

Binôme :  
Pierrick CHOVELON  
Julien TIRON

FIP 1A



# Rapport Projet ToutAvis INF112

**Formation d'ingénieur par partenariat**

Version : 1

26 mai 2015



# Table des matières

<b>Introduction</b>	<b>3</b>
<b>Fiches tâches/temps</b>	<b>4</b>
<b>Diagramme de classes et diagrammes de séquences</b>	<b>5</b>
<b>Rapport d'audit</b>	<b>9</b>
<b>Rapport de recette</b>	<b>10</b>
<b>Bilan</b>	<b>11</b>
<b>Bilan général</b>	<b>12</b>
<b>Annexes</b>	<b>13</b>

# Introduction

Ce rapport a pour but de dresser un bilan le plus complet possible du projet mené en INF112. Vous trouverez dans ce rapport les différentes informations concernant le projet ToutAvis et le travail que nous avons fourni pour le mener à bien. Vous trouverez dans chaque section quelques explications permettant de mieux comprendre notre démarche.

## Fiche tâches/temps

# Diagramme de classes et diagrammes de séquences

Vous trouverez dans les pages suivantes, les 3 diagrammes qui nous ont été demandés.

Nous avons essayé de faire un diagramme de classe le plus simple possible pour faciliter la compréhension de la structure du projet pour des personnes extérieures à celui-ci.

Le diagramme de séquence de `reviewOpinion` (Fig. 2) peut sembler trop simple. Nous avons choisi de ne représenter que le cas de fonctionnement normal (i.e les données passées en paramètres sont valides).

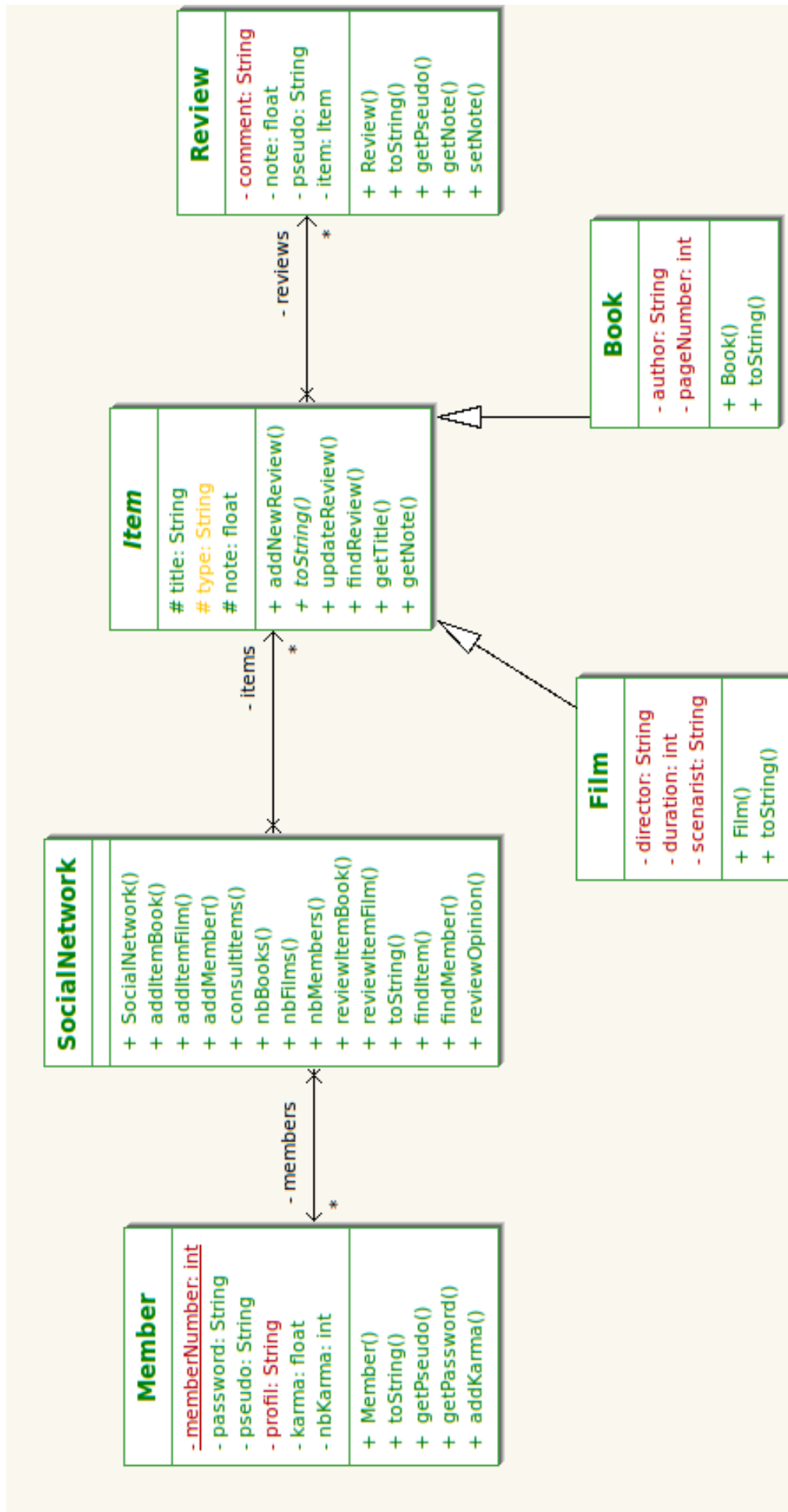


Figure 1 – Diagramme général de classes

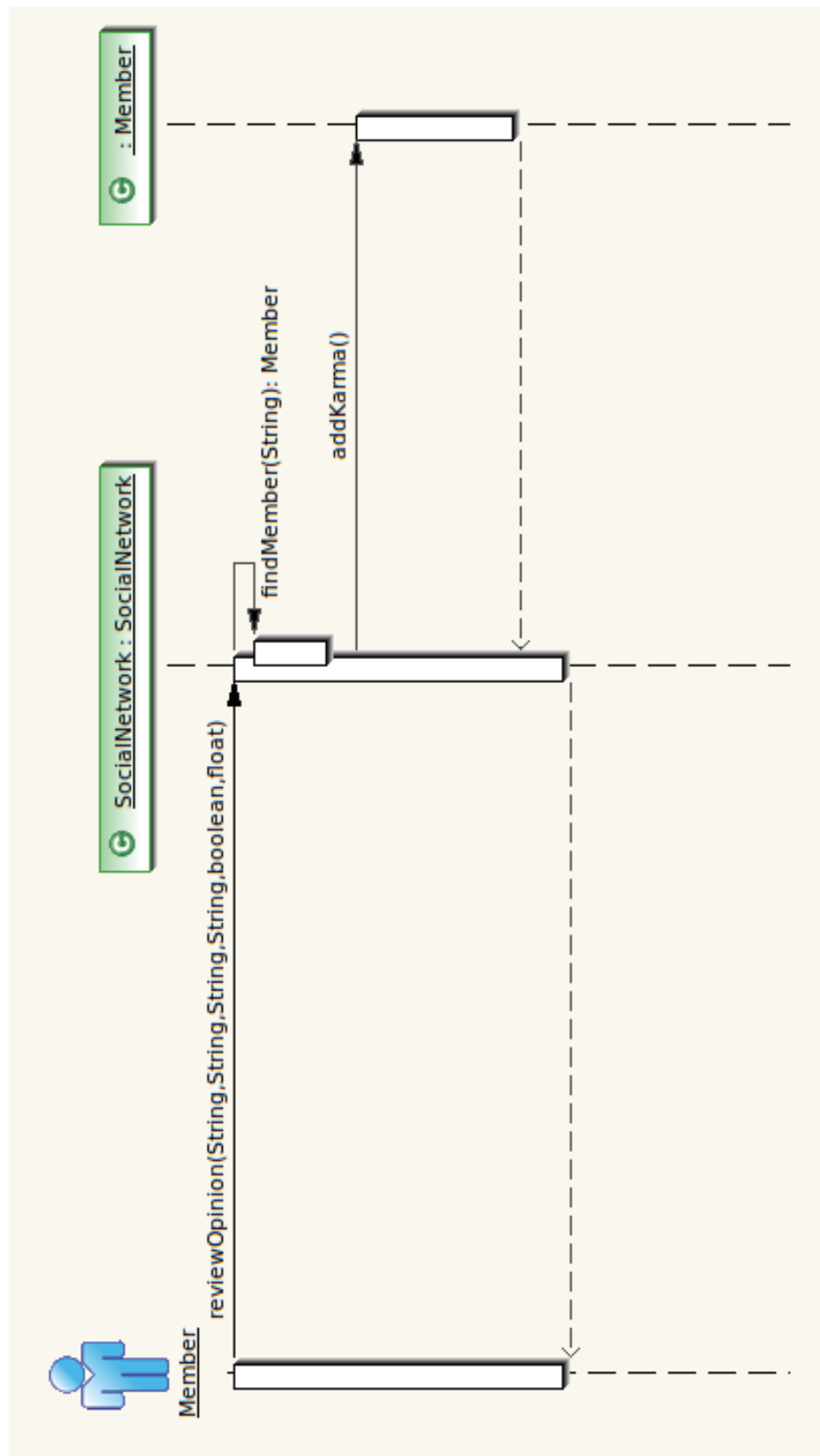


Figure 2 – Diagramme de séquence de reviewOpinion

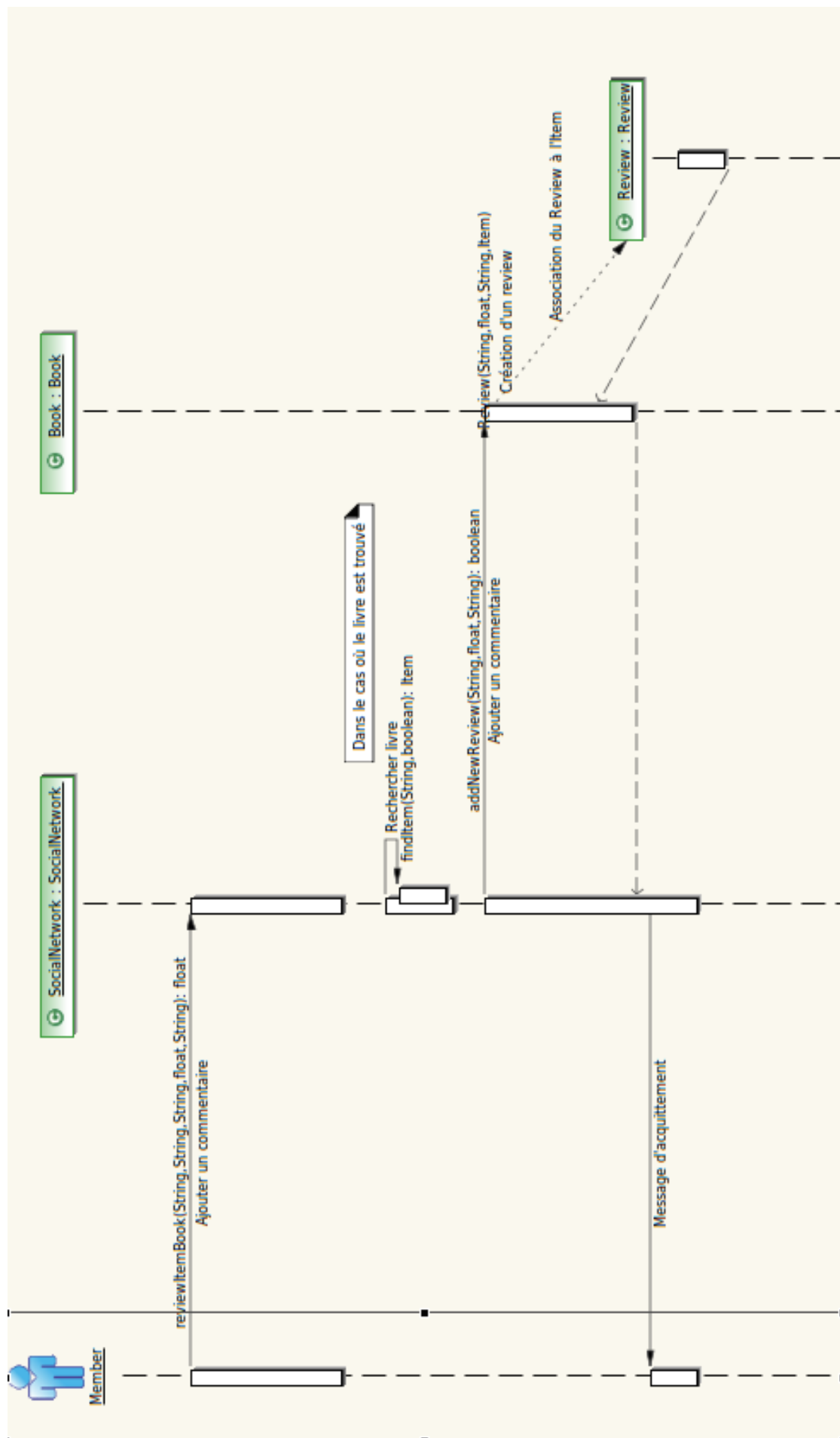


Figure 3 – Diagramme de séquence de reviewItemBook



# Rapport d'audit

Le rapport d'audit effectué par un autre binôme nous a amené à changer notre code. Au total, se sont 8 remarques qui nous ont été faites. 7 d'entre elles concernaient un problème de fond et 1 un problème de forme. Nous avons acceptés 7 remarques. La remarque que nous avons refusée ne nous semblait pas opportune (gravité mineure).

La remarque de forme était très pertinente. En effet, le problème remonté était que nous utilisions toujours les mêmes conditions pour tester la validité des paramètres donnés (*pseudo*, *password*, etc). Le copier-coller, rendant la tâche facile, nous à permis de réutiliser les condions des blocs *if* aisément. Cependant, il est plus ingénieux et réfléchi de créer des fonctions qui renvoient *true* ou *false* selon un paramètre donné. Ainsi, s'il faut modifier la manière dont est testé le *password* par exemple, il suffit de modifier la fonction. Toutes les conditions utilisant cette fonction seront donc à jour, contrairement à la méthode que nous avons utilisée.

Les autres remarques, bien que pertinentes, n'étaient pas rès grave et relevait la plupart du temps d'un manque d'explication dans les commentaires.

# Rapport de recette

Lors de cette recette, aucune erreur n'a été relevée par le binôme client. Aucune modification n'a donc été apportée. Le fait que nos tests, ainsi que les tests du binôme client, ne relèvent aucune erreur nous permet donc de s'assurer de la conformité de notre livrable par rapport au cahier des charges.

# Bilan

# Bilan général

Pour dresser un bilan général du travail réalisé, nous pouvons tout d'abord dire que nous avons réussi à s'organiser au sein du binôme pour se répartir au mieux les tâches. De plus, l'utilisation de solution de «versioning» à grandement facilité le partage et la mise en commun du code.

- *Quels sont les points faibles et les points forts de votre livrable ?*

Le livrable que nous proposons est, d'une part, simple à comprendre grâce à la javadoc et, d'autre part, fidèle au cahier des charges grâce notamment à la batterie de tests effectués sur celui-ci. En plus des tests «classiques» sur les méthodes et les classes qui vont être utilisées, nous avons effectués des tests de rendement sur l'utilisation de celle-ci.

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

Pour reprendre la remarque relevée lors du rapport d'audit, nous pouvons dire qu'une des erreurs que nous avons faite a été de commencer le codage des méthodes trop rapidement. En soit, le premier code proposé n'était pas mauvais, mais il n'était pas correctement structuré. En effet, un temps de réflexion au préalable, sur les fonctions et les conditions qui allaient être appelées fréquemment, nous aurait permis de gagner du temps en fin de projet.

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

# Annexes