

# INF112 - Dossier de validation prévisionnel

*Clément LE GRUIEC – Elouan LE DUC*



**Avertissement :**

Cette fiche est similaire à celle qui sera utilisée lors de la recette finale de façon à ce que vous puissiez vous préparer aussi complètement que possible à cette opération. Pour le moment (BE2), vous ne renseignerez ici que la description des scénarios de tests.

<b>Identification du logiciel à valider</b>						
Nom du logiciel : <b>ToutAvis</b>			Nom des Auteurs : <b>LE DUC Elouan</b> <b>LE GRUIEC Clément</b>			
Taille : <b>13 pages</b>			Date de dernière mise à jour : <b>24/04/2020</b>			
<b>Identification des valideurs</b>						
Nom des valideurs : LE DUC Elouan LE GRUIEC Clément			Date de validation		Effort (heures)	
			<b>24/04/2020</b>		<b>6</b>	
<b>Fonctionnalités à tester :</b>  addMember ( )  <b>Critère (s) de validation<sup>1</sup> :</b>  0 Anomalies Majeures 0 Anomalies Mineures  <b>Bilan général de validation :</b>  <b>Les tests ont tous été passés avec succès !</b>						
Nb tests prévus	Tests passés	Tests OK <sup>2</sup>	Tests NOK	Tests NS	Anomalies majeures	Anomalies mineures
<b>15</b>	<b>15</b>	<b>15</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

<sup>1</sup> Ces critères peuvent comprendre : le nb de tests passés ( / nb de tests prévus), le nb de tests OK/NonOK, le nb d'anomalies majeures, le nb d'anomalies mineures ....

Synthèse des tests			
No	Identification du test	Résultat <sup>2</sup>	Gravité de l'anomalie <sup>3</sup>
1.1	Utilisation de addMember avec en paramètre un pseudo non instancié.	OK	Majeur
1.2	Utilisation de addMember avec en paramètre un pseudo ne contenant que des espaces.	OK	Majeur
1.3	Utilisation de addItemFilm avec en paramètre un password non instancié.	OK	Majeur
1.4	Utilisation de addMember avec en paramètre un password de moins de 4 caractères, autres que des «leading or trailing blanks ».	OK	Majeur
1.5	Utilisation de addMember avec en paramètre un profil non instancié.	OK	Majeur
2.1	Utilisations de addMember pour l'ajout de plusieurs membres « avec paramètres d'entrées corrects »	OK	Majeur
2.2	Utilisation de addMember avec en paramètre un pseudo identique au pseudo du premier membre « correct » ajouté.	OK	Majeur
2.3	Utilisation de addMember avec en paramètre un pseudo identique au pseudo du dernier membre « correct » ajouté.	OK	Majeur
2.4	Utilisation de addMember avec en paramètre le pseudo d'un membre existant (mais avec casse différente).	OK	Majeur
2.5	Utilisation de addMember avec en paramètre le pseudo d'un membre existant (mais avec des leading/trailing blanks).	OK	Majeur

<sup>2</sup> OK : le résultat du test correspond à ce qui était attendu

NOK : le résultat du test ne correspond pas à ce qui était attendu

NS : test non significatif (ne permet pas de conclure sur la validité ou non du résultat)

<sup>3</sup> Majeure / mineure

## Fiche de test N°3

### Objectif du test :

éprouver la méthode addMember sur les cas d'anomalie

Vérification de la levée de l'exception BadEntryException lors de l'utilisation de addMember « avec des paramètres d'entrées incorrects ».

Pour rappel : Les 5 cas de levée de l'exception BadEntryException par la méthode addMember prévus dans l'API sont :

- si le pseudo n'est pas instancié ou a moins de 1 caractère autre que des espaces,
- si le password n'est pas instancié ou a moins de 4 caractères, autres que des « leading or trailing blanks »,
- si le profil n'est pas instancié.

### Description (scénario) :

- Instanciation d'un Social Network
- 1.1 Utilisation de addMember avec en paramètre un pseudo non instancié.
- 1.2 Utilisation de addMember avec en paramètre un pseudo ne contenant que des espaces.
- 1.3 Utilisation de addMember avec en paramètre un password non instancié.
- 1.4 Utilisation de addMember avec en paramètre un password de moins de 4 caractères, autres que des « leading or trailing blanks ».
- 1.5 Utilisation de addMember avec en paramètre un profil non instancié.

### Résultats attendus :

Dans les 5 utilisations de addMember, l'exception BadEntryException doit être levée. De plus, après levée de l'exception BadEntryException lors de l'utilisation de addMember, le nombre de membres doit rester identique au nombre de membres avant utilisation de addMember.

### Résultats observés :

Correspond aux résultats attendus.

### Conclusion :

La vérification des saisies fonctionne parfaitement.

# Fiche de test N°4

## Objectif du test :

éprouver la méthode addMember sur les cas de fonctionnement standard :

- But principal : vérification de l'ajout de membres après utilisation de addMember « avec des paramètres d'entrées corrects »
- But secondaire : vérification de la levée de l'exception MemberAlreadyExistsException par la méthode addMember pour tous les cas prévus dans l'API.

## Description(scénario) :

- Instanciation d'un Social Network
- 2.1 Utilisations de addMember pour l'ajout de plusieurs membres « avec paramètres d'entrées corrects »
- 2.2 Utilisation de addMember avec en paramètre un pseudo identique au pseudo du premier membre « correct » ajouté.
- 2.3 Utilisation de addMember avec en paramètre un pseudo identique au pseudo du dernier membre « correct » ajouté.
- 2.4 Utilisation de addMember avec en paramètre le pseudo d'un membre existant (mais avec casse différente).
- 2.5 Utilisation de addMember avec en paramètre le pseudo d'un membre existant (mais avec des leading/trailing blanks).

## Résultats attendus :

L'ajout de membres « avec paramètres d'entrées corrects » ne doit pas lever d'exception, le nombre de membres après ces ajouts doit avoir augmenté en conséquence. Dans les 4 utilisations suivantes de addMember, l'exception MemberAlreadyExistsException doit être levée. De plus après levée de l'exception MemberAlreadyExistsException lors de l'utilisation de addMember, le nombre de membres doit rester identique au nombre de membres avant utilisation de addMember.

## Résultats observés :

Correspond au résultats attendus.

## Conclusion :

L'ajout des membre fonctionne parfaitement.

Résultat :

**Testing addMember()**

Final state of the social network : opinion.SocialNetwork@555590

**AddMemberTest : [ Nb of performed tests : 15 / nb of detected error(s) : 0 ]**

## AddItemFilmTest.java

```
package tests;

import opinion.ISocialNetwork;
import opinion.SocialNetwork;

import exceptions.BadEntryException;
import exceptions.MemberAlreadyExistsException;
import exceptions.NotTestReportException;

/**
 * Tests for the SocialNetwork.<i>addMember()</i> method.
 *
 * @author B. Prou, E. Cousin, GO
 * @version V2.0 - April 2018
 */

public class AddMemberTest {

    /**
     * Check that trying to add this new member (login, pwd, profile) raises a
     * BadEntry exception and does not change content of the
     * <i>ISocialNetwork</i>. If OK, the method just returns 0. If not OK,
     * displays an error message and returns 1.
     *
     * @param sn
     *      - the <i>ISocialNetwork</i>
     * @param login
     *      - new member's login
     * @param pwd
     *      - new member's password
     * @param profile
     *      - new member's profile
     * @param testId
     *      - the test ID that will prefix any error message displayed by
     *        this method
     * @param errorMessage
     *      - the error message that will be displayed if no exception is
     *        thrown when adding this member
     * @return 0 if the test is OK, 1 if not
     */
    private static int addMemberBadEntryTest(ISocialNetwork sn, String login,
                                             String pwd, String profile, String testId, String errorMessage) {

        int nbMembers = sn.nbMembers(); // Number of members when starting to
                                         // run this method

        try {
            sn.addMember(login, pwd, profile); // Try to add this member
            // Reaching this point means that no exception was thrown by
            // addMember()
            System.out.println("Err " + testId + " : " + errorMessage); // display
                                     // message
            return 1; // and return the "error" value
        } catch (BadEntryException e) { // BadEntry exception was thrown by
                                         // addMember() :
```

```

this is a good start!
                                                                    // Let's now check if
'sn' was not
                                                                    // impacted
        if (sn.nbMembers() != nbMembers) { // The number of members has
                                                                    //
changed : this is an error
                                                                    //
case.
        System.out.println("Err "+ testId+ " : BadEntry was thrown but the
number of members was changed"); // Display

        return 1; // return "error" value

    } else
        // The number of members hasn't changed, which is considered a
        // good indicator that 'sn' was not modified
        return 0; // return success value : everything seems OK, nothing
        // to display
    } catch (Exception e) { // An exception was thrown by addMember(), but
        // it was not the expected exception
BadEntry
        System.out.println("Err " + testId + " : unexpected exception. "
            + e); // Display a specific error message
        e.printStackTrace(); // Display contextual info about what happened
        return 1; // return error value
    }
}

/**
 * Check that trying to add this new member (login, pwd, profile) raises an
 * AlreadyExists exception and does not change content of the
 * <i>ISocialNetwork</i>. If OK, the method just returns 0. If not OK,
 * displays an error message and returns 1.
 *
 * @param sn
 *      - the <i>ISocialNetwork</i>
 * @param login
 *      - new member's login
 * @param pwd
 *      - new member's password
 * @param profile
 *      - new member's profile
 * @param testId
 *      - the test ID that will prefix any error message displayed by
 *      this method
 * @param errorMessage
 *      - the error message that will be displayed if no exception is
 *      thrown when adding this member
 * @return 0 if the test is OK, 1 if not
 */
private static int addMemberAlreadyExistsTest(ISocialNetwork sn,
        String login, String pwd, String profile, String testId,
        String errorMessage) {
    int nbMembers = sn.nbMembers(); // Number of members when starting to
        // process this
method

```



```

        try {
            sn.addMember(login, pwd, profile); // Try to add this member
            // Reaching this point means that no exception was thrown by
            // addMember()
            System.out.println("Err " + testId + " : " + errorMessage); // display

                                // the

                                // error

                                // message
            return 1; // and return the "error" value
        } catch (MemberAlreadyExistsException e) { // AlreadyExists exception was

// thrown by addMember() :

// this is a good start!

// Let's now check if 'sn'

// was not impacted
            if (sn.nbMembers() != nbMembers) {
                System.out
                    .println("Err "
                        + testId
                        + " : MemberAlreadyExists was
thrown, but the number of members was changed"); // Display

                                // a

                                // specific

                                // error

                                // message
            return 1; // and return the "error" value
        } else
            return 0; // return success value : everything is OK, nothing to
                                // display
    } catch (Exception e) { // An exception was thrown by addMember(), but
                                // it was not the expected exception
                                // AlreadyExists
        System.out.println("Err " + testId + " : unexpected exception. "
            + e); // Display a specific error message
        e.printStackTrace(); // Display contextual info about what happened
        return 1; // return error value
    }
}

/**
 * Check that this new member (login, pwd, profile) can be (and <i>is</i>)
 * added to the <i>ISocialNetwork</i>.</br> If OK, the method just returns 0
 * : the new member was added.</br> If not OK, an error message is displayed

```

```

* and 1 is returned ; the new member was not correctly added.
*
* @param sn
*     - the <i>ISocialNetwork</i>
* @param login
*     - new member's login
* @param pwd
*     - new member's password
* @param profile
*     - new member's profile
* @param testId
*     - the test ID that will prefix any error message displayed by
*       this method
* @return 0 if the test is OK, 1 if not
*/
private static int addMemberOKTest(ISocialNetwork sn, String login,
    String pwd, String profile, String testId) {
    int nbMembers = sn.nbMembers(); // Number of members when starting to
                                     // process this
method
    try {
        sn.addMember(login, pwd, profile); // Try to add this member
        // No exception was thrown. That's a good start !
        if (sn.nbMembers() != nbMembers + 1) { // But the number of members
// hasn't changed
// accordingly
            System.out.println("Err " + testId
                + " : the number of members (" + nbMembers
                + ") was not incremented"); // Error message
displayed
            return 1; // return error code
        } else
            return 0; // return success code : everything is OK, nothing to
                       // display
    } catch (Exception e) { // An exception was thrown by addMember() : this
                           // is an error case
        System.out
Error
            .println("Err " + testId + " : unexpected exception " + e); //
// message
// displayed
        e.printStackTrace(); // Display contextual info about what happened
        return 1; // return error code
    }
}

/**
 * <i>addMember()</i> main test :
 * <ul>
 * <li>check if members can be added</li>
 * <li>check if incorrect parameters cause addMember() to throw BadEntry
 * exception</li>
 * <li>check if adding already registered members cause addMember() to throw

```

```

* AlreadyExists exception</li>
* </ul>
*
* @return a summary of the performed tests
*/
public static TestReport test(){

    ISocialNetwork sn = new SocialNetwork();

    int nbBooks = sn.nbBooks(); // number of books in 'sn' (should be 0
                                // here)
    int nbFilms = sn.nbFilms(); // number of films in 'sn' (should be 0
                                // here)

    int nbTests = 0; // total number of performed tests
    int nbErrors = 0; // total number of failed tests

    System.out.println("Testing addMember()");

    // <=> test n°1

    // check if incorrect parameters cause addMember() to throw BadEntry
    // exception

    nbTests++;
    nbErrors += addMemberBadEntryTest(sn, null, "qsd fgh", "", "1.1",
                                       "addMember() doesn't reject null logins");
    nbTests++;
    nbErrors += addMemberBadEntryTest(
        sn,
        " ",
        "qsd fgh",
        "",
        "1.2",
        "addMember() doesn't reject logins that don't contain at least one
character other than space");
    nbTests++;
    nbErrors += addMemberBadEntryTest(sn, "B", null, "", "1.3",
                                       "addMember() doesn't reject null passwords");
    nbTests++;
    nbErrors += addMemberBadEntryTest(
        sn,
        "B",
        " qwd ",
        "",
        "1.4",
        "addMember() doesn't reject passwords that don't contain at least 4
characters (not taking into account leading or trailing blanks)");
    nbTests++;
    nbErrors += addMemberBadEntryTest(sn, "BBBB", "bbbb", null, "1.5",
                                       "addMember() doesn't reject null profiles");

    // <=> test n°2

    // populate 'sn' with 3 members

    nbTests++;

```

```

        nbErrors += addMemberOKTest(sn, "Paul", "paul", "lecteur impulsif",
                                    "2.1a");
        nbTests++;
        nbErrors += addMemberOKTest(sn, "Antoine", "antoine",
                                    "grand amoureux de la litt  rature", "2.1b");
        nbTests++;
        nbErrors += addMemberOKTest(sn, "Alice", "alice",
                                    "passionn  e de bande dessin  e", "2.1c");

        // try to add already registered members

        nbTests++;
        nbErrors += addMemberAlreadyExistsTest(sn, new String("Paul"),
        "abcdefghij", "", "2.2",
        "The login of the first member was accepted as login for a new
member");
        nbTests++;
        nbErrors += addMemberAlreadyExistsTest(sn, new String("Alice"),
        "abcdefghij", "", "2.3",
        "The login of the last member was accepted as login for a new
member");
        nbTests++;
        nbErrors += addMemberAlreadyExistsTest(
            sn,
            new String("anToine"),
            "abcdefghij",
            "",
            "2.4",
            "An already registered login, but with different case, was accepted
as login for a new member");
        nbTests++;
        nbErrors += addMemberAlreadyExistsTest(
            sn,
            new String(" Antoine "),
            "abcdefghij",
            "",
            "2.5",
            "An
already registered login, surrounded by leading/trailing blanks, was accepted as login for a new
member");
        nbTests++;
        nbErrors += addMemberAlreadyExistsTest(
            sn,
            "An" + "toi" + "ne",
            "abcdefghij",
            "",
            "2.6",
            "A String concatenation building an already registered login was
accepted as login for a new member");

        nbTests++;
        // check that 'sn' was not modified
        if (nbFilms != sn.nbFilms()) {
            System.out
                .println("Error : the number of films was unexpectedly
changed by addMember()");
            nbErrors++;

```

```

        }
        nbTests++;
        if (nbBooks != sn.nbBooks()) {
            System.out
changed by addMember());
                .println("Error : the number of books was unexpectedly
                nbErrors++;
        }

        // Display final state of 'sn'
        System.out.println("Final state of the social network : " + sn);

        // Print a summary of the tests and return test results
        try{
            TestReport tr = new TestReport(nbTests, nbErrors);
            System.out.println("AddMemberTest : " + tr);
            return tr;
        }
        catch (NotTestReportException e){ //This shouldn't happen
            System.out.println("Unexpected error in AddMemberTest test code - Can't
return valuable test results");
            return null;
        }
    }

    /**
     * Launches test()
     * @param args not used
     */
    public static void main(String[] args) {
        test();
    }
}

```