

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.

Identification du logiciel à valider

Nom du logiciel : ToutAvis	Nom des Auteurs : LE DUC Elouan LE GRUIEC Clément
Taille : 11 pages	Date de dernière mise à jour : 24/04/2020

Identification des valideurs

Nom des valideurs : LE DUC Elouan LE GRUIEC Clément	Date de validation	Effort (heures)
	24/04/2020	6

Fonctionnalités à tester :

addItemFilm()

Critère (s) de validation¹ :

0 Anomalies Majeures

Bilan général de validation :

Les tests ont tous été passés avec succès !

Nb tests prévus	Tests passés	Tests OK ²	Tests NOK	Tests NS	Anomalies majeures	Anomalies mineures
18	18	18	0	0	0	0

¹ 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	Résultat²	Gravité de l'anomalie ³
3.1	Utilisation de addItemFilm avec en paramètre un login non instancié	OK	Majeur
3.2	Utilisation de addItemFilm avec en paramètre un login non instancié	OK	Majeur
3.3	Utilisation de addItemFilm avec en paramètre un password non instancié.	OK	Majeur
3.4	Utilisation de addItemFilm avec en paramètre un password de moins de 4 caractères, autres que des « leading or trailing blanks »	OK	Majeur
3.5	Utilisation de addItemFilm avec en paramètre un titre non instancié	OK	Majeur
3.6	Utilisation de addItemFilm avec en paramètre un titre de moins de 1 caractères, autres que des espaces.	OK	Majeur
3.7	Utilisation de addItemFilm avec en paramètre un kind non instancié.	OK	Majeur
3.8	Utilisation de addItemFilm avec en paramètre un director non instancié.	OK	Majeur
3.9	Utilisation de addItemFilm avec en paramètre un scenarist non instancié.	OK	Majeur
3.10	Utilisation de addItemFilm avec en paramètre une durée non instancié.	OK	Majeur
3.11	Utilisation de addItemFilm avec en paramètre une durée négative.	OK	Majeur
4.1	Utilisations de addItemFilm pour l'ajout de plusieurs films « avec paramètres d'entrées corrects ».	OK	mineure
4.2	Utilisations de addItemFilm pour l'ajout d'un film existant (parmi les correct ajoutés).	OK	mineure
4.3	Utilisations de addItemFilm pour l'ajout d'un film existant avec une casse différentes pour le titre .	OK	mineure
4.4	Utilisations de addItemFilm pour l'ajout d'un film avec un utilisateur qui n'existe pas.	OK	Majeur
4.5	Utilisations de addItemFilm pour l'ajout d'un film avec un utilisateur qui existe mais mot de pass incorrect.	OK	Majeur

² 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)

³ Majeure / mineure

Fiche de test N°3

Objectif du test :

Eprouver la méthode addItemFilm() 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 11 cas de levée de l'exception BadEntryException par la méthode addItemFilm prévus dans l'API sont :

- Si le login 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 title (titre) n'est pas instancié ou a moins de 1 caractère autre que des espaces
- si le kind (genre) n'est pas instancié
- si le director (metteurs en scène) n'est pas instancié
- si le scenarist (scénariste) n'est pas instancié
- si la durée n'est pas strictement positive

Description (scénario) :

- Instanciation d'un Social Network
- **3.1** Utilisation de addItemFilm avec en paramètre un login non instancié.
- **3.2** Utilisation de addItemFilm avec en paramètre un login ne contenant que des espaces.
- **3.3** Utilisation de addItemFilm avec en paramètre un password non instancié.
- **3.4** Utilisation de addItemFilm avec en paramètre un password de moins de 4 caractères, autres que des « leading or trailing blanks ».
- **3.5** Utilisation de addItemFilm avec en paramètre un title non instancié
- **3.6** Utilisation de addItemFilm avec en paramètre un title de moins de 1 caractères, autres que des espaces.
- **3.7** Utilisation de addItemFilm avec en paramètre un kind non instancié.
- **3.8** Utilisation de addItemFilm avec en paramètre un director non instancié.
- **3.9** Utilisation de addItemFilm avec en paramètre un scenarist non instancié.
- **3.10** Utilisation de addItemFilm avec en paramètre une durée non instancié.
- **3.11** Utilisation de addItemFilm avec en paramètre une durée négative.

Résultats attendus :

Tous les test doivent réalisés avec succès. Les exceptions doivent être levées dans les 11 cas.

Résultats observés :

Correspond aux résultats attendus.

Conclusion :

La vérification des saisies fonctionne parfaitement.

Fiche de test N°4

Objectif du test :

éprouver la méthode **addItemFilm** sur les cas de fonctionnement standard

- But principal :
 - vérification de l'ajout de films après utilisation de **addItemFilm** « avec paramètres d'entrées corrects »
- But secondaire :
 - vérification de la levée de l'exception **NotMemberException** par la méthode.
 - vérification de la levée de l'exception **ItemFilmAlreadyExistsException** par la méthode.

Description(scénario) :

- Instanciation d'un Social Network
- **4.1** Utilisations de **addItemFilm** pour l'ajout de plusieurs films « avec paramètres d'entrées corrects ».
- **4.2** Utilisations de **addItemFilm** pour l'ajout d'un film existant (parmi les correct ajoutés).
- **4.3** Utilisations de **addItemFilm** pour l'ajout d'un film existant avec une casse différentes pour le titre .
- **4.4** Utilisations de **addItemFilm** pour l'ajout d'un film avec un utilisateur qui n'existe pas.
- **4.5** Utilisations de **addItemFilm** pour l'ajout d'un film avec un utilisateur correct mais mauvais password.

Résultats attendus :

Dans les 4 utilisations (4.2 à 4.5) de **addItemFilm**, les exceptions **ItemFilmAlreadyExistsException** ou **NotMemberException** doivent être relevées.

Lors de l'utilisation de **addItemFilm**, le nombre de films doit rester identique au nombre de films avant utilisation de **addItemFilm** si une exception est levée.

Autant de test que d'erreurs...

Résultats observés :

Correspond au résultats attendus.

Conclusion :

Les vérification des films et du membre fonctionnent parfaitement.

Résultat :

Testing addItemFilm()

--BadEntry --

Test 3.1 : BadEntryException was thrown exceptions.BadEntryException: Mauvaise entrée : paramètre non instancié

Test 3.2 : BadEntryException was thrown exceptions.BadEntryException: Mauvaise entrée : null ou longueur invalide, inférieur à 1

Test 3.3 : BadEntryException was thrown exceptions.BadEntryException: Mauvaise entrée : paramètre non instancié

Test 3.4 : BadEntryException was thrown exceptions.BadEntryException: Mauvaise entrée : null ou longueur invalide, inférieur à 4

Test 3.5 : BadEntryException was thrown exceptions.BadEntryException: Mauvaise entrée : paramètre non instancié

Test 3.6 : BadEntryException was thrown exceptions.BadEntryException: Mauvaise entrée : null ou longueur invalide, inférieur à 1

Test 3.7 : BadEntryException was thrown exceptions.BadEntryException: Mauvaise entrée : paramètre non instancié

Test 3.8 : BadEntryException was thrown exceptions.BadEntryException: Mauvaise entrée : paramètre non instancié

Test 3.9 : BadEntryException was thrown exceptions.BadEntryException: Mauvaise entrée : paramètre non instancié

Test 3.10 : BadEntryException was thrown exceptions.BadEntryException: Mauvaise entrée : la durée doit être supérieur à 0.0

Test 3.11 : BadEntryException was thrown exceptions.BadEntryException: Mauvaise entrée : la durée doit être supérieur à 0.0

--AlreadyExist --

Test4.1.1 : Film ajouté avec succès, tot de film: 1

Test4.1.2 : Film ajouté avec succès, tot de film: 2

Test4.1.3 : Film ajouté avec succès, tot de film: 3

Test 4.2 : AlreadyExistException was thrown exceptions.ItemFilmAlreadyExistsException - tot de film:3

Test 4.3 : AlreadyExistException was thrown exceptions.ItemFilmAlreadyExistsException - tot de film:3

-- Member no exist --

Test 4.4 : NotMember was thrown exceptions.NotMemberException: Echec de la connexion : L'utilisateur n'existe pas !

Test 4.5 : NotMember was thrown exceptions.NotMemberException: Echec de la connexion : Le mot de passe est incorrect !

AddItemFilmTest : [Nb of performed tests : 18 / nb of detected error(s) : 0]

AddItemFilmTest.java

```
package tests;

import opinion.ISocialNetwork;
import opinion.SocialNetwork;
import exceptions.*;

public class addItemFilmTest {

    /*
     *
     * Test du bad entry
     *
     */

    private static int addItemFilmBadEntryTest(ISocialNetwork sn, String login, String password,
String title,

        String kind, String director, String scenarist, int duration,

        String testId, String errorMessage) {

        //System.out.println("addItemFilmBadEntryTest");
        try {
            sn.addItemFilm(login, password, title, kind, director, scenarist, duration); //
Try to add a review
reviewItemFilm
            // Reaching this point means that no exception was thrown by
            System.out.println("Err " + testId + " : " + errorMessage); // display the error
message

            return 1; // and return the "error" value

        } catch (BadEntryException e) { // BadEntry exception was thrown by reviewItemFilm
: this is a good start! Let's now check if 'sn' was not impacted
            System.out.println("Test "+ testId+ " : BadEntryException was thrown " +e); //
Display a specific error message
            return 0; // return "error" value

        } catch (NotMemberException e) { // BadEntry exception was thrown by
reviewItemFilm : this is a good start! Let's now check if 'sn' was not impacted
            System.out.println("Test "+ testId+ " : NotMemberException was thrown "
+e); // Display a specific error message
            return 0; // return "error" value

        } catch (ItemFilmAlreadyExistsException e) { // BadEntry exception was thrown by
reviewItemFilm : this is a good start! Let's now check if 'sn' was not impacted
            System.out.println("Test "+ testId+ " : ItemFilmAlreadyExistsException was
thrown" +e); // Display a specific error message
            return 0; // return "error" value

        } catch (Exception e) { // An exception was thrown by reviewItemFilm, but it was not
the expected exception BadEntry

            System.out.println("Test " + testId + " : unexpected exception. "+ e); //
```

```

Display a specific error message
        e.printStackTrace(); // Display contextual info about what happened
        return 1; // return error value
    }

}

/*
 *
 * Tests standards NotMemberException
 *
 */

private static int addItemFilmNotMemberTest(ISocialNetwork sn, String login, String pwd,
String title, String kind,
String director, String scenarist, int duration, String testId, String errorMessage) {

    try {
        sn.addItemFilm(login, pwd, title, kind, director, scenarist, duration);
        System.out.println("Err " + testId + " : " + errorMessage); // display the error message
        return 1; // and return the "error" value

    } catch (NotMemberException e) {
        System.out.println("Test " + testId + " : NotMember was thrown "+ e); // display the error
message
        return 0;

    } catch (Exception e) { // An exception was thrown by AddItemFilm, but it was not the expected
exception
        // NotMember
        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
    }
}

/*
 *
 * addItemFilmTest
 *
 */

private static int addItemFilmTest(ISocialNetwork sn, String login, String password, String
title,
String kind, String director, String scenarist, int duration,
String testId) {

    int nbFilms = sn.nbFilms();
    // process this method
    try {
        sn.addItemFilm(login, password, title, kind, director, scenarist, duration);
        // No exception was thrown. That's a good start !
    }
}

```



```

changed accordingly      if (sn.nbFilms() != nbFilms + 1) { // But the number of members hasn't
                          System.out.println("Err " + testId+ " : the number of members (" + nbFilms+
") was not incremented"); // Error message displayed
                          return 1; // return error code
                          } else
                          System.out.println("Test" + testId+ " : Film ajouté avec succès, tot de film:
"+sn.nbFilms());
                          return 0; // return success code : everything is OK, nothing to
                          // display
                          } catch (ItemFilmAlreadyExistsException e) {
                          System.out
                          .println("Test " + testId + " : AlreadyExistexception  was thrown " + e +" - tot
de film:"+ sn.nbFilms());
                          return 0; // return error code

                          } catch (NotMemberException e) {
                          System.out.println("Test" + testId + " : NotMemberexception  was thrown" + e);
                          return 0;

                          }

                          } catch (BadEntryException e) {
                          System.out.println("Test " + testId + " : BadEntryExeption was thrown " + e);
                          return 0; // return error code

                          }

                          } catch (Exception e) {
                          System.out.println("Err " + testId + " : unexpected exception " + e);
                          return 1;
                          }

                          }

/*
 *
 * Réalisation des tests
 *
 */
public static TestReport test(){

    ISocialNetwork sn = new SocialNetwork();
    int nbTests=0,nbErrors=0;
    int nbBooks = sn.nbBooks();
    int nbFilms = sn.nbFilms();

    System.out.println("Testing addItemFilm()");
    System.out.println("\n--BadEntry -- \n");

    nbTests++;
    nbErrors+=addItemFilmBadEntryTest(sn, null, "1234",
"ouioui","DessAnim","UneMetteurEnScne", "unScenariste", 45, "3.1", "login non instancié");

    nbTests++;
    nbErrors+=addItemFilmBadEntryTest(sn, " ", "1234",
"ouioui","DessAnim","UneMetteurEnScne", "unScenariste", 45, "3.2", "login ne contenant que des
espaces");

```

```

        nbTests++;
        nbErrors+=addItemFilmBadEntryTest(sn, "michel", null,
"ouioui","DessAnim","UneMetteurEnScne", "unScenariste", 45, "3.3", "password non instancié");

        nbTests++;
        nbErrors+=addItemFilmBadEntryTest(sn, "michel", "1 ",
"ouioui","DessAnim","UneMetteurEnScne", "unScenariste", 45, "3.4", "password de moins de 4
caractères");

        nbTests++;
        nbErrors+=addItemFilmBadEntryTest(sn, "michel", "1234",
null,"DessAnim","UneMetteurEnScne", "unScenariste", 45, "3.5", "title non instancié");

        nbTests++;
        nbErrors+=addItemFilmBadEntryTest(sn, "michel", "1234",
"", "DessAnim", "UneMetteurEnScne", "unScenariste", 45, "3.6", "title de moins de 1 caractères");

        nbTests++;
        nbErrors+=addItemFilmBadEntryTest(sn, "michel", "1234", "ouioui", null,
"UneMetteurEnScne", "unScenariste", 45, "3.7", "kind non instancié");

        nbTests++;
        nbErrors+=addItemFilmBadEntryTest(sn, "michel", "1234", "ouioui", "DessAnim", null,
"unScenariste", 45, "3.8", "director non instancié");

        nbTests++;
        nbErrors+=addItemFilmBadEntryTest(sn, "michel", "1234",
"ouioui", "DessAnim", "UneMetteurEnScne", null, 45, "3.9", "scenarist non instancié");

        nbTests++;
        nbErrors+=addItemFilmBadEntryTest(sn, "michel", "1234",
"ouioui", "DessAnim", "UneMetteurEnScne", "unScenariste", 0, "3.10", "durée non instancié.");
//durée null crash

        nbTests++;
        nbErrors+=addItemFilmBadEntryTest(sn, "michel", "1234",
"ouioui", "DessAnim", "UneMetteurEnScne", "unScenariste", -1, "3.11", "durée négative");


        System.out.println("\n--AlreadyExist -- \n");

        //je créer un user valide pour ajouter des films:
        try {
            sn.addMember("michel", "1234", "Mon profile");
        } catch (BadEntryException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        } catch (MemberAlreadyExistsException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }

        //j'ajoute 3 films

        nbTests++;
        nbErrors+=addItemFilmTest(sn, "michel", "1234",

```

```

"ouioui","DessAnim","UneMetteurEnScne", "unScenariste", 45, "4.1.1");
    nbTests++;
    nbErrors+=addItemFilmTest(sn, "michel", "1234",
"nonon","DessAnim","UneMetteurEnScne", "unScenariste", 45, "4.1.2");
    nbTests++;
    nbErrors+=addItemFilmTest(sn, "michel", "1234",
"OuiNon","DessAnim","UneMetteurEnScne", "unScenariste", 45, "4.1.3");

    //System.out.println(sn.nbFilms());

    //ajout d'un film existant
    nbTests++;
    nbErrors+=addItemFilmTest(sn, "michel", "1234",
"ouioui","DessAnim","UneMetteurEnScne", "unScenariste", 45, "4.2");

    //ajout d'un film existant mais avec des espaces et une casse différente
    nbTests++;
    nbErrors+=addItemFilmTest(sn, "michel", "1234", " OuiOui
","DessAnim","UneMetteurEnScne", "unScenariste", 45, "4.3");

    System.out.println("\n-- Member no exist -- \n");

    nbTests++;
    nbErrors += addItemFilmNotMemberTest(sn, "dfgdrg", "gdjkgghrdhgjfd", "PeutEtre", "comédie",
"freud", "kant", 98, "4.4", "Member doesn't exist maid add quand même");
    nbTests++;
    nbErrors += addItemFilmNotMemberTest(sn, "michel", "gdjkgghrdhgjfd", "PeutEtre", "comédie",
"freud", "kant", 98, "4.5", "Member doesn't exist maid add quand même");

    System.out.println("\n\n");

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

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

}

```