|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| Unit Test  Version 1.0 |
| |  |  |  | | --- | --- | --- | | Panupak Wichaidit | 2/20/14 | SE323 | |

# Revision history

## Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Revision Description** |
| V1.0 | March 1, 2014 | Panupak Wichaidit | Create User unit test documentation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Reviewer

|  |  |  |
| --- | --- | --- |
| **Name** | **Version Approved** | **Date** |
| Maytat Siriangkul | V1.0 | March 1, 2014 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Contents

[Revision history 1](#_Toc381448271)

[Change Record 1](#_Toc381448272)

[Reviewer 1](#_Toc381448273)

References

Class Diagram

Sequence Diagram

Introduction

Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application.

Responsibility

|  |  |
| --- | --- |
| **Task** | **Team member name** |
| Write Unit test plan | Panupak Wichaidit |
| Verify Unit test plan | Narangrit Saisuwan |
| Verify Unit test plan | Maytat Siriangkul |

Test Item

Media E-shopping v1.0

Test tools

J-Unit

Test Environment requirement

Features to be tested

All of methods in a program.

Test case of Unit Test (UTC)

## Class: CustomerDAO

### UTC-01: add Customer (customer: Customer): void

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test status |
| 1 | Test addCustomer(customer: Customer ) : void.Method can return the add customer follow input value. | Cus = new Customer(“panupak”,”Wichaidt”,”home”,”newnok6@gmail.com”,”newnok6”) |  |

## Class: CustomerDAO

### UTC-02: getAllCustomer (): Customer[]

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input value | Test Status |
| 1 | Test getAllCustomer():Customer[]Method can return the result as list of customer |  |  |

## Class: CustomerDAO

### UTC-03: getCustomerByUsername (userName: String): Customer

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test getCustomerByUsername (userName: String): Customer. Method can return the result follow the expect result | Cus01 |  |

## Class: CustomerDAO

### UTC-04: updateCustomer (Customer: customer): Boolean

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test updateCustomer (Customer: customer): Boolean.Method can return the result follow the expect result | Cus1 = new Customer(“panupak2”,”Wichaidt”,”home”,”newnok6@gmail.com”,”newnok6”) |  |

## Class: CustomerService

### UTC-05: register (customer: Customer): boolean

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input value | Test Status |
| 1 | Test register (customer: Customer): Boolean. If method working correctly it will pass. | Cus1 = new Customer(“panupak”,”Wichaidt”,”home”,”newnok6@gmail.com”,”newnok6”) |  |

## Class: CustomerService

### UTC-06: checkCustomer(customer:Customer): boolean

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test checkCustomer(customer:Customer): boolean. If method working correctly it will pass. | Newnok6 |  |

## Class: CustomerService

### UTC-08: editProfile(customer:Customer): boolean

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test editProfile(username:String): boolean. . If method working correctly it will pass. | “newnok6” |  |

## Class: CustomerService

### UTC-09: checkNameFormat(name:String): boolean

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test checkNameFormat(name:String): boolean. If method working correctly it will pass. | “Panupak” |  |

## Class: CustomerService

### UTC-10: checkUsernameFormat(username:String): boolean

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test checkFormUsername(username:String): boolean. If method working correctly it will pass. | “newnok6” |  |

## Class: CustomerService

### UTC-11: checkPasswordFormat(password:String): boolean

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test checkPasswordFormat(password:String): boolean. If method working correctly it will pass. | “1234” |  |

## Class: CustomerService

### UTC-12: checkPictureFormat (Image:Image): boolean

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test checkFormUsername(username:String): boolean. If method working correctly it will pass. | “cus1.jpg” |  |

## Class: AdminDAO

### UTC-14: getAdminFromID (id:String): Admin

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test getAdminFromID (id:String): Admin. If method working correctly it will pass. | “001” |  |

## Class: LoginService

### UTC-15: loginByAdmin (username:String, password:String): boolean

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test : loginByAdmin (username:String, password:String): boolean. If method working correctly it will pass. | Username :”admin01”  Password : “1234” |  |

## Class: LoginService

### UTC-16: loginByCustomer (username:String, password:String): boolean

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test : loginByCustomer (username:String, password:String): booleanIf method working correctly it will pass. | Username :”cus01”  Password : “1234” |  |

## Class: LoginService

### UTC-17: isStatus (): boolean

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test : isStatus (): boolean .If method working correctly it will pass. |  |  |

## Class: LoginService

### UTC-18: setStatus (): boolean

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test : setStatus (): boolean.If method working correctly it will pass. |  |  |

## Class: LoginService

### UTC-19: setCurrentUser (username:String): void.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test : : setCurrentUser (username:String): void.If method working correctly it will pass. | Cus01 |  |

## Class: LoginService

### UTC-20: getCurrentUser (): String.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test : : getCurrentUser (): String.If method working correctly it will pass. |  |  |

## Class: MediaDAO

### UTC-21: addMedia (media:Media): Boolean.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test : : addMedia (media:Media): Boolean.If method working correctly it will pass. | Media = new Media(“Hobbit an Unexpected Journey”,” Blu-ray 3D”,” Fantasy”,” 20”,” 1500.00”,” media\_03.jpg”,” In-stock”,” 08/04/2013”) |  |

## Class: MediaDAO

### UTC-22: getMediaById (int : id): Media.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test getMediaById (int : id): Media. If method working correctly it will pass. | 01 |  |

## Class: MediaDAO

### UTC-23: getAllMedia (): Media[].

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test : getAllMedia (): Media[].If method working correctly it will pass. |  |  |

## Class: MediaDAO

### UTC-24: getMediaByKey (name : String): Media[].

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test getMediaByName (): Media[].If method working correctly it will pass. | “Frozen” |  |

## Class: MediaService

### UTC-25: checkTitleFormat(title : String): Boolean.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test checkTitleFormat(title : String): Boolean. If method working correctly it will pass. | “Frozen” |  |

## Class: MediaService

### UTC-26: checkImageFormat(image: Image): Boolean.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test checkImageFormat(image: Image): Boolean.If method working correctly it will pass. | “media01.jpg” |  |

## Class: MediaService

### UTC-27: addNewMedia(media: Media): Boolean.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | description | Input value | Test Status |
| 1 | Test addNewMedia(media: Media): Boolean. If method working correctly it will pass. | Media = new Media(“Hobbit an Unexpected Journey”,” Blu-ray 3D”,” Fantasy”,” 20”,” 1500.00”,” media\_03.jpg”,” In-stock”,” 08/04/2013”) |  |

## Class: CartDAO

### UTC-28: getAllOrder(): Order[].

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input value | Test Status |
| 1 | Test getAllOrder(): Order[]. If method working correctly it will pass. |  |  |

## Class: CartDAO

### UTC-29: getOrderByCustomer(customer:Customer):Order[]

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input value | Test Status |
| 1 | Test getOrderByCustomer(customer:Customer):Order[]. If method working correctly it will pass. |  |  |

## Class: CartDAO

### UTC-30: getOrderByID(int :id): Order[].

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input value | Test Status |
| 1 | Test getOrderByID(int :id): Order[]. If method working correctly it will pass. | 1 |  |

## Class: CartDAO

### UTC-31: getOrderDetail(order :Order): OrderDetail.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input value | Test Status |
| 1 | Test getOrderDetail(order :Order): OrderDetail. If method working correctly it will pass. | Order = new Order(cus1,”08/04/2013”,”08/04/2013”,3000,false,orderDetailList) |  |

## Class: CartDAO

### UTC-32: getOrderByPaymentDate(paymentDate :PaymentDate): Order[].

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input value | Test Status |
| 1 | Test getOrderByID(int :id): Order[]. If method working correctly it will pass. | “08/04/2013” |  |

## Class: CartService

### UTC-33: setCurrentOrder(order :Order): Boolean.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input value | Test Status |
| 1 | Test setCurrentOrder(order :Order): Boolean. If method working correctly it will pass. | Order = new Order(cus1,”08/04/2013”,”08/04/2013”,3000,false,orderDetailList) |  |

## Class: CartService

### UTC-33: checkOutOrder(order :Order): Boolean.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input value | Test Status |
| 1 | Test : checkOutOrder(order :Order): Boolean. If method working correctly it will pass. | Order = new Order(cus1,”08/04/2013”,”08/04/2013”,3000,false,orderDetailList) |  |

## Class: CartService

### UTC-33: showOrderDetail(order :Order): OrderDetail.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input value | Test Status |
| 1 | Test showOrderDetail(order :Order): OrderDetail. If method working correctly it will pass. | Order = new Order(cus1,”08/04/2013”,”08/04/2013”,3000,false,orderDetailList) |  |

## Class: CartService

### UTC-33: showAllOrder():Order[]

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input value | Test Status |
| 1 | Test showAllOrder():Order[]. If method working correctly it will pass. |  |  |