

Rapport de synthèse : Bibliothèque Blu-rays

Le but de ce projet était d'implémenter 4 fonctionnalités à propos d'une bibliothèque de Blu-rays :

- Un gestionnaire de Blu-rays avec un affichage des Blu-rays à notre disposition. Ce gestionnaire permet d'afficher des informations complémentaires concernant un Blu-ray en cliquant dessus ou bien de le supprimer.
- Une page d'ajout d'un Blu-ray avec plusieurs informations le concernant. Dans notre cas, nous n'avons pas implémenté la possibilité d'ajouter de langue ni de sous-titre à celui-ci car ce n'étaient pas les informations les plus pertinentes à avoir. Cependant, la base de données MySQL possède les tables permettant d'implémenter tout ça.
- Une page permettant d'ajouter une personne en base de données qui était optionnelle mais qui permet de peupler plus rapidement notre bdd.
- La possibilité d'emprunter un Blu-ray à quelqu'un à travers une API et de pouvoir lui rendre. Cela implique que lorsque l'on emprunte un Blu-ray, il apparaît dans notre bibliothèque même s'il ne nous appartient pas réellement. Nous ne pouvons pas consulter plus d'informations le concernant car l'API ne renvoie que quelques données. A la place, nous pouvons le rendre, ce qui a pour effet de le supprimer de notre liste et de le faire réapparaître chez la personne à qui nous l'avons emprunté.

Toutes ces fonctionnalités ont été implémentées et sont fonctionnelles sauf peut-être le rendu du Blu-ray car nous l'avons fait après la dernière séance de cours donc nous n'avons pas pu tester avec un autre groupe ou même en interne entre nous. Pour organiser le travail, nous avons décidé de séparer le front-end du back-end plutôt que d'implémenter une fonctionnalité entière de bout en bout. Ainsi, Briac s'est occupé de la partie front-end et communication depuis et vers le back et Hugo de la partie de traitement back-end et appels vers la BDD car chacun était plus à l'aise dans ce domaine respectif. Nous communiquions donc beaucoup sur les besoins de l'autre pour par exemple les données à afficher et donc à aller chercher en base de données ou les données qui étaient renvoyées par un formulaire avec le front et qui devaient insérer de nouvelles données en bdd. Cette organisation a été efficace même si elle ne nous a peut-être pas permis de voir toutes les subtilités du .Net pour chacun.

Difficultés rencontrées :

Notre première difficulté a été rencontrée lors de l'installation de l'IDE car nous étions tous les deux sur un environnement Linux ne nous permettant pas de télécharger Visual Studio mais de passer par Riders (IDE développé par JetBrains) qui fonctionne très bien mais qui demande plus de configuration pour créer un premier projet car nous n'avions pas de version 6 de .Net et nous avons donc du télécharger .Net core 3.1 en suivant plusieurs tuto pas toujours très bien expliqué car il ne doit pas y avoir beaucoup de développeur de ce langage sous Linux. Malgré cela, nous avons réussi à avoir un projet fonctionnel avec quelques différences dans les fichiers de configuration notamment au niveau du Program.cs qui avaient

des méthodes inconnues en .Net core 3.1. Un autre problème a été la compréhension de l'architecture du projet qui était nouveau pour tous les deux. Plutôt habitué au modèle MVC, l'apparition des ViewModels nous a perturbé après la première séance et il a fallu plonger un peu plus dans le code pour comprendre son fonctionnement. La deuxième partie du projet sur l'API a été beaucoup plus simple car nous avons compris le fonctionnement du .Net ainsi que le modèle de conception utilisé. Le dernier problème était encore une fois lié à l'environnement Linux qui ne nous permettait pas d'installer le serveur web IIS. Nous avons essayé chacun de notre côté de déployer un serveur soit Apache pour Hugo, soit Nginx pour Briac. Briac a rencontré des problèmes d'autorisations une fois le serveur lancé et ayant déjà perdu pratiquement notre matinée à trouver comment déployer le tout, nous nous sommes rabattus sur la deuxième solution proposée qui était de paramétrer le projet (grâce au fichier Program.cs) pour exposer notre code aux autres. Même si moins sécurisée, elle nous permettait au moins de tester notre interface avec les autres groupes.