Wireless Sensor-based Application for Managing Theme Park

Test Record

**By**

Pabhawee Chuacharoen 552115037

Department of Software Engineering

College of Arts, Media and Technology

Chiang Mai University

Project Advisor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr.Noppon Choosri

**Document History**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Version | Status | Change Context | Viewable | Reviewer | Responsible | Date |
| V.1.0 | Draft | Introduction  Unit test | PC, NC | PC | PC | 20/08/15 |
| V.1.1 | Draft | System Test  Fix Unit test | PC, NC | PC | PC | 21/08/15 |
| V.2.0 | Release | Change project name | PC, NC | PC, NC | PC | 27/08015 |

PC = Pabhawee Chuacharoen , NC = Dr. Noppon Choosri

**Table of Content**

Contents

[Chapter One | Introduction 3](#_Toc428536235)

[1.1. Object 3](#_Toc428536236)

[1.2. Acronyms and Definitions 3](#_Toc428536237)

[Chapter Two | Unit Test Record 3](#_Toc428536238)

[2.1. Test Attraction Model Record 3](#_Toc428536239)

[Unit Test Case 1 (UTC-01): 3](#_Toc428536240)

[Unit Test Case 2 (UTC-02): 4](#_Toc428536241)

[Unit Test Case 3 (UTC-03): 4](#_Toc428536242)

[Unit Test Case 4 (UTC-04): 5](#_Toc428536243)

[Unit Test Case 5 (UTC-05): 5](#_Toc428536244)

[2.2. Test Guest Model Record 6](#_Toc428536245)

[Unit Test Case 6 (UTC-06): 6](#_Toc428536246)

[Unit Test Case 7 (UTC-07): 6](#_Toc428536247)

[Unit Test Case 8 (UTC-08): 7](#_Toc428536248)

[Unit Test Case 9 (UTC-09): 7](#_Toc428536249)

[Unit Test Case 10 (UTC-10): 8](#_Toc428536250)

[2.3. Test RegisterData Model 8](#_Toc428536251)

[Unit Test Case 12 (UTC-12): 8](#_Toc428536252)

[Unit Test Case 13 (UTC-13): 8](#_Toc428536253)

[Chapter Three | System Test Record 9](#_Toc428536254)

[System Test Case 01 (STC-01): 9](#_Toc428536255)

[System Test Case 02 (STC-02): 9](#_Toc428536256)

[System Test Case 03 (STC-03): 10](#_Toc428536257)

[System Test Case 04 (STC-04): 10](#_Toc428536258)

[System Test Case 05 (STC-05): 10](#_Toc428536259)

[System Test Case 06 (STC-06): 11](#_Toc428536260)

[System Test Case 07 (STC-07): 11](#_Toc428536261)

[System Test Case 08 (STC-08): 11](#_Toc428536262)

[System Test Case 09 (STC-09): 11](#_Toc428536263)

[System Test Case 10 (STC-10): 12](#_Toc428536264)

[System Test Case 11 (STC-11): 13](#_Toc428536265)

[System Test Case 12 (STC-12): 13](#_Toc428536266)

# Chapter One | Introduction

## Object

The Test Record was created for the purposes of recording the actual result of each test case. The test cases have been designed in the Test Plan document and aiming to compare the actual result, the expected results and the test results (Pass or Fail) are detailed in this document. This document records the test results for only in the Progress I of the project.

## Acronyms and Definitions

|  |  |
| --- | --- |
| Name | Explain |
| WSAT | Wireless sensor-based application for managing theme park |
| GIMS | Guest information management system |
| AIMS | Attraction information management system |
| SRS | Software Requirement Specification |
| URS | User Requirement Specification |
| UC | Use Case Description |
| UI | User Interface |
| UID | User Identification |
| RFID | Radio-frequency identification |
| ER | Entity–relationship |
| UTC | Unit Test Case |
| STC | System Test case |

# Chapter Two | Unit Test Record

## Test Attraction Model Record

### Unit Test Case 1 (UTC-01):

public function getAttractions()

* **Test date:**
* **Tested by:**Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| UTC-01-01 | The method should return the all existed Attraction objects in database | array(  array(     'aid' =>'9',      'aname' =>'twister',      'description' =>'beware of the storm',      'height\_limit' =>'120'),  array(     'aid' =>'10',      'aname' =>'magic carpet',      'description' =>'a whole new world',      'height\_limit' =>'0'),  array(     'aid' =>'11',      'aname' =>'coaster',      'description' =>'like a rolling stone',      'height\_limit' =>'120')          ); | array(  array(     'aid' =>'9',      'aname' =>'twister',      'description' =>'beware of the storm',      'height\_limit' =>'120'),  array(     'aid' =>'10',      'aname' =>'magic carpet',      'description' =>'a whole new world',      'height\_limit' =>'0'),  array(     'aid' =>'11',      'aname' =>'coaster',      'description' =>'like a rolling stone',      'height\_limit' =>'120')          ); | N/A |

### Unit Test Case 2 (UTC-02):

public function getAttraction($id)

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| UTC-02-01 | The method should return the right Attraction object (has Id equals to input) with valid id | array(  'aid' =>'9',  'aname' =>'twister',  'description' =>'beware of the storm',  'height\_limit' =>'120'); | array(  'aid' =>'9',  'aname' =>'twister',  'description' =>'beware of the storm',  'height\_limit' =>'120'); | N/A |
| UTC-02-02 | The method should return null with invalid id | null | null | N/A |

### Unit Test Case 3 (UTC-03):

public function insertAttraction($data)

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| UTC-03 | The method should return Attraction object which is same as input | array(  'aname' =>'jet coaster',  'description' =>'not for senior citizen and child',  'height\_limit' =>'140'  ); | array(  'aname' =>'jet coaster',  'description' =>'not for senior citizen and child',  'height\_limit' =>'140'  ); | N/A |

### Unit Test Case 4 (UTC-04):

public function editAttraction($id,$data)

* **Test date:**
* **Tested by:**Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| UTC-04-01 | The method should return Attraction object which is same as input | array(  'aname' =>'jet coaster',  'description' =>'better than the viking',  'height\_limit' =>'140'  ); | array(  'aname' =>'jet coaster',  'description' =>'better than the viking',  'height\_limit' =>'140'  ); | N/A |
| UTC-04-02 | The method should return null when the Id does not exist | Null | Null | N/A |

### Unit Test Case 5 (UTC-05):

public function getGuests()

* **Test date:**
* **Tested by:**Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| UTC-05 | The method should return null when the input does not exist | Null | Null | N/A |

## Test Guest Model Record

### Unit Test Case 6 (UTC-06):

public function getGuests()

* **Test date:**
* **Tested by:**Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| UTC-06 | The method should return the all existed Guest objects in database | array(  array(  'gid' =>'4',  'guid' =>'00ee4535',  'gname' =>'Faifai',  'gmail' =>'fai@mail.com',  'gtel' =>'123456789',  'status' =>'0'),  array(  'gid' =>'5',  'guid' =>'3153f052',  'gname' =>'Biabia',  'gmail' =>'bia@mail.com',  'gtel' =>'76544321',  'status' =>'0')  ); | array(  array(  'gid' =>'4',  'guid' =>'00ee4535',  'gname' =>'Faifai',  'gmail' =>'fai@mail.com',  'gtel' =>'123456789',  'status' =>'0'),  array(  'gid' =>'5',  'guid' =>'3153f052',  'gname' =>'Biabia',  'gmail' =>'bia@mail.com',  'gtel' =>'76544321',  'status' =>'0')  ); | N/A |

### Unit Test Case 7 (UTC-07):

public function getGuest($id)

* **Test date:**
* **Tested by:**Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| UTC-07-01 | The method should return the right Guest object (has Id equals to input) with valid id | array(  'gid' =>'4',  'guid' =>'00ee4535',  'gname' =>'Faifai',  'gmail' =>'fai@mail.com',  'gtel' =>'123456789',  'status' =>'0'); | array(  'gid' =>'4',  'guid' =>'00ee4535',  'gname' =>'Faifai',  'gmail' =>'fai@mail.com',  'gtel' =>'123456789',  'status' =>'0'); | N/A |
| UTC-07-02 | The method should return null with invalid id | null | null | N/A |

### Unit Test Case 8 (UTC-08):

public function insertGuest($data)

* **Test date:**
* **Tested by:**Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| UTC-08 | The method should return Guest object which is same as input | $data = array(  'guid' =>'33ee5f00',  'gname' =>'pakapol',  'gmail' =>'duangdee',  'gtel' =>'0952238845'  ); | $data = array(  'guid' =>'33ee5f00',  'gname' =>'pakapol',  'gmail' =>'duangdee',  'gtel' =>'0952238845'  ); | N/A |

### Unit Test Case 9 (UTC-09):

public function editGuest($id,$data)

* **Test date:**
* **Tested by:**Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| UTC-09-01 | The method should return Guest object which is same as input | array(  'guid' =>'00ee4535',  'gname' =>'FaiFai',  'gmail' =>'duangdee@mail.com',  'gtel' =>'0952238845'  ); | array(  'guid' =>'00ee4535',  'gname' =>'FaiFai',  'gmail' =>'duangdee@mail.com',  'gtel' =>'0952238845'  ); | N/A |
| UTC-09-02 | The method should return null | null | null | N/A |

### Unit Test Case 10 (UTC-10):

public function deleteGuest($id)

* **Test date:**
* **Tested by:**Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| UTC-10 | The method should return null when the input does not exist | Null | Null | N/A |

## Test RegisterData Model

### Unit Test Case 12 (UTC-12):

public function insertUid($uid)

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| UTC-12 | The method should return uid which is same as input | $uid = ‘a3cf8x99’; | $uid = ‘a3cf8x99’; | N/A |

### Unit Test Case 13 (UTC-13):

public function getLastInserted()

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Desctiption | Expected Result | Actual Result | Pass/Fail |
| UTC-12 | The method should return UID which is lasted insert | $uid = ‘a3cf8x99’; | $uid = ‘a3cf8x99’; | N/A |

# Chapter Three | System Test Record

## System Test Case 01 (STC-01):

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| STC-01-01 | Ticket seller can register a guest with correct name, email and mobile no. formats | System shows message ‘Inserted Data Successful’ | System shows message ‘Inserted Data Successful’ | Pass |
| STC-01-02 | Ticket seller can not register a guest with empty name, email and mobile no. fields | System shows error messages ‘The Name field is required’, ‘The Email field is required’ ‘The Mobile No. field is required’’ | System shows error messages ‘The Name field is required’, ‘The Email field is required’ ‘The Mobile No. field is required’’ | Pass |
| STC-01-03 | Ticket seller can not register a guest with a name that has less than 5 character length | System shows error messages ‘’ | System shows error messages ‘’ | Pass |
| STC-01-04 | Ticket seller can not register a guest with invalid email format | System shows error messages ‘The Email field must contain a valid email address.’ | System shows error messages ‘The Email field must contain a valid email address.’ | Pass |

## System Test Case 02 (STC-02):

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| STC-02-01 | Ticket seller can register a guest with correct name, email and mobile no. formats | Ticket seller chooses to view the information of all guests | System shows message ‘Inserted Data Successful’ | Pass |
| STC-02-02 | Ticket seller can not register a guest with empty name, email and mobile no. fields | System shows empty table | System shows empty table | Pass |

s

## System Test Case 03 (STC-03):

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| STC-03-01 | Ticket seller can edit the guest information | System shows the updated guest information :  ID = ‘4’  UID = ‘00ee4535’  Name = ‘Ann Living’  Email = ‘hoocurr[@mail.com](mailto:thongsuk@mail.com)’  Tel. = ‘5524567345’  Status = ‘false’ | System shows the updated guest information :  ID = ‘4’  UID = ‘00ee4535’  Name = ‘Ann Living’  Email = ‘hoocurr[@mail.com](mailto:thongsuk@mail.com)’  Tel. = ‘5524567345’  Status = ‘false’ | Pass |

## System Test Case 04 (STC-04):

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| STC-04-01 | Ticket seller can delete the guest information | System shows all guest information without the deleted guest information | System shows all guest information without the deleted guest information | Pass |

## System Test Case 05 (STC-05):

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| STC-05-01 | User can login as a guest with existing RFID tag and click login within 10 seconds after scanned the tag | System shows guest information | System shows guest information | Pass |
| STC-05-02 | User can not login as a guest with existing RFID tag and click login after 10 seconds after scanned the tag | System shows error message ‘Sorry, try again’ | System shows error message ‘Sorry, try again’ | Pass |
| STC-05-03 | User can not login as a guest without scanning the RFID tag at the entrance gate | System shows error message ‘Your status is not activated’ | System shows error message ‘Your status is not activated’ | Pass |

## System Test Case 06 (STC-06):

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| STC-06-01 | Guest can logout after logged in | System shows login page | System shows login page | Pass |

## System Test Case 07 (STC-07):

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| STC-07-01 | Guest can scan the RFID tag | Guest can login into the system | Guest can login into the system | Pass |

## System Test Case 08 (STC-08):

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| STC-08-01 | Guest can scan the RFID tag | Guest can not login into the system | Guest can not login into the system | Pass |

## System Test Case 09 (STC-09):

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| STC-08-01 | Admin chooses to view the information of all attractions | System shows the information of attractions:  ID=’9’  Name=’twister’  Description =’ beware of the storm’  Height limit=’120’ ID=’10’  Name=’magic carpet’  Description=’a whole new world’  Height limit=’0’  ID= ‘11’  Name=’coaster’ Description=’like a rolling stone’ Height limit=’120’ | System shows the information of attractions:  ID=’9’  Name=’twister’  Description =’ beware of the storm’  Height limit=’120’ ID=’10’  Name=’magic carpet’  Description=’a whole new world’  Height limit=’0’  ID= ‘11’  Name=’coaster’ Description=’like a rolling stone’ Height limit=’120’ | Pass |
| 2 | Admin chooses to view the information when the attraction information is not exist | System shows empty table | System shows empty table | Pass |

## System Test Case 10 (STC-10):

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| STC-10-01 | Admin can add the attraction information with correct format | System shows the added attraction information within the table among others attraction information :  ID=’10’  Name=’magic carpet’  Description=’This is for children’  Height limit=’0’ | System shows the added attraction information within the table among others attraction information :  ID=’10’  Name=’magic carpet’  Description=’This is for children’  Height limit=’0’ | Pass |
| STC-02-02 | Admin can not add the attraction information with Name, Description and Height limit fields empty | System shows error messages ‘The name field is required.’,’The description field is required.’,’The height limit field is required.’ | System shows error messages ‘The name field is required.’,’The description field is required.’,’The height limit field is required.’ | Pass |
| STC-10-03 | Admin can not add the attraction information with name field shorter than 5 characters | System shows error messages ‘The name field is required.’,’The description field is required.’,’The height limit field is required.’ | System shows error messages ‘’ | Pass |
| STC-01-04 | Admin can not add the attraction information with description field shorter than 5 characters | System shows error messages ‘The description field must be at least 5 characters in length.’ | System shows error messages ‘The description field must be at least 5 characters in length.’ | Pass |
| STC-01-05 | Name=’magic carpet’  Description=’This is for children’  Height limit=’zero’ | System shows error messages ‘The height limit field must contain only numbers.’ | System shows error messages ‘The height limit field must contain only numbers.’ | Pass |

## System Test Case 11 (STC-11):

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| STC-11-01 | Admin can edit the attraction information | System shows the updated attraction information :  ID= 11  Name=’coaster’  Description=’who care about the description’  Height limit=’120’ | System shows the updated attraction information :  ID= 11  Name=’coaster’  Description=’who care about the description’  Height limit=’120’ | Pass |

## System Test Case 12 (STC-12):

* **Test date:**
* **Tested by:** Pabhawee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Description | Expected Result | Actual Result | Pass/Fail |
| STC-12-01 | Admin can delete the selected attraction from the system | System shows all attraction information without the deleted attraction information | System shows all attraction information without the deleted attraction information | Pass |