

Coo: Choix de conception P2

Charles Ninane, Baptiste Degryse, Niels Sayez, Victorien
Sonneville, Lucie Hermand, Olivier Bronchain

1 Introduction

Durant ces dernières semaines, il nous a été demandé de modeliser les classes qui se retrouveraient dans le code de notre application WhatToWatch. Pour se faire nous avons d'abord écrit des Users Stories, ensuite nous avons écrit des cartes CRC. Avec celles-ci nous avons pu modeliser nos schémas URM relationnelle et puis sequencielle.

2 Creation des diagrammes

2.1 Users-Stories

La premiere étape pour realiser nos classes était de savoir exactement quelles fonctionnalités notre application devait remplir. En se basant sur nos faits élémentaires, sur nos envies et les consignes fournies, nous avons écrit ces histoires d'utilisateurs. Celles-ci sont disponibles dans un autre fichier. Pour permettre une meilleur visibilité pour la suite des opérations, il a été choisit de les divisées en plusieurs catégories:

- L'interaction avec l'utilisateur
- Les recherches faites par l'utilisateur
- L'interaction entre l'utilisateur et les films
- La gestion des utilisateurs
- La gestion des films
- Les droits relatifs aux administrateurs

2.2 Schema relationel UML

En ce qui concerne le schéma UML, nous avons décidé de choisir un certain nombre de classes. Celles-ci sont orchestrées autour des deux plus importantes (*Utilisateur* et *Film*). En effet toutes les autres classes hérites ou sont des arguments de ces deux classes. Voici donc une explication des choix de conception.

Premièrement, nous avons décidé de definir la classe *Administrateur* comme héritant de *Utilisateur* car un administrateur c'est qu'un utilisateur avec des droits entendus.

Deuxiemement, il a été choisi de mettre les classes *Suggestion* et *Eval* comme des compositions car une fois qu'un utilisateur est supprimé, nous voulons de

toutes ses traces disparaissent. Pour les mêmes raisons, les classes *Illustration*, *Prix*, *Commentaire* et *MotCle* sont des compositions de la classe *Film*.

Enfin, pour qu'un utilisateur puisse changer les cotes qu'il donne à un film, la classe *Utilisateur* contient une liste avec toutes les *Eval* qu'un utilisateur a faite

2.3 Schema sequentielle UML

2.3.1 Ajout d'une note à un film

Comme dit plus haut, quand un utilisateur ajoute une note à un film il faut qu'à la fin de cette opération l'utilisateur se souvienne de la note qu'il a donné et que le film ait actualisé la valeur de sa note globale. Pour y arriver nous avons choisis la séquence qui se trouve dans le fichier ci joint.

Une fois que l'utilisateur veut rajouter une evaluation, il appelle la fonction *AddEvalToFilm()* sur le film choisit. Ce film se charge de récupérer la note qui doit être attribué grace à la fonction *getInt()*. Il cree alors une nouvelle *Eval* qui contient la cote et son nom. Il actualise sa note global et la renvoie à l'utilisateur en question. La classe *Utilisateur* insert alors cette *Eval* dans sa liste. Ainsi les objectifs sont remplis.

2.4 Rechercher un film dans les cinemas

Quand un utilisateur veut aller voir un film dans un cinema, il faut que le cinema le plus proche de lui soit proposer. Pour se faire voici l'explication de la séquence ci-jointe.

Quand un utilisateur a chosit de rechercher un film, celui-ci appelle la fonction *checkCinema()* sur ce film. Le film se charge alors de regarder quels *Cinema* jouent le film en faisant une boucle sur toutes les instances de cette classe. La fonction *isPlayingFilm()* permet de dire si un film joue ou non le film. On compare alors la distance qui sépare l'utilisateur et le cinema. La classe *Film* retourne alors à *Utilisateur* le cinema et la scéance la plus proche.

2.5 Suggestion d'un film

Il est souhaité de pouvoir suggerer à une autre *Utilisateur* un film qu'il pourrait regarder. Afin d'implementer cette fonctionnalité, nous avons choisis de proceder de la manière suivante.

Quand un utilisateur choisit de suggerer un film, il appelle la fonction *creeSuggestion()* sur le film choisit qui s'occupe de creer la *Suggestion*. Celle-ci est initialisé avec le nom du film et de l'utilisateur A. Une fois que la *Suggestion* est crée, l'utilisateur A la récupère. Il appelle alors la fonction *SuggereToUtilisateur()* sur l'utilisateur B. Celui place alors la nouvelle suggestion dans sa liste de suggestion grace à la méthode *addSuggestion()*.