**Project Report**

**Training Center Website**

**Mai Mohamed Maryam Mohamed**

**Mayar Ahmed Rania Gamal**

# Software Under Test:

Our software project is a website for a training center which provides educational courses for its customers. It contains two main modules: the admin functionalities and the customer functionalities.

The SUT is chosen to be the customer’s functionalities which are:

Course Registration   
Search for courses   
Download Course contents   
Give feedback   
Rate courses  
viewing course details

# Coverage Criteria:

## White box testing:

Logic Coverage (CACC)

Branch Coverage

## Black box testing:

Logic Coverage

Input space partitioning (Base Choice Coverage)

# Registration Function testing

White Box Logic Coverage:

*Predicates:*

* P1: this->validate (), always active
* P2: ($regist!=null && $regist->ssn!=$req->ssn), reachability: validate is true
* P3: ($regist===null) reachability: p2 is true
* P4: ($course\_reg==null) reachbility: p2 is false

Black Box Logic Coverage:

*Specs:*

All fields are required, name doesn’t contain numbers, email of valid format, national id consists of numbers only , national id is 14 digits , phone number consists of numbers only, phone number is 11 digits, name is less than 100 characters, course id is a positive integer

*Predicate:*

Predicate: !empty(name) ^ !empty (email) ^ !empty( ssn) ^ !empty(address) ^ !empty(phone number) ^ !numeric(name) ^ valid\_format(email) ^ len(ssn) =14 ^ len(name)<100 ^len(phone number)=11 ^ numeric(ssn) ^ numeric(phone number) ^ integer(course id) ^ greater\_than\_zero(course id)

Black Box input space partitioning: Base Choice

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic** | **Blocks** | | |
| 1. Name is empty | True | **False** |  |
| 1. Name is a number | True | **False** |  |
| 1. Name size | **<=100** | >100 |  |
| 1. Address is empty | True | **False** |  |
| 1. Phone number is empty | True | **False** |  |
| 1. Phone number size | <11 | **=11** | >11 |
| 1. Phone number is an integer | **True** | False |  |
| 1. Email is empty | True | **False** |  |
| 1. Email format | **valid** | Not valid |  |
| 1. Ssn is empty | True | **False** |  |
| 1. Ssn is an integer | **True** | False |  |
| 1. Ssn size | <14 | **=14** | >14 |
| 1. Course id is an integer | **True** | False |  |
| 1. Course id relation to zero | <=0 | **>0** |  |

## Test Requirements

|  |  |  |
| --- | --- | --- |
| **Coverage Criteria** | **Test Requirements** | **Test Cases** |
| **White Box Testing** | | |
| Branch Coverage | New registrant | T1 |
| Another registrant has the same email | T2 |
| Registered in the same course before | T3 |
| Ssn larger than 14 numbers (invalid input) | T4 |
| Logic Coverage | P1=True | T1 |
| P1=False | T4 |
| P2: [T T] | T2 |
| P2: [F T] | T1 |
| P2: [T F] | T3 |
| P3: True | T1 |
| P3: False | T3 |
| P4: True | T1 |
| P4: False | T3 |
| **Black Box Testing** | | |
| Logic Coverage | [T T T T T T T T T T T T T T] | T1 |
| [T F T T T T T T T T T T T T] | T6 |
| [T T F T T T T T T T T T T T] | T7 |
| [T T T F T T T T T T T T T T] | T8 |
| [T T T T F T T T T T T T T T] | T9 |
| [T T T T T F T T T T T T T T] | T10 |
| [T T T T T T F T T T T T T T] | T11 |
| [T T T T T T T F T T T T T T] | T12 |
| [T T T T T T T T F T T T T T] | T13 |
| [T T T T T T T T T F T T T T] | T14 |
| [T T T T T T T T T T F T T T] | T15 |
| [T T T T T T T T T T T F T T] | T16 |
| [T T T T T T T T T T T T F T] | T17 |
| [T T T T T T T T T T T T T F] | T18 |

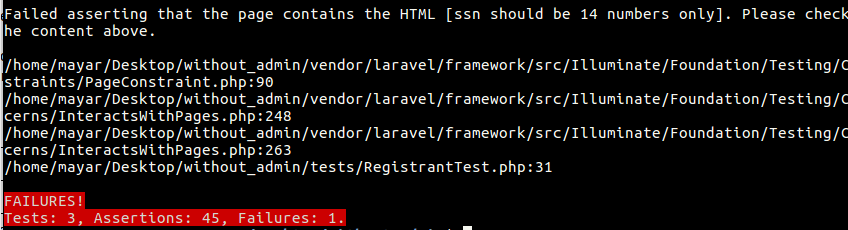
Base Choice Coverage

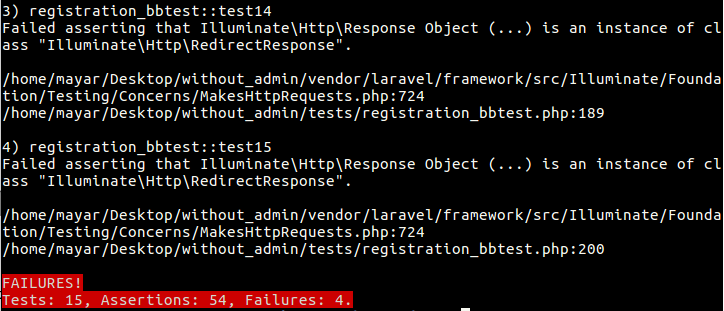
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Requirements** | | | | | | | | | | | | | | **Test Cases** |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N |  |
| F | F | <=100 | F | F | =11 | T | F | valid | F | T | =14 | T | >0 | T1 |
| F | F | <=100 | F | F | =11 | T | F | valid | F | T | =14 | T | **<=0** | T18 |
| F | F | <=100 | F | F | =11 | T | F | valid | F | T | =14 | **F** | >0 | **T17** |
| F | F | <=100 | F | F | =11 | T | F | valid | F | T | **<14** | T | >0 | T11 |
| F | F | <=100 | F | F | =11 | T | F | valid | F | T | **>14** | T | >0 | T19 |
| F | F | <=100 | F | F | =11 | T | F | valid | F | **F** | =14 | T | >0 | T12 |
| F | F | <=100 | F | F | =11 | T | F | valid | **T** | T | =14 | T | >0 | T7 |
| F | F | <=100 | F | F | =11 | T | F | **invalid** | F | T | =14 | T | >0 | T14 |
| F | F | <=100 | F | F | =11 | T | **T** | valid | F | T | =14 | T | >0 | T6 |
| F | F | <=100 | F | F | =11 | **F** | F | valid | F | T | =14 | T | >0 | T13 |
| F | F | <=100 | F | F | **<11** | T | F | valid | F | T | =14 | T | >0 | T20 |
| F | F | <=100 | F | F | **>11** | T | F | valid | F | T | =14 | T | >0 | T16 |
| F | F | <=100 | F | **T** | =11 | T | F | valid | F | T | =14 | T | >0 | T9 |
| F | F | <=100 | **T** | F | =11 | T | F | valid | F | T | =14 | T | >0 | T8 |
| F | F | **>100** | F | F | =11 | T | F | valid | F | T | =14 | T | >0 | T15 |
| F | **T** | <=100 | F | F | =11 | T | F | valid | F | T | =14 | T | >0 | T10 |
| **T** | F | <=100 | F | F | =11 | T | F | valid | F | T | =14 | T | >0 | T5 |

## Test Cases:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected output** | **Result** |
| T1 | Name= “mohamed” email= 'mohamed@gmail.com'  Ssn=12312512312311  Address=’haram’ phone\_number=01065518447 | Registration successful | Passed |
| T2 | Name=’mohamed’  Email=’salma@gmail.com’  Ssn=12312382312511  Address=’haram’  Phone\_number=01063518447 | This email belongs to another ssn | Passed |
| T3 | Name=’salma’  Email=’salma@gmail.com’  Ssn=12312312312311  Address=haram  Phone\_number=01065518447 | You are already registered to this course | Passed |
| T4 | Name=1234  Email=mohamed@gmail.com’  Ssn= 1231231231231199  Address=haram  Phone\_number=mm | Ssn should be 14 numbers only  Name shouldn’t be numeric  Phone number must be a number | Failed |
| T5 | Course id=3  Name=’’  Email=’ [omar23@gmail.com](mailto:omar23@gmail.com)’  Ssn= 1231231289231199  Address==’haram’  Phone\_number=01282340938 | The name field is required | Passed |
| T6 | Course id=3  Name=’omar’  Email=’’  Ssn= 1231231289231199  Address==’haram’  Phone\_number=01282340938 | The email field is required | Passed |
| T7 | Course id=3  Name=’omar’  Email=’ [mayar@gmail.com](mailto:mayar@gmail.com)’  Ssn=  Address==’haram’  Phone\_number=01282340938 | The ssn field is required | Passed |
| T8 | Course id=3  Name=’omar’  Email=’ [mayar@gmail.com](mailto:mayar@gmail.com)’  Ssn= 1231231289231199  Address==’haram’  Phone\_number=01282340938 | The Address field is required | Passed |
| T9 | Course id=3  Name=’omar’  Email=’ [mayar@gmail.com](mailto:mayar@gmail.com)’  Ssn= 1231231289231199  Address==’haram’  Phone\_number= | The phone number field is required | Passed |
| T10 | Course id=3  Name= ‘1234dd’  Email=’ [omar@gmail.com](mailto:omar@gmail.com)’  Ssn= 1231231289231199  Address==’haram’  Phone\_number= 01282340938 | The name format is invalid | Passed |
| T11 | Course id=4  Name= ‘omar  Email=’ [omarahmed@gmail.com](mailto:omarahmed@gmail.com)’  Ssn= 12923  Address==’haram’  Phone\_number= 01282340938 | The ssn must be 14 characters | Failed |
| T12 | Course id=3  Name= ‘omar  Email=’ omarahmedmahmoud@gmail.com’  Ssn= ‘bla bla’  Address==’haram’  Phone\_number= 01282340938 | The ssn must be a number | Passed |
| T13 | Course id=3  Name= ‘omar  Email=’ omarahmedma@gmail.com’  Ssn= ‘012823409383’  Address== ‘address’  Phone\_number= 01ma | The phone number must be a number | Passed |
| T14 | Course id=3  Name= ‘omar  Email=’ omarahmdmahgmailm’  Ssn= ‘012823409383’  Address== 01ma  Phone\_number= 01282340938 | The email must be a valid email address | Passed |
| T15 | Course id=3  Name= ‘mayar ahmed mahmoud hefny mostafa hassan kareem mayar nahla ahmed maya maya kareem ahmed mahmoud hefny msostafa’  Email=’ [omar@gmail.com](mailto:omar@gmail.com)’  Ssn= 1231231289231199  Address==’haram’  Phone\_number= 01282340938 | The name may not be greater than 100 characters. | Passed |
| T16 | Course id=3  Name= ‘omar  Email=’ omarahmedmahmoud@gmail.com’  Ssn= 01282340938883  Address==’haram’  Phone\_number= 0128234093888553 | The phone number must be 11 characters | Failed |
| T17 | Course id = c  Name=’omar ahmed’  Email=’ [omar@gmail.com](mailto:omar@gmail.com)’  Ssn= 12312312312311  Address==’haram’  Phone\_number= 01066518447 | Redirect back without registration | Failed |
| T18 | Course id = -1  Name=’omar ahmed’  Email=’ [omar@gmail.com](mailto:omar@gmail.com)’  Ssn= 12312312312311  Address==’haram’  Phone\_number= 01066518447 | Redirect back without registration | Failed |
| T19 | Course id =4  Name=’omar’  Email=’ [omarahmed@gmail.com](mailto:omarahmed@gmail.com)’  Ssn= 1292323451234123  Address==’haram’  Phone\_number=01282340938 | The ssn must be 14 characters | Failed |
| T20 | Course id =3  Name=’omar’  Email=’ omarahmedmahmoud@gmail.com’  Ssn= 01282340938883  Address==’haram’  Phone\_number= 01282343 | The phone number must be 11 characters | Failed |

Execution Results: **7 Failures**







# Add Review Function Testing

White Box Logic Coverage Criteria:

*Predicates:*

* P1: this->validate ()
* P2: (! empty ($course)), reachability: p1=true
* P3: (Hash:check($req->code,$reg->code)) reachability: p2=true
* P4: (!$reg->confirmed) reachability: p3 true

Black Box Logic Coverage:

*Specs:*

review field is required and code field is required

*Predicate:*

!empty(review) ^ !empty(code) ^ integer (course\_id) ^ positive (course\_id)

Black Box input space partitioning: Base Choice

|  |  |  |
| --- | --- | --- |
| **Characteristics** | **Blocks** | |
| A-review is empty | True | **False** |
| B-code is empty | True | **False** |
| C-Course id is an integer | **True** | False |
| D-Course id relation to zero | <=0 | **>0** |

## Test Requirements

|  |  |  |
| --- | --- | --- |
| **Coverage Criteria** | **Test Requirements** | **Test Cases** |
| **White Box Testing** | | |
| Branch Coverage | Invalid course id | T1 |
| Empty review | T2 |
| Registrant confirmed | T3 |
| Registrant not confirmed | T4 |
| Not registered in this course | T5 |
| Logic Coverage | P1: True | T3 |
| P1: False | T2 |
| P2: True | T3 |
| P2: False | T1 |
| P3: True | T3 |
| P3: False | T5 |
| P4: True | T4 |
| P4: False | T3 |
| **Black Box Testing** | | |
| Logic Coverage | [T T T T] | T3 |
|  | [F T T T] | T2 |
|  | [T F T T] | T6 |
|  | [T T F T] | T7 |
|  | [T T T F] | T1 |

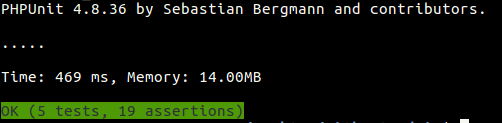
Input Space Partitioning: Base Choice Coverage

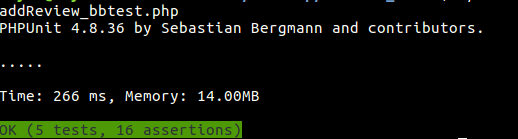
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Requirements** | | | | **Test Cases** |
| **A** | **B** | **C** | **D** |  |
| F | F | T | >0 | T3 |
| **T** | F | T | >0 | T2 |
| F | **T** | T | >0 | T6 |
| F | F | **F** | >0 | T7 |
| F | F | T | **<=0** | T1 |

## Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected output** | **Results** |
| T1 | Course Id=-1  Code=222  Review= ‘my review’ | Redirect back to home page | Passed |
| T2 | Course Id=2  Code=12345  Review=’’ | The review field is required | Passed |
| T3 | Course Id=2  Code= e39d565  Review=’ this course was great’ | Review Added successfully | Passed |
| T4 | Course Id=2  Code= 06e47c6  Review=’ this course was great’ | your registration isn't confirmed yet to add a review | Passed |
| T5 | Course Id=2  Code= 12345  Review=’ this course was great’ | you aren't registered in this course | Passed |
| T6 | Course id= 2  Code=’’  Review=’ this course was great’ | The code field is required | Passed |
| T7 | Course id= c  Code=222  Review=’ this course was great’ | Redirect back without adding review | Passed |

Execution Results: **No Failures**



****

# Add Rating Function Testing

White Box Logic Coverage Criteria:

*Predicates:*

* P1: this->validate ()
* P2: (! empty ($course)), reachability: p1=true
* P3: (Hash:check($req->code,$reg->code)) reachability: p2=true
* P4: (!$reg->confirmed) reachability: p3 true
* P5: ($user\_ratings !=0) reachbility: confirmed=true

Black Box Logic Coverage Criteria:

*Specs:* rating field is required, code field is required, rating must be a number

*Predicate:*

!empty(rating) ^ !empty(code) ^ numeric (rating) ^ integer (course\_id) ^ positive (course\_id)

Input Space Partitioning: Base Choice Coverage

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristics** | **Blocks** | | |
| A-rating is empty | True | **False** |  |
| B-code is empty | True | **False** |  |
| C-Course id is an integer | **True** | False |  |
| D-Course id relation to zero | <=0 | **>0** |  |
| E-rating is a number | **True** | False |  |
| F-rating relation to 0 | <0 | **>=0 and <=5** | >5 |

## Test Requirements

|  |  |  |
| --- | --- | --- |
| **Coverage Criteria** | **Test Requirements** | **Test Cases** |
| **White Box Testing** | | |
| Branch Coverage | Invalid course id | T1 |
| Empty rating | T2 |
| Registrant confirmed and hasn’t rated before | T3 |
| Registrant confirmed but already rated | T4 |
| Registrant not confirmed | T5 |
| Not registered in this course | T6 |
| Logic Coverage | P1: True | T3 |
| P1: False | T2 |
| P2: True | T3 |
| P2: False | T1 |
| P3: True | T3 |
| P3: False | T6 |
| P4: True | T5 |
| P4: False | T3 |
| P5: True | T3 |
| P5: False | T4 |
| **Black Box Testing** | | |
| Logic Coverage | [T T T T T] | T3 |
| [F T T T T] | T2 |
| [T F T T T] | T7 |
| [T T F T T] | T8 |
| [T T T F T] | T9 |
| [T T T T F] | T1 |

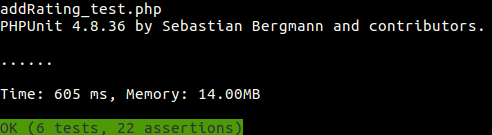
Input Space Partitioning: Base Choice Coverage

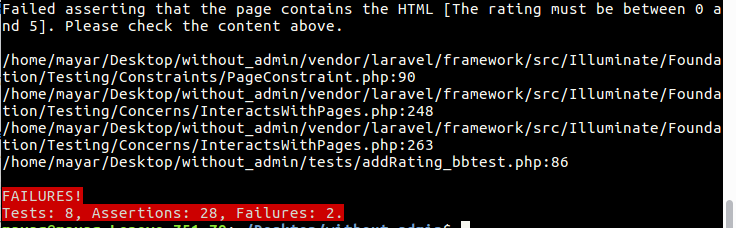
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Requirements** | | | | | | **Test Cases** |
| **A** | **B** | **c** | **D** | **E** | **F** |  |
| F | F | T | >0 | T | >=0 and <=5 | T3 |
| **T** | F | T | >0 | T | >=0 and <=5 | T2 |
| F | **T** | T | >0 | T | >=0 and <=5 | T7 |
| F | F | **F** | >0 | T | >=0 and <=5 | T9 |
| F | F | T | **<=0** | T | >=0 and <=5 | T1 |
| F | F | T | >0 | **F** | >=0 and <=5 | T8 |
| F | F | T | >0 | T | **<0** | T10 |
| F | F | T | >0 | T | **>5** | T11 |

## Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected output** | **Result** |
| T1 | Course Id=-1  Code=’222’  Rating=’12’ | Redirect back to home page | Passed |
| T2 | Course id= 2  Code=’222’  Rating=’’ | The rating field is required | Passed |
| T3 | Course id= 2  Code=’cda5b5c’  Rating=2.5 | Rating Added successfully | Passed |
| T4 | Course id =2  Code= e39d565  Rating=3.5 | your already rated this course | Passed |
| T5 | Course id= 2  Code= 06e47c6  Rating=2 | your registration isn't confirmed yet to rate | Passed |
| T6 | Course id =2  Code=12345  Rating=4 | you aren't registered in this course | Passed |
| T7 | Course id =2  Code= ‘’  Rating=2.5 | The code field is required | Passed |
| T8 | Course id =2  Code= cda5b5c  Rating=’mm’ | The rating must be a number | Passed |
| T9 | Course id =c  Code= 222  Rating=3 | Redirect back without adding review | Passed |
| T10 | Course id =2  Code= cda5b5c  Rating=-1 | Rating must be between 0 and 5 | Failed |
| T11 | Course id =2  Code= cda5b5c  Rating=12 | Rating must be between 0 and 5 | Failed |

Execution Results: **2 failures**





# Search Function Testing:

Black Box Logic Coverage Criteria:

*Specs:*

Search field is required

*Predicate:*

!empty(search field)

Input Space Partitioning: Base Choice Coverage

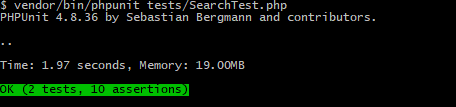
|  |  |  |
| --- | --- | --- |
| **Characteristic** | **Blocks** | |
| 1. Empty Search Field | True | False |

## Test Requirements

|  |  |  |
| --- | --- | --- |
| **Coverage Criteria** | **Test Requirements** | **Test Cases** |
| **White Box Testing** | | |
| Branch Coverage | Valid request | T1 |
| Invalid request | T2 |
| Logic Coverage | infeasible | Infeasible |
| **Black Box Testing** | | |
| Logic Coverage | P = [True] | T1 |
|  | P = [False] | T2 |
| Base Choice Coverage | [True] | T2 |
|  | [False] | T1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected Output** | **Results** |
| 1 | Search field = “Management” | Management Course Page | Passed |
| 2 | Search field = “” | “The name field is required” | Passed |

Execution Results: **No Failures**

**

# Course Description Function Testing:

Black Box Logic Coverage Criteria:

*Specs:*

Course id: non-empty, integer, number, greater than zero.

*Predicate:*

!empty(Course id) ^ integer(Course id) ^ greater than zero(Course id).

Input Space Partitioning:

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic** | **Blocks** | | |
| 1. Numerical course id | **True** | False |  |
| 1. Integer course id | **True** | False |  |
| 1. Empty course id | True | **False** |  |
| 1. Course id compared to zero | <0 | =0 | **>0** |

## Test Requirements

|  |  |  |
| --- | --- | --- |
| **Coverage Criteria** | **Test Requirements** | **Test Cases** |
| Branch Coverage | Cover the only one branch | T1 |
| Logic Coverage | Infeasible | Infeasible |
| Black box: Logic coverage | [T,T,T] | T1 |
| [F,T,T] | Infeasible |
| [T,F,T] | T3 |
| [T,T,F] | T5 |

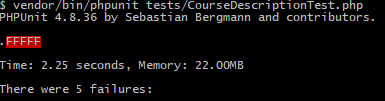
Base Choice Coverage:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Requirements** | | | | **Test Cases** |
| **A** | **B** | **C** | **D** |  |
| T | T | F | >0 | T1 |
| F | T | F | >0 | T2 |
| T | F | F | >0 | T3 |
| T | T | T | >0 | T4 |
| T | T | F | <0 | T5 |
| T | T | F | =0 | T6 |

## Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case** | **Input** | **Expected Output** | **Results** |
| T1 | Course ID = 2 | Management Course Description Page | Passed |
| T2 | Course ID = g | Redirect back to home page | Failed |
| T3 | Course ID = 0.5 | Redirect back to home page | Failed |
| T4 | Course ID = “” | Redirect back to home page | Failed |
| T5 | Course ID = -1 | Redirect back to home page | Failed |
| T6 | Course ID = 0 | Redirect back to home page | Failed |

Execution Results: **5 Failures**





# Download Function Testing:

White Box Logic Coverage Criteria:

*Predicates:*

* P1: valid(request)
* P2: ! empty(Course)
* P3: Confirmed(Registration)
* P4: Hash::check($req->reg\_code,$reg->code)

*Note: Each predicate has only on clause which is always active.*

Black Box Logic Coverage:

*Specs:*

Registration code: non-empty string, Course id: non-empty, integer, number, greater than zero.

*Predicate:*

! empty (Reg code) ^! Empty (course id) ^ Number (course id) ^ Integer (course id) ^ course id>0

Input Space Partitioning:

|  |  |  |  |
| --- | --- | --- | --- |
| Characteristic | Blocks | | |
| 1. Empty Reg Code | True | **False** |  |
| 1. Numerical course id | **True** | False |  |
| 1. Integer course id | **True** | False |  |
| 1. Empty course id | True | **False** |  |
| 1. Course id compared to zero | <0 | =0 | **>0** |

## Test Requirements:

|  |  |  |
| --- | --- | --- |
| **Coverage Criteria** | **Test Requirements** | **Test Cases** |
| Branch Coverage | Invalid Registration Code | T2 |
|  | Course id doesn’t exist | T3 |
|  | Other registrations exist for the course (More than one) | T6 |
|  | No Registrations for the course | T5 |
|  | Registration hasn’t been confirmed | T4 |
|  | Reg code isn’t registered | T6 |
|  | Registration is confirmed | T1 |
| Logic Coverage | P1: True | T1 |
|  | P1: False | T2 |
|  | P2: True | T1 |
|  | P2: False | T3 |
|  | P3: True | T1 |
|  | P3: False | T4 |
|  | P4: True | T1 |
|  | P4: False | T6 |
| Black box Logic Coverage | [T T T T T] | T1 |
|  | [F T T T T] | T2 |
|  | [T F T T T] | T9 |
|  | [T T F T T] | T7 |
|  | [T T T F T] | T8 |
|  | [T T T T F] | T10 |

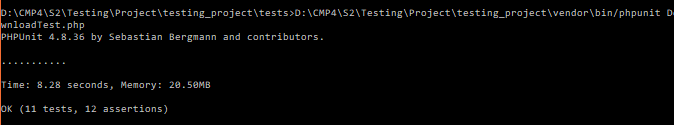
Base Choice Coverage:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Requirements** | | | | | **Test Cases** |
| **A** | **B** | **C** | **D** | **E** |  |
| F | T | T | F | >0 | T1 |
| **T** | T | T | F | >0 | T2 |
| F | **F** | T | F | >0 | T7 |
| F | T | **F** | F | >0 | T8 |
| F | T | T | **T** | >0 | T9 |
| F | T | T | F | **=0** | T11 |
| F | T | T | F | **<0** | T10 |

## Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected Output** | **Results** |
| T1  Registration is confirmed  (Success Scenario) | Course id: 2  Reg code: 0170faf  Registrant => Confirmed = 1 | File is downloaded  Session flash: download successful | Passed |
| T2 | Course id: 2  Reg code: “” | Redirect back  File isn’t downloaded | Passed |
| T3 | Course id: 123  Reg code: 123 | Redirect back  File isn’t downloaded | Passed |
| T4 | Course id: 2  Reg code: 44ad48e  Registrant => Confirmed = 0 | Redirect back  File isn’t downloaded | Passed |
| T5 | Course id: 4  Reg code: 123 | Redirect back  File isn’t downloaded | Passed |
| T6 | Course id: 3  Reg code: 123 | Redirect back  File isn’t downloaded | Passed |
| T7 | Course ID = g  Reg code: 123 | Redirect back  File isn’t downloaded | Passed |
| T8 | Course ID = 0.5  Reg code: 123 | Redirect back  File isn’t downloaded | Passed |
| T9 | Course ID = “”  Reg code: 123 | Redirect back  File isn’t downloaded | Passed |
| T10 | Course ID = -1  Reg code: 123 | Redirect back  File isn’t downloaded | Passed |
| T11 | Course ID = 0  Reg code: 123 | Redirect back  File isn’t downloaded | Passed |

Execution Results: **No Failures**



# Feedback Function Testing:

Black Box Logic Coverage Criteria:

*Specs:*

Name: non-empty string without numbers, email: non-empty & follows the email format, Message: non-empty string.

*Predicate:* ! empty (name) ^ valid(name)^ !empty(email) ^ valid(email)^!empty(message)

Input Space Partitioning:

|  |  |  |
| --- | --- | --- |
| **Characteristic** | **Blocks** | |
| 1. Empty name | T | **F** |
| 1. Name containing numbers | T | **F** |
| 1. Empty email | T | **F** |
| 1. Valid email format | **T** | F |
| 1. Empty message | T | **F** |

## Test Requirements

|  |  |  |
| --- | --- | --- |
| **Coverage Criteria** | **Test Requirements** | **Test Cases** |
| **White Box Testing** | | |
| Branch Coverage | Valid inputs | T1 |
|  | Invalid inputs (e.g. email) | T2 |
| Logic coverage | infeasible | Infeasible |
| **Black Box Testing** | | |
| Logic Coverage | [T T T T T] | T1 |
|  | [F T T T T] | T3 |
|  | [T F T T T] | T4 |
|  | [T T F T T] | T5 |
|  | [T T T F T] | T2 |
|  | [T T T T F] | T6 |

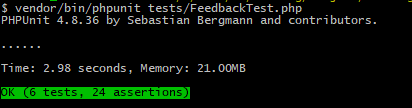
Base Choice Coverage

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Requirements** | | | | | **Test Cases** |
| **A** | **B** | **C** | **D** | **E** |  |
| F | F | F | T | F | T1 |
| **T** | F | F | T | F | T3 |
| F | **T** | F | T | F | T4 |
| F | F | **T** | T | F | T5 |
| F | F | F | **F** | F | T2 |
| F | F | F | T | **T** | T6 |

## Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected Output** | **Results** |
| T1  Valid Request | Name= Ahmed  Email= ahmed@yahoo.com  Message= This is a message. | “Success! Message submited.”  Msg appears | Passed |
| T2  Invalid Request | Name= Ahmed  Email= ahmed@com  Message= This is a message. | The email must be a valid email address. | Passed |
| T3 | Name= “”  Email= ahmed@yahoo.com  Message= This is a message. | The name field is required. | Passed |
| T4 | Name= Ahmed23  Email= ahmed@yahoo.com  Message= This is a message. | The name format is invalid. | Passed |
| T5 | Name= Ahmed  Email= “”  Message= This is a message. | The email field is required. | Passed |
| T6 | Name= Ahmed  Email= ahmed@yahoo.com  Message=”” | The message field is required. | Passed |

Execution Results: **No Failures**



# Workload

|  |  |
| --- | --- |
| Team Member | Activities |
| Mai Mohamed [[Mai.mawadalla@gmail.com](mailto:Mai.mawadalla@gmail.com)] | 1. Download function testing 2. Search function testing |
| MaryamMohamed[[maryam.m.abdelmoneam@gmail.com](mailto:maryam.m.abdelmoneam@gmail.com)] | 1. Add review function testing 2. Add rating function testing |
| Mayar Ahmed[[mayar.ahefny@gmail.com](mailto:mayar.ahefny@gmail.com)] | Registration function testing |
| Rania Gamal [[raniagml12@gmail.com](mailto:raniagml12@gmail.com)] | 1. Course details view function testing 2. Feedback function testing |

# CD Contents:

|  |  |
| --- | --- |
| Item | Containing Folder |
| Source Code | testing\_project\app\Http\Controllers |
| Testing scripts | testing\_project\tests |