

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 modéliser les classes qui se retrouveraient dans le code de notre application WhatToWatch. Pour ce faire nous avons d’abord écrit des Users Stories, ensuite nous avons écrit des cartes CRC. Avec celles-ci nous avons pu modéliser nos schémas UML relationnels et puis séquentiels.

2 Création des diagrammes

2.1 Users-Stories

La première étape pour réaliser 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 meilleure visibilité pour la suite des opérations, il a été choisi de les diviser en plusieurs catégories:

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

2.2 Schéma relationnel 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 héritent où sont des arguments de ces deux classes. Voici donc une explication des choix de conception.

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

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

que 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 faites.

2.3 Schema sequentiel 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 choisi la séquence qui se trouve dans le fichier ci joint.

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

2.4 Rechercher un film dans les cinémas

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

Quand un utilisateur a choisi 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 cinéma. La classe *Film* retourne alors à *Utilisateur* le cinéma et la séance la plus proche.

2.5 Suggestion d'un film

Il est souhaité de pouvoir suggérer à une autre *Utilisateur* un film qu'il pourrait regarder. Afin d'implémenter cette fonctionnalité, nous avons choisi de procéder de la manière suivante.

Quand un utilisateur choisit de suggérer un film, il appelle la fonction *creerSuggestion()* sur le film choisi qui s'occupe de créer la *Suggestion*. Celle-ci est initialisée avec le nom du film et de l'utilisateur A. Une fois que la *Suggestion* est créée, l'utilisateur A la récupère. Il appelle alors la fonction *SuggereToUtilisateur()* sur l'utilisateur B. Celui-ci place alors la nouvelle suggestion dans sa liste de suggestion grâce à la méthode *addSuggestion()*.