|  |
| --- |
| **Rapport du projet ToutAvis**  **type de document**  nom de la formation  nom de domaine/semestre/projet  Année scolaire 2014/2015 |

|  |
| --- |
| auteur  département/service/fonction  courriel  auteur  département/service/fonction  courriel |

Sommaire

[1. organisation du projet 2](#_Toc420592272)

[1.1 Fiche de temps 2](#_Toc420592273)

[2. Rapport d’audit 3](#_Toc420592274)

[3. Rapport de recette 4](#_Toc420592275)

[4. Etat du produit livré 5](#_Toc420592276)

[4.1 Fiches de test 3 7](#_Toc420592277)

[4.2 Fiche de test 4 : 8](#_Toc420592278)

[4.3 Fiche de test 7 9](#_Toc420592279)

[4.4 Fiche de test 8 10](#_Toc420592280)

[4.5 Fiche de test 9 11](#_Toc420592281)

[4.6 Fiche de test 10 12](#_Toc420592282)

# organisation du projet

## Fiche de temps

Voici un tableau représentant le temps passé sur chacune des étapes du projet ToutAvis par notre binôme :

|  |  |  |  |
| --- | --- | --- | --- |
| **Séance** | **Temps en séance** | **Temps Hors séance** | **Notes** |
| BE1 | 3x2 | 0 | Exo cahier des charges |
| BE2 | 3x2 | 2 | Réflexion sur les tests à effectuer pour vérifier le bon fonctionnement du lot 1. |
| BE3 | 3x2 | 2 | Conception UML du lot 1. |
| TP1 | 3x2 | 2 | Conception UML du lot 1 et début de code de ce lot. |
| TP2 | 3x2 | 1 | Codage lot 1. |
| TP3 | 3x2 | 6 | Codage lot 1 + code test du lot1 et réflexion sur le code de test du lot 2. |
| TP4 | 3x2 | 20 | Code lot 2 et production du code test de ce lot. |
| TP5 | 3x2 | 6 | Relecture du code par un autre groupe et production d’un rapport d’audit par ce groupe. Critique de cet audit et modification de quelques éléments du code. |
| TP6 | 3x2 | 2 | Test du logiciel par un client qui nous transmet un rapport de test contenant les éventuelles erreurs rencontrées. Eventuelles corrections du code.  Finalisation du projet. |
| Total | 18 | 41 | * Temps total : 59 heures |

# Rapport d’audit

**A FINIR**

# Rapport de recette

Lors de la séance de TP n°5 le client (représenté par le binôme Finaritra RAOILIHARIZAVA/Valentin DOUDEMENT) a tester notre logiciel ToutAvis en deux différentes étapes :

* Test du logiciel via notre programme de test qui était compris dans le livrable
* Tets du logiciel avec leur propre programme de test

Une fois ces tests effectués le client nous a renvoyé ce rapport :

**A FINIR**

# Etat du produit livré

* Lot 1 :
  + Ajout de membres au réseau social. Fonctionnel
  + Ajout de livres au réseau social. Fonctionnel
  + Ajout de films au réseau social. Fonctionnel
  + Ajout d’avis par les membres sur les films et les lires. Fonctionnel
  + Possibilité de consulter les items via une recherche par le titre. Fonctionnel
* Lot 2 :
  + Un membre peut noter les avis d’autres membres. Fonctionnel
  + Un membre possède un Karma qui dépend des notes qu’il a reçus sur ses avis. Fonctionnel
  + Le poids d’une note contenue dans un avis ajouté par un membre à un item dépend de son karma. Fonctionnel
* Lot 3 :

Non étudié.

* Quels sont les points faibles et les points forts de notre livrable :

**A FINIR**

* Si c’était à refaire, que changerions-nous, le cas échéant, dans notre démarche de conception-réalisation ?

A travers ce projet nous nous sommes rendu compte de l’importance de la réflexion lors de la conception d’un logiciel. En effet il s’agit vraiment de penser à tous les cas possibles d’utilisations de ce logiciel pour pouvoir répondre à ces situations correctement. Lors de ce projet nous n’avons pas consacré assez de temps à ces étapes de réflexion et avons donc oublié certains cas d’utilisation qui mettaient en défaut notre logiciel et qu’il a donc fallu corriger par la suite. Un autre problème a été le respect d’une méthode objet pour la réalisation de notre livrable. N’ayant jamais codé en objet nous n’avons pas adopté une bonne démarche au départ en considérant les objets Membres et Items par exemple comme des structure de données et non des éléments répondants à des services. Nous n’avons donc pas assez utilisé de délégation.

**A FINIR/relire**

* Quels sont les avantages et inconvénients dans la démarche de conception en W ?

Pour ce projet nous avons donc utilisées une méthode de conception en W pour la réalisation des lots 1 et 2 ? Voici comment nous avons procédé :

Livraison finale

Analyse des besoins et conception

Test du livrable par le client

Audit du logiciel par un groupe externe

Test du logiciel grâce à notre programme de test.

Réalisation du lot1

Réalisation du lot2

Conception détaillée du lot1

Conception détaillée du lot2

L’avantage d’avoir utilisé cette méthode pour ce projet est d’avoir pris du recul entre la réalisation du Lot 1 et celle du Lot 2 en revenant à une étape de conception UML. Lors de cette étape de conception détaillée nous avons réfléchie à comment bien intégrer les fonctionnalités du Lot 2 dans le projet déjà existant.

**A FINIR**

Annexe

## Fiches de test 3

**Objectif du test : éprouver la méthode addItemBook sur les cas d'anomalie**

Vérification de la levée de l’exception BadEntry lors de l'utilisation de addItemBook « avec des paramètres d'entrées incorrects ».

Pour rappel : Les 9 cas de levée de l'exception BadEntry par la méthode addItemBook 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 leadings or trailing blanks.
* si le titre n'est pas instancié ou a moins de 1 caractère autre que des espaces.
* si le genre n'est pas instancié.
* si l'auteur n'est pas instancié.
* si le nombre de pages n'est pas positif.

Dépendances :

* TestInitialisation OK
* TestAddMember OK

**Description (scénario) :**

* Instanciation d'un Social Network contenant des membres, livres et films.
* 3.1 Utilisation de addItemBook avec en paramètre un pseudo non instancié.
* 3.2 Utilisation de addItemBook avec en paramètre un pseudo ne contenant que des espaces.
* 3.3 Utilisation de addItemBook avec en paramètre un password non instancié.
* 3.4 Utilisation de addItemBook avec en paramètre un password de moins de 4 caractères, autres que des « leadings or trailing blanks ».
* 3.5 Utilisation de addItemBook avec en paramètre un titre non instancié.
* 3.6 Utilisation de addItemBook avec en paramètre un titre ne contenant que des espaces.
* 3.7 Utilisation de addItemBook avec en paramètre un genre non instancié.
* 3.8 Utilisation de addItemBook avec en paramètre un auteur non instancié.
* 3.9 Utilisation de addItemBook avec en paramètre un nombre de page négatif.

**Résultats attendus :**

Dans les utilisations de addItemBook, l'exception BadEntry doit être levée. De plus, après levée de l'exception, le nombre de livres doit rester identique au nombre de livres avant utilisation de addItemBook.

**Résultats observés :**

L’exception BadEntry est bien levée pour les différents cas de test présentés ci-dessus.

**Conclusion :**

**A FINIR**

## Fiche de test 4 :

**Objectif du test : éprouver la méthode addItemBook sur les cas de fonctionnement standrard**

* Principal objectif : vérification de l'ajout de livres après l'utilisation de addItemBook « avec des paramètres d'entrée corrects ».
* Objectif secondaire : vérification de la levée de l’exception NotMember par la méthode addItemBook pour tous les cas prévus dans l'API.
* Objectif seondaire : vérification de la levée de l’exception ItemBookAlreadyExists par la méthode addItemBook pour tous les cas prévus dans l'API.

Pour rappel : le cas de levée de l'exception NotMember par la méthode addItemBook prévu dans l'API est :

* si le pseudo n'est pas celui d'un membre ou si le pseudo et le password de correspondent pas.

et le cas pour de levée pour l'exception ItemBookAlreadyExists est :

* si un item livre de même titre est déjà présent dans le SocialNetwork (même titre : indifférents à la casse et aux leadings et trailings blanks)

Dépendances :

* TestInitialisation OK
* TestAddMember OK

**Description (scénario) :**

* Instanciation d'un Social Network contenant des membres, livres et films.
* 4.1 Utilisation de addItemBook pour l'ajout de plusieurs livres « avec paramètres d'entrées corrects ». (Ajout de trois livres avec entrée correcte)
* 4.2 Utilisation de addItemBook avec en paramètre un pseudo d'un membre inexistant.
* 4.3 Utilisation de addItemBook avec en paramètre un pseudo d'un membre existant et avec un mauvais password.
* 4.4 Utilisation de addItemBook avec en paramètre un titre identique au titre du premier livre « correct » ajouté.
* 4.5 Utilisation de addItemBook avec en paramètre un titre identique au titre du dernier livre « correct » ajouté.
* 4.6 Utilisation de addItemBook avec en paramètre un titre d'un livre existant (mais avec casse différente).
* 4.7 Utilisation de addItemBook avec en paramètre un titre d'un livre existant (mais avec des leadings et trailings blanks).

**Résultats attendus :**

Dans les utilisations de addItemBook, les exceptions NotMember et ItemBookAlreadyExists doivent être levées. De plus, après levée de ces exceptions, le nombre de livres doit rester identique au nombre de livres avant utilisation de addItemBook.

**Résultats observés :**

Aucune exception n’est levée pour les cas corrects. Les exceptions NotMember et ItemBookAlreadyExists sont bien levées pour les cas où le membre n’est pas enregistré sur le réseau social et si le livre est déjà présent dans la base de donnée.

**Conclusion :**

**A FINIR**

## Fiche de test 7

**Objectif du test :** éprouver la méthode reviewItemBook sur les cas d’anomalies BadEntry.

Vérifier la levée de l’exception BadEntry lors de l’utilisation de reviewItemBook avec des entrées de paramètres incorrects.

Voici les différents cas de levée de l’exception BadEntry :

* 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 leadings or trailing blanks.
* si le titre n'est pas instancié ou a moins de 1 caractère autre que des espaces.
* si la note n'est pas comprise entre 0.0 et 5.0.
* si le commentaire n'est pas instancié.

Dépendances :

* TestInitialisation OK
* TestAddMember OK
* TestAddItemBook OK

**Description (scénario) :**

* Instancier un nouveau SocialNetwork contenant des membres, livres et films.
* 7.1 Utilisation de reviewItemBook avec en paramètre un pseudo non instancié ou qui a moins de 1 caractère autre que des espaces.
  + - 7.1a : Test avec un pseudo null.
    - 7.1b : Test avec un pseudo ne contenant que des espaces.
* 7.2 Utilisation de reviewItemBook avec en paramètre un password non instancié ou a moins de 4 caractères autres que des leadings or trailing blanks.
  + - 7.2a : Test avec un password null.
    - 7.2b : Test avec un password contenant une lettre entourée d’espaces.
* 7.3 Utilisation de reviewItemBook avec en paramètre un titre non instancié ou qui a moins de 1 caractère autre que des espaces.
  + - 7.3a : Test avec un titre null.
    - 7.3b : Test avec un titre contenant que des espaces.
* 7.4 Utilisation de reviewItemBook avec une note non comprise entre 0.0 et 5.0.
* 7.5 Utilisation de reviewItemBook avec un commentaire non instancié.

**Résultats attendus :**

Les utilisations précédentes de reviewItemBook avec des paramètres d’entrée incorrects doivent tous lever l’exception BadEntry.

**Résultats observés :**

L’exception BadEntry est bien déclenchée dans ces différents cas.

**Conclusion :**

**A FINIR**

## Fiche de test 8

**Objectif du test :** Eprouver la méthode reviewItemBook dans le cas d’un fonctionnement standard.

* Vérifier que l’ajout d’un avis sur un livre fonctionne correctement avec des paramètres d’entrées correctes.
* Vérifier la levée de l’exception NotMember si le pseudo et/ou le password ne correspondent pas lors de l’ajout d’un commentaire.
* Vérifier la levée de l’exception NotItem lorsque le titre mis en paramètre ne correspond pas à un livre enregistré.

Dépendances :

* TestInitialisation OK
* TestAddMember OK
* TestAddItemBook OK

**Description (scénario) :**

* Instancier un nouveau SocialNetwork contenant des membres, livres et films.
* 8.1 Ajout d’avis sur les books (Les 3 membres ajoute un avis par livre (il y a trois livres déjà enregistrés).
* 8.2 Utilisation de reviewItemBook avec un pseudo non enregistré.
* 8.3 Utilisation de reviewItemBook avec un pseudo bon mais un password faux.
* 8.4 Utilisation de reviewItemBook avec un titre de livre non enregistré.
* 8.5 On vérifie la mise à jour de la note d’un utilisateur si celui-ci redonne son avis sur le même livre.

**Résultats attendus :**

Le scénario ci-dessous permet de :

* Vérifier que l’ajout d’avis sur des livres en entrant des paramètres correctes fonctionne bien (pas de levée d’exception).
* Vérifier la levée de l’exception NotMember pour un avis donnée avec un pseudo et/ou un password erroné et de l’exception NotItem pour un titre de livre non enregistré.

**Résultats observés :**

Aucune exception n’est levée dans les cas sans erreurs et les exceptions NotMember et NotItem sont bien déclenchées dans les cas de tests correspondants.

##### **Conclusion :**

**A FINIR**

## Fiche de test 9

**Objectif du test :** éprouver la méthode reviewOpinions sur les cas d’anomalies BadEntry.

Vérifier la levée de l’exception BadEntry lors de l’utilisation de reviewOpinions avec des entrées de paramètres incorrects.

Voici les différents cas de levée de l’exception BadEntry :

* 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 leadings or trailing blanks.
* si le titre n'est pas instancié ou a moins de 1 caractère autre que des espaces.
* si la note n'est pas comprise entre 0.0 et 5.0.
* si le commentaire n'est pas instancié.

**Description (scénario) :**

* Instancier un nouveau SocialNetwork contenant des membres, livres et films.
* Ajout d’un avis sur un item déjà présent dans le réseau social par « personne1 »
* 9.1 Utilisation de reviewOpinions avec en paramètre un pseudo non instancié ou qui a moins de 1 caractère autre que des espaces.
  + 9.1a : Test avec un pseudo1 null.
  + 9.1b : Test avec un pseudo1 ne contenant que des espaces.
  + 9.1c : Test avec un pseudo2 null.
  + 9.1d : Test avec un pseudo2 ne contenant que des espaces.
* 9.2 Utilisation de reviewOpinions avec en paramètre un password non instancié ou a moins de 4 caractères autres que des leadings or trailing blanks.
  + 9.2a : Test avec un password null.
  + 9.2b : Test avec un password contenant une lettre entourée d’espaces.
* 9.3 Utilisation de reviewOpinions avec en paramètre un titre non instancié ou qui a moins de 1 caractère autre que des espaces.
  + 9.3a : Test avec un titre null.
  + 9.3b : Test avec un titre contenant que des espaces.
* 9.4 Utilisation de reviewOpinions avec une note non comprise entre 0.0 et 5.0.
* 9.5 Utilisation de reviewOpinions avec un type non instancié ou différent de « book » ou « film ».
  + 9.5a : Test avec un type null
  + 9.5b : Test avec un type égal à « typeincorrect »

**Résultats attendus :**

L’excpetion BadEntry doit être levée pour chacun de ces cas de tests avec des entrées incorrectes.

**Résultats observés :**

L’exception Badentry est bien levée pour tous ces cas de tests.

##### **Conclusion :**

**A FINIR**

## Fiche de test 10

**Objectif du test :** Eprouver la méthode reviewOpinions dans le cas d’un fonctionnement standard.

* Vérifier que l’ajout d’un avis sur un livre ou un film fonctionne correctement avec des paramètres d’entrées correctes.
* Vérifier la levée de l’exception NotMember si le pseudo et/ou le password ne correspondent pas lors de l’ajout d’une opinion.
* Vérifier la levée de l’exception NotItem lorsque le titre mis en paramètre ne correspond pas à un item enregistré.
* Vérifier la levée de l’exception NotReview lorsque l’on ne trouve pas de review correspondant aux critères entrés

**Description(scénario)  :**

* Instancier un nouveau SocialNetwork contenant des membres, livres et films.
* 10.1 Ajout d’avis sur les items déjà présents sur le réseau social
* 10.1a : « personne2 » note l’avis de « personne1 » sur le livre « Arts martiaux ».
* 10.1b : « personne3 » note l’avis de « personne1 » sur le livre « Arts martiaux ».
* 10.1c : « personne2 » note l’avis de « personne1 » sur le livre « La Chine ».
* 10.1d : « personne3 » note l’avis de « personne1 » sur le livre « La Chine ».
* 10.2 Un membre tente de noter un avis qu’il a déjà noté.
* 10.3 Un membre tente de noter un de ses propres avis.
* 10.4 Vérification que le Karma d’un membre noté à bien été mis à jour (« personne1 ») et que son karma à bien une influence sur le poids de la note qu’il met à un item. (Pour cela nous avons calculer des valeurs qui doivent apparaitre si le karma à bien une influence et nous vérifions ces résultats).

**Résultats attendus :**

L’ajout d’opinions sur les avis d’un membre de doit pas déclenché de levée d’exception lorsque les paramètres entrés sont corrects.

Lorsqu’un membre tente de noter un avis qu’il a déjà noté une exception est levée : « Member Already Opinion ».

Lorsqu’un membre tente de noter un de ses propres avis une exception est levée : « SameMember ».

**Résultats observés :**

**A FINIR**

**Conclusion :**

**A FINIR**

|  |
| --- |
| 3_adresses_dossier |