**SWEN90006**

**SOFTWARE TESTING AND RELIABILITY**

**ASSIGNMENT1**

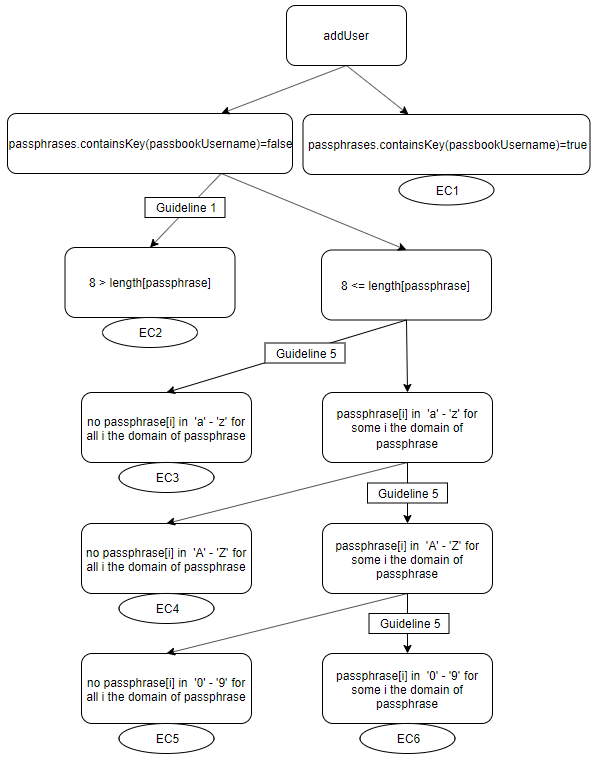
**TASK1**

**Test Template Tree**

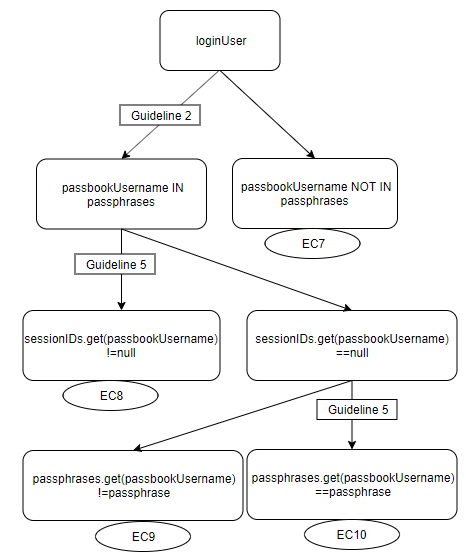
SWEN90006-Assignment 1

Test Template Tree for Task1

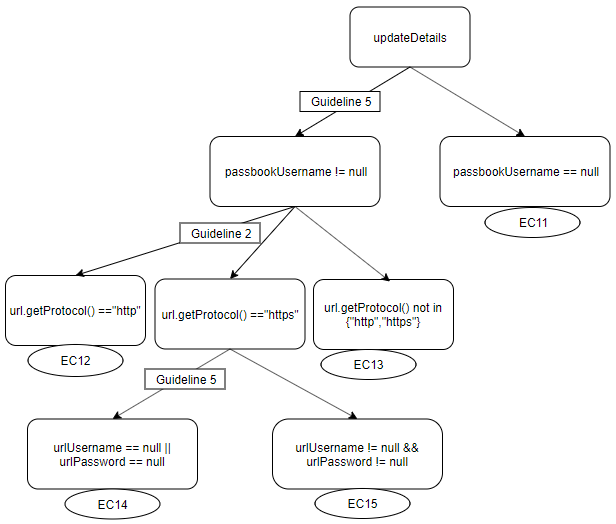
1. addUser()



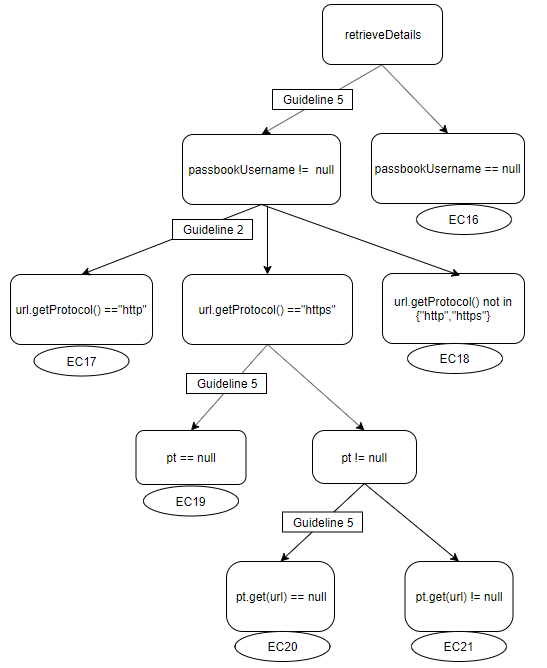
2. loginUser()



3. updateUser()



4. retrieveDetails ()



**Question**

Do your set of equivalence classes cover the input space? Justify your claim

**Answer**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EC | Boundary | Assumption | Test case selections | Expected Output |
| 1 | passphrases.containsKey(passbookUsername)=true | passBook.addUser(“Chris”,” Chris123”); | Using Guideline 3:  1. On point:  passphrases.containsKey(“Chris”)  2. Off point:  passphrases.containsKey(“Mary”) | 1. DuplicateUserException  2. Valid |
| 2 | length[passphrase] < 8 |  | Using Guideline 2:  1. On point: length[passphrase]=8  2. Off point: length[passphrase]=7 | 1. Valid  2. WeakPassphraseException |
| 3 | no passphrase[i] in  'a' - 'z' for all i the domain of passphrase |  | Using Guideline 3:  1. On point: passphrase =“ABCD1234”  2. Off point: passphrase =“abcABC123” | 1. WeakPassphraseException  2. Valid |
| 4 | no passphrase[i] in  'A' - 'Z' for all i the domain of passphrase |  | Using Guideline 3:  1. On point: passphrase = “abcd1234”  2. Off point: passphrase = “abcABC123” | 1. WeakPassphraseException  2. Valid |
| 5 | no passphrase[i] in  '0' - '9' for all i the domain of passphrase |  | Using Guideline 3:  1. On point: passphrase = “abcdABCD”  2. Off point: passphrase = “abcABC123” | 1. WeakPassphraseException  2. Valid |
| 6 | passphrases.containsKey(passbookUsername)=false **&&**  8<=length[passphrase] **&&**  passphrase[i] in  'a' - 'z'  **&&** in  'A' - 'Z' **&&**  in  '0' - '9' for some i the domain of passphrase | passBook.addUser(“Chris”,” Chris123”); | Using Guideline 4:  1. On point: identical test case as EC 1  2. Off point: identical test case as EC 1  Using Guideline 4:  8<=passphrase[i]:  3. On point: identical test case as EC 2  4. Off point: identical test case as EC 2  Using Guideline 4:  passphrase[i] in  'a' - 'z' for some i the domain of passphrase:  7. On point: identical test case as EC 3  8. Off point: identical test case as EC 3  Using Guideline 4:  passphrase[i] in  'A' - 'Z' for some i the domain of passphrase:  9. On point: identical test case as EC 4  10. Off point: identical test case as EC 4  Using Guideline 4:  passphrase[i] in  '0' - '9' for some i the domain of passphrase:  11. On point: identical test case as EC 5  12. Off point: identical test case as EC 5 | 1. Valid  2. DuplicateUserException  3. Valid  4. WeakPassphraseException  7. Valid  8. WeakPassphraseException  9. Valid  10. WeakPassphraseException  11. Valid  12. WeakPassphraseException |
| 7 | passbookUsername NOT IN passphrases | passBook.addUser(“Chris”,” Chris123”); | Using Guideline 3:  1.On point: passbookUsername != passphrases[i] for all i in the domain of passphrases  2.Off point: passbookUsername = passphrases[i] for i in the domain of passphrases | 1. NoSuchUserException  2. Valid |
| 8 | sessionIDs.get(passbookUsername) !=null | passBook.addUser(“Chris”,” Chris123”);  passBook.loginUser(“Chris”,” Chris123”); | Using Guideline 3:  1. On point: sessionIDs.get(  “Chirs”) !=null  2. Off point: sessionIDs.get(  ) ==null | 1. AlreadyLoggedInException  2. Valid |
| 9 | passphrases.get(passbookUsername) != passphrase | passBook.addUser(“Chris”,” Chris123”); | Using Guideline 1:  1. On point: passphrases.get(  “Chris”) == “Passw0rd”  2. Off point: passphrases.get(  “Chris”) == “Chris123” | 1. IncorrectPassphraseException  2. Valid |
| 10 | PassbookUsername IN passphrases **&&** sessionIDs.get(passbookUsername) ==null **&&** passphrases.get(passbookUsername) == passphrase | passBook.addUser(“Chris”,” Chris123”); | Using Guideline 4:  PassbookUsername IN passphrases:  1. On point: identical test case as EC 7  2. Off point: identical test case as EC 7  Using Guideline 4:  sessionIDs.get(passbookUsername) ==null:  3. On point: identical test case as EC 8  4. Off point: identical test case as EC 8  Using Guideline 4:  passphrases.get(passbookUsername) == passphrase :  5. On point: identical test case as EC 9  6. Off point: identical test case as EC 9 | 1. Valid  2. NoSuchUserException  3. Valid  4. AlreadyLoggedInException  5. Valid  6. IncorrectPassphraseException |
| 11 | passbookUsername == null |  | Using Guideline 3:  1. On point: passbookUsername = null  2. Off point: passbookUsername = “Chris” | 1. InvalidSessionIDException  2. Valid |
| 12 | url.getProtocol() =="http" |  | Using Guideline 3:  1. On point: url.getProtocol() ="http"  2. Off point: url.getProtocol() ="ftp" | 1. Valid  2. MalformedURLException |
| 13 | url.getProtocol() not in {"http","https"} |  | Using Guideline 4:  1. On point: identical test case as EC 12  2. Off point: identical test case as EC 12 | 1. MalformedURLException  2. Valid |
| 14 | passbookUsername != null **&&**  url.getProtocol() == “https” **&&**  urlUsername == null && urlPassword == null | passBook.addUser(“Chris”,” Chris123”); | Using Guideline 4:  1. On point: identical test case as EC 11  2. Off point: identical test case as EC 11  Using Guideline 4:  3. On point: identical test case as EC 13  4. Off point: identical test case as EC 13  Using Guideline 3:  5. On point: urlUsername =“Chirs”, urlPassword = null  6. Off point: urlUsername = “Chris” && urlPassword = “Chris123” | 1. Valid  2. InvalidSessionIDException  3. Valid  4. MalformedURLException  5. Valid  6. Valid |
| 15 | passbookUsername != null **&&**  url.getProtocol() == “https” **&&**  urlUsername != null && urlPassword != null |  | Using Guideline 4:  1. On point: identical test case as EC 11  2. Off point: identical test case as EC 11  Using Guideline 4:  3. On point: identical test case as EC 13  4. Off point: identical test case as EC 13  Using Guideline 3:  5. On point: identical test case as EC 14  6. Off point: identical test case as EC 14 | 1. Valid  2. InvalidSessionIDException  3. Valid  4. MalformedURLException  5. Valid  6. Valid |
| 16 | passbookUsername == null |  | Using Guideline 4:  1. On point: identical test case as EC 11  2. Off point: identical test case as EC 11 | 1. InvalidSessionIDException  2. Valid |
| 17 | url.getProtocol() =="http" |  | Using Guideline 4:  1. On point: identical test case as EC 12  2. Off point: identical test case as EC 12 | 1. Valid  2. MalformedURLException |
| 18 | url.getProtocol() not in {"http","https"} |  | Using Guideline 4:  1. On point: identical test case as EC 13  2. Off point: identical test case as EC 13 | 1. MalformedURLException  2. Valid |
| 19 | passbookUsername != null **&&**  url.getProtocol() == “https” **&&**  pt == null | passBook.addUser(“Chris”,” Chris123”); | Using Guideline 4:  1. On point: identical test case as EC 11  2. Off point: identical test case as EC 11  Using Guideline 4:  3. On point: identical test case as EC 13  4. Off point: identical test case as EC 13  Using Guideline 3:  5. On point: pt = null  6. Off point: pt ={“www.google.com”, {“Bob”,”Bob123”}} | 1. Valid  2. InvalidSessionIDException  3. Valid  4. MalformedURLException  5. NoSuchURLException  6. Valid |
| 20 | passbookUsername != null **&&**  url.getProtocol() == “https” **&&**  pt != null **&&** pt.get(url) == null | pt ={“www.google.com”, {“Bob”,”Bob123”}} | Using Guideline 4:  1. On point: identical test case as EC 11  2. Off point: identical test case as EC 11  Using Guideline 4:  3. On point: identical test case as EC 13  4. Off point: identical test case as EC 13  Using Guideline 4:  5. On point: identical test case as EC 19  6. Off point: identical test case as EC 19  Using Guideline 3:  5. On point: pt.get(“https://www.z.com”)  6. Off point:  pt.get(“https://www.google.com”) | 1. Valid  2. InvalidSessionIDException  3. Valid  4. MalformedURLException  5. NoSuchURLException  6. Valid  5. NoSuchURLException  6. Valid |
| 21 | passbookUsername != null **&&**  url.getProtocol() == “https” **&&**  pt != null **&&** pt.get(url) != null |  | Using Guideline 4:  1. On point: identical test case as EC 11  2. Off point: identical test case as EC 11  Using Guideline 4:  3. On point: identical test case as EC 13  4. Off point: identical test case as EC 13  Using Guideline 4:  5. On point: identical test case as EC 19  6. Off point: identical test case as EC 19  Using Guideline 4:  5. On point: identical test case as EC 20  6. Off point: identical test case as EC 20 | 1. Valid  2. InvalidSessionIDException  3. Valid  4. MalformedURLException  5. NoSuchURLException  6. Valid  5. Valid  6. NoSuchURLException |

**Task 5**

1. addUser():

**A**: if (passphrases.containsKey(passbookUsername))

else

**B**: if (passphrase.length() < MINIMUM\_PASSPHRASE\_LENGTH)

**C**: for (int i = 0; i < passphrase.length(); i++)

**D**: if ('a' <= passphrase.charAt(i) && passphrase.charAt(i) <= 'z')

**E**: else if ('A' <= passphrase.charAt(i) && passphrase.charAt(i) <= 'Z')

**F**: else if ('0' <= passphrase.charAt(i) && passphrase.charAt(i) <= '9')

**G**: if (!containsLowerCase || !containsUpperCase || !containsNumber)

|  |  |  |
| --- | --- | --- |
|  | Possible Outputs | Objective |
| A | true | 1 |
| false | 2 |
| B | true | 3 |
| false | 4 |
| C | true | 5 |
| false | 6 |
| D | true true | 7 |
| true false | 8 |
| false true | 9 |
| false false | 10 |
| E | true true | 11 |
| true false | 12 |
| false true | 13 |
| false false | 14 |
| F | true true | 15 |
| true false | 16 |
| false true | 17 |
| false false | 18 |
| G | true true true | 19 |
| true true false | 20 |
| true false true | 21 |
| false true true | 22 |
| true false false | 23 |
| false true false | 24 |
| false false true | 25 |
| false false false | 26 |

2. loginUser ():

**H**: if (!passphrases.containsKey(passbookUsername))

**I**: else if (sessionIDs.get(passbookUsername) != null)

**J**: else if (!passphrases.get(passbookUsername).equals(passphrase))

**K**: while (userIDs.containsKey(sessionID))

|  |  |  |
| --- | --- | --- |
|  | Possible Outputs | Objective |
| H | true | 27 |
| false | 28 |
| I | true | 29 |
| false | 30 |
| J | true | 31 |
| false | 32 |
| K | true | 33 |
| false | 34 |

3. updateDetails():

**L**: if (passbookUsername == null)

**M**: else if (!Arrays.asList(VALID\_URL\_PROTOCOLS).contains(url.getProtocol()))

**N**: if (urlUsername == null || urlPassword == null)

else

|  |  |  |
| --- | --- | --- |
|  | Possible Outputs | Objective |
| L | true | 35 |
| false | 36 |
| M | true | 37 |
| false | 38 |
| N | true true | 39 |
| true false | 40 |
| false true | 41 |
| false false | 42 |

4. retrieveDetails ():

**P**: if (passbookUsername == null)

**Q**: else if (!Arrays.asList(VALID\_URL\_PROTOCOLS).contains(url.getProtocol()))

**R**: if (pt == null)

**S**: if (pair == null)

|  |  |  |
| --- | --- | --- |
|  | Possible Outputs | Objective |
| O | true | 43 |
| false | 44 |
| P | true | 45 |
| false | 46 |
| Q | true | 47 |
| false | 48 |
| R | true | 49 |
| false | 50 |

• **Multiple-condition coverage in equivalence partitioning**

|  |  |  |  |
| --- | --- | --- | --- |
| addUser() | Pre-condition | Test Case | Meet with Objective |
| EC 1 | User “Chirs” has already in passphrases | addUser(“Chirs”,”Chris123”) | 1 |
| EC 2 | User “Chirs” is not in passphrases | addUser(“Chirs”,”Chris12”) | 2, 3 |
| EC 3 | User “Chirs” is not in passphrases | addUser(“Chirs”,”!CHRIS1}”) | 2, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 6, 22 |
| EC 4 | User “Chirs” is not in passphrases | addUser(“Chirs”,”!chris1}”) | 2, 4, 5, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 6, 21 |
| EC 5 | User “Chirs” is not in passphrases | addUser(“Chirs”,”! 123456}”) | 2, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 6, 20 |
| EC 6 | User “Chirs” is not in passphrases | addUser(“Chirs”,”! Chris1}”) | 2, 4, 5, 7, 11, 15, 6, 19 |

In terms of equivalence partitioning, there are 26 conditions for the method of addUser(). 4 conditions (23,24,25,26) are not met. The reason is that we use branch coverage. As a result, some redundant conditions are not considered.

Multiple-condition coverage result for equivalence partitioning : 22/26 = 84%

|  |  |  |  |
| --- | --- | --- | --- |
| loginUser() | Pre-condition | Test Case | Meet with Objective |
| EC 7 | User “Chirs” is not in passphrases | loginUser(“Chirs”,”Chris123”) | 28 |
| EC 8 | User “Chirs” has already logged in | loginUser (“Chirs”,”! adsasdAd}”) | 27, 30 |
| EC 9 | User “Chirs” has not logged in | loginUser (“Chirs”,”! Chris1}”) | 37, 29, 32 |
| EC 10 | User “Chirs” has not logged in | loginUser (“Chirs”,”! Chris1}”) | 27, 29, 31, 34, 33 |

In terms of equivalence partitioning, there are 8 conditions for the method of addUser(). all conditions are met.

Multiple-condition coverage result for equivalence partitioning : 8/8 = 100%

|  |  |  |  |
| --- | --- | --- | --- |
| updateDetails () | Pre-condition | Test Case | Meet with Objective |
| EC 11 | User “Chirs” has not logged in | updateDetails(“”, “http://[www.google.com](http://www.google.com)”, ”Bob”, ”123456”) | 36 |
| EC 12 | User “Chirs” has already logged in | updateDetails(“1543242”, “http://[www.google.com](http://www.google.com)”, ”Bob”, ”123456”) | 35, 37 |
| EC 13 | User “Chirs” has already logged in | updateDetails(“1543242”, “ftp://[www.google.com](http://www.google.com)”, ”Bob”, ”123456”) | 35, 38 |
| EC 14 | User “Chirs” has already logged in | updateDetails(“1543242”, “https://[www.google.com](http://www.google.com)”, ””, ””) | 35, 37, 39 |
| EC 15 | User “Chirs” has already logged in | updateDetails(“1543242”, “https://[www.google.com](http://www.google.com)”, ”Bob”, ”123456”) | 35, 37, 42 |

In terms of equivalence partitioning, there are 8 conditions for the method of addUser(). Two conditions (40, 41) are not met. The reason is that we use branch coverage. As a result, some redundant conditions are not considered.

Multiple-condition coverage result for equivalence partitioning : 6/8 = 75%

|  |  |  |  |
| --- | --- | --- | --- |
| retrieveDetails () | Pre-condition | Test Case | Meet with Objective |
| EC 16 | User “Chirs” has not logged in | retrieveDetails (“”, “http://[www.google.com](http://www.google.com)”) | 43 |
| EC 17 | User “Chirs” has already logged in | retrieveDetails (“1543242”, “http://[www.google.com](http://www.google.com)”) | 44, 45 |
| EC 18 | User “Chirs” has already logged in | retrieveDetails (“1543242”, “ftp://[www.google.com](http://www.google.com)”, ”Bob”, ”123456”) | 44, 46 |
| EC 19 | User “Chirs” has already logged in  User “Chris” PasswordTable is empty | retrieveDetails (“1543242”, “https://[www.google.com](http://www.google.com)”) | 44, 45, 47 |
| EC 20 | User “Chirs” has already logged in  User “Chris” PasswordTable is not empty  User “Chris” has not stored the URL of “https://[www.google.com](http://www.google.com)” | updateDetails(“1543242”, “https://[www.google.com](http://www.google.com)”, ”Bob”, ”123456”) | 44, 45, 48, 49 |
| EC 21 | User “Chirs” has already logged in  User “Chris” PasswordTable is not empty  User “Chris” has stored the URL of “https://[www.google.com](http://www.google.com)” | updateDetails(“1543242”, “https://[www.google.com](http://www.google.com)”, ”Bob”, ”123456”) | 44, 45, 48, 50 |

In terms of equivalence partitioning, there are 8 conditions for the method of addUser(). all conditions are met.

Multiple-condition coverage result for equivalence partitioning : 8/8 = 100%

• **Multiple-condition coverage in boundary-value**

|  |  |  |  |
| --- | --- | --- | --- |
| addUser() | Pre-condition | Test Case | Meet with Objective |
| EC 1 | User “Chirs” has already in passphrases | addUser(“Chirs”,”Chris123”) | 1 |
| EC 2 | User “Chirs” is not in passphrases | addUser(“Chirs”,”Chris12”) | 2, 3 |
| EC 3 | User “Chirs” is not in passphrases | addUser(“Chirs”,”!CHRIS1}”) | 2, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 6, 22 |
| EC 4 | User “Chirs” is not in passphrases | addUser(“Chirs”,”!chris1}”) | 2, 4, 5, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 6, 21 |
| EC 5 | User “Chirs” is not in passphrases | addUser(“Chirs”,”! 123456}”) | 2, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 6, 20 |
| EC 6 | User “Chirs” is not in passphrases | addUser(“Chirs”,”! Chris1}”) | 2, 4, 5, 7, 11, 15, 6, 19 |

In terms of boundary-value, there are 26 conditions for the method of addUser(). 4 conditions (23,24,25,26) are not met. The reason is that we use branch coverage. As a result, some redundant conditions are not considered.

Multiple-condition coverage result for boundary-value: 22/26 = 84%

|  |  |  |  |
| --- | --- | --- | --- |
| loginUser() | Pre-condition | Test Case | Meet with Objective |
| EC 7 | User “Chirs” is not in passphrases | loginUser(“Chirs”,”Chris123”) | 28 |
| EC 8 | User “Chirs” has already logged in | loginUser (“Chirs”,”! adsasdAd}”) | 27, 30 |
| EC 9 | User “Chirs” has not logged in | loginUser (“Chirs”,”! Chris1}”) | 37, 29, 32 |
| EC 10 | User “Chirs” has not logged in | loginUser (“Chirs”,”! Chris1}”) | 27, 29, 31, 34, 33 |

In terms of boundary-value, there are 8 conditions for the method of addUser(). all conditions are met.

Multiple-condition coverage result for boundary-value: 8/8 = 100%

|  |  |  |  |
| --- | --- | --- | --- |
| updateDetails () | Pre-condition | Test Case | Meet with Objective |
| EC 11 | User “Chirs” has not logged in | updateDetails(“”, “http://[www.google.com](http://www.google.com)”, ”Bob”, ”123456”) | 36 |
| EC 12 | User “Chirs” has already logged in | updateDetails(“1543242”, “http://[www.google.com](http://www.google.com)”, ”Bob”, ”123456”) | 35, 37 |
| EC 13 | User “Chirs” has already logged in | updateDetails(“1543242”, “ftp://[www.google.com](http://www.google.com)”, ”Bob”, ”123456”) | 35, 38 |
| EC 14 | User “Chirs” has already logged in | updateDetails(“1543242”, “https://[www.google.com](http://www.google.com)”, ””, ””) | 35, 37, 39 |
| EC 15 | User “Chirs” has already logged in | updateDetails(“1543242”, “https://[www.google.com](http://www.google.com)”, ”Bob”, ”123456”) | 35, 37, 42 |

In terms of boundary-value, there are 8 conditions for the method of addUser(). Two conditions (40, 41) are not met. The reason is that we use branch coverage. As a result, some redundant conditions are not considered.

Multiple-condition coverage result for boundary-value: 6/8 = 75%

|  |  |  |  |
| --- | --- | --- | --- |
| retrieveDetails () | Pre-condition | Test Case | Meet with Objective |
| EC 16 | User “Chirs” has not logged in | retrieveDetails (“”, “http://[www.google.com](http://www.google.com)”) | 43 |
| EC 17 | User “Chirs” has already logged in | retrieveDetails (“1543242”, “http://[www.google.com](http://www.google.com)”) | 44, 45 |
| EC 18 | User “Chirs” has already logged in | retrieveDetails (“1543242”, “ftp://[www.google.com](http://www.google.com)”, ”Bob”, ”123456”) | 44, 46 |
| EC 19 | User “Chirs” has already logged in  User “Chris” PasswordTable is empty | retrieveDetails (“1543242”, “https://[www.google.com](http://www.google.com)”) | 44, 45, 47 |
| EC 20 | User “Chirs” has already logged in  User “Chris” PasswordTable is not empty  User “Chris” has not stored the URL of “https://[www.google.com](http://www.google.com)” | updateDetails(“1543242”, “https://[www.google.com](http://www.google.com)”, ”Bob”, ”123456”) | 44, 45, 48, 49 |
| EC 21 | User “Chirs” has already logged in  User “Chris” PasswordTable is not empty  User “Chris” has stored the URL of “https://[www.google.com](http://www.google.com)” | updateDetails(“1543242”, “https://[www.google.com](http://www.google.com)”, ”Bob”, ”123456”) | 44, 45, 48, 50 |

In terms of boundary-value, there are 8 conditions for the method of addUser(). all conditions are met.

Multiple-condition coverage result for boundary-value: 8/8 = 100%