parameters to output parameters based on   
   a specified algorithm. It describes the functionality of a product in   
   the language of the product. Used for requirements analysis, design,   
   coding, testing or maintenance.[IEEE90]
* Use Case (1) Concept to describe a system based on usage of system   
   resources by its environment. Characterized by an objective set of   
   interactions within and at the borders of that system. (2) Notation   
   from UML for describing a scenario (Usage approach, operational   
   scenario) from the perspective of its users. [IEEE90]
* Web A computer software application that is coded in a browser   
  Application supported and reliant on a common web browser to render the   
   application executable.
* Software The application of a systematic, disciplined, quantifiable approach to   
  Engineering the development, operation, and maintenance of software; that is,   
   the application of engineering to software.” [IEEE90]
* Specification Precise description of an activity or work product which serves as   
   basis or input for further activities or work product. A specification   
   can comprise requirements to a product and how they will be   
   solved. Different parts of a specification (e.g., what is to be done,   
   how it will be done) must not be mixed.[IEEE90]

# 

# Chapter 2 | Test plan and Test procedure

## 2.1 Test Objectives

The objectives of testing Promotion provider are:

1. All failures are discovered
2. All failures are fixed
3. All functions are covered requirements

## 2.2 The scope of testing

Promotion provider will test unit testing. Then, it will record the test result in test record

## 2.3 Test Responsibility

|  |  |
| --- | --- |
| Item | Responsibility |
| Unit test | Tanawat |
| Unit test record | Tanawat |

## 2.4 Test Strategy

Promotion provider test strategy will be follow by:

1. Design test case for each method
2. Prepare test data for each method
3. Determine expected result for each method
4. Perform testing on individual method
5. Record result of testing
6. Store test file in project repository

## 2.5 Result of testing

In the test record the result will separate into two part, which are:

1. Actual output: the actual output those are perform by each test case.
2. Pass/Fail criteria:
   1. Pass: the result of actual result is same like expected result
   2. Fail: the result of actual result is not same like expected result

## 2.6 Test Environment

### 2.6.1 Hardware

1. **Laptop**
   * + MacBook Pro (15-inch, Late 2011)
     + Processor 2.2 GHz Intel Corei7
     + Memory 8 GB 1600 MHz DDR3
     + OS X Yosemite
2. **Mobile**
   * + iPhone 4s

### 2.6.2 Software

* PhpStrom 7.3.1
* Terminal 2.5.3 (343.7)
* Google Chrome 43.0.2357.132

# Chapter 3 | Unit testing

## Web server

**Class:** AccountDAOImpl

**Unit Test Case 1 (UTC01):**

doLogin(email:String, password:String): AccountInfo

**Description:**

This test case is for evaluating the result of procedure when user log in to system. System  
will find matching email and password in database then return matched as AccountInfo object.

**Test Data:**

|  |  |  |
| --- | --- | --- |
| **Type** | AccountInfo | AccountInfo |
| **Id** | 1 | 3 |
| **Email** | [admin@wap.com](mailto:admin@wap.com) | [coffeesmith@wap.com](mailto:coffeesmith@wap.com) |
| **Password** | 81dc9bdb52d04dc20036dbd8313ed055 | 81dc9bdb52d04dc20036dbd8313ed055 |
| **Role** | admin | user |
| **Join\_date** | 2015-06-19 | 2015-07-03 |
| **Status** | 1 | 1 |

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Description | Input | Expected result |
| 1 | Able to log in because matched email and password in database | Email : [admin@wap.com](mailto:admin@wap.com), Password: 1234 | Object(AccountInfo)[accountId: 1, email: [admin@wap.com](mailto:admin@wap.com), password: 81dc9bdb52d04dc20036dbd8313ed055, role: admin, join\_date: 2015-06-19, status: 1] |
| 2 | Unable to log in because password is not matched in database | Email : [admin@wap.com](mailto:admin@wap.com), Password: 1111 | null |
| 3 | Unable to log in because email and password is not exist in database | Email : [test@wap.com](mailto:admin@wap.com), Password: test | null |
| 4 | Able to log in because matched email and password in database | Email : [coffeesmith@wap.com](mailto:coffeesmith@wap.com), Password: 1234 | Object(AccountInfo)[accountId: 1, email: [coffeesmith@wap.com](mailto:coffeesmith@wap.com), password: 81dc9bdb52d04dc20036dbd8313ed055, role: user, join\_date: 2015-07-03, status: 1] |
| 5 | Able to log in because email is not matched in database | Email : [admin](mailto:admin@wap.com), Password: 1234 | null |

**Class:** AccountDAOImpl

**Unit Test Case 2 (UTC02):**

addNewAccount(accountinfo:AccountInfo): Int

**Description:**

This test case is for evaluating the result of procedure when add new account into database then system will return data id from database

**Test Data:**

-

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Description | Input | Expected result |
| 1 | Add account data to data to database | accountId: “”,  Email: warmup@wap.com, Password: 1234, role: user, status: 1 | 10 |
| 2 | Add account data to data to database | accountId: “”,  Email : [hey@wap.com](mailto:admin@wap.com), Password: 1111, role: user, status: 1 | 11 |
| 3 | Add account data to data to database | accountId: “”,  Email : ohman@wap.com, Password: 4321, role: user, status: 1 | 12 |
| 4 | Add account data to data to database | accountId: 90,  Email : [coffeeshop@gmail.com](mailto:coffeeshop@gmail.com), Password: 1234, role: user, status: 1 | 90 |
| 5 | Add account data to data to database | accountId: “”,  Email : nicebar@gmail.com, Password: 4444, role: user, status: 1 | 91 |

**Class:** AccountDAOImpl

**Unit Test Case 3 (UTC03):**

getAccountById(id:Int): AccountInfo

**Description:**

This test case is for evaluating the result of procedure when gather account by using id. System will find account in database where id is equal to inputted id.

**Test Data:**

|  |  |  |
| --- | --- | --- |
| **Type** | AccountInfo | AccountInfo |
| **Id** | 1 | 10 |
| **Email** | nettanawat@gmail.com | [admin@wap.com](mailto:admin@wap.com) |
| **Password** | c09e487ab20c68cdc5c21dce226b0426 | 81dc9bdb52d04dc20036dbd8313ed055 |
| **Role** | admin | user |
| **Join\_date** | 2015-06-19 | 2015-07-03 |
| **Status** | 1 | 1 |

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Description | Input | Expected result |
| 1 | Can gather account data from database because there is account that has id equal to 1 | 1 | Object(AccountInfo)[accountId: 1, email: nettanawat@gmail.com, password: 81dc9bdb52d04dc20036dbd8313ed055, role: admin, join\_date: 2015-06-19, status: 1] |
| 2 | Can gather account data from database because there is account that has id equal to 2 | 10 | Object(AccountInfo)[accountId: 1, email: [admin@wap.com](mailto:admin@wap.com), password: c09e487ab20c68cdc5c21dce226b0426, role: admin, join\_date: 2015-06-19, status: 1] |
| 3 | Can gather account data from database because there is account that has id equal to 3 | 28 | null |
| 4 | Can not gather account data from database because there is no account that has id equal to 100 | 69 | null |
| 5 | Can not gather account data from database because there is no account that has id equal to 101 | 101 | null |