

---

# Rapport Projet de Génie Logiciel

Réalisation de la partie back end d'une Application web  
MarvApi

---

**Élèves :**

DURIEUX Romain

MALLARD Tudual

**Professeur :**

Baptiste PESQUET

## Table des matières

<b>I. Introduction</b>	<b>2</b>
<b>II. Spécifications</b>	<b>2</b>
A. Analyse fonctionnelle	2
B. Analyse des besoins pour l'interface avec l'utilisateur	3
C. Schémas UML	3
D. Choix de conception et de production du code source	5
E. Répartition des tâches et planning	6
<b>III. Résultats et tests</b>	<b>7</b>
A. Exigences métier respectées par le programme	7
B. Exigences techniques	8
C. Résultats du programme	9
D- Description des protocoles de tests et résultats	13
<b>IV. Bilan et perspectives</b>	<b>14</b>
A. Points positifs et négatifs	14
B. Evolutions possibles	14

## I. Introduction

Dans le cadre du projet Génie logiciel, nous avons pour but de réaliser la partie back-end d'une application mobile. Pour ce projet, nous avons décidé de créer une application web sur le thème de l'univers MARVEL. En effet, cet univers est très vaste et présente à l'heure actuelle 27 films divisés en 4 phases, tous liés entre eux et il est nécessaire d'en regarder l'intégralité pour comprendre la trame principale. Notre application aura donc pour but de recenser ces films et de caractériser l'avancée de l'utilisateur sur l'univers cinématographique.

## II. Spécifications

### A. Analyse fonctionnelle

---

Notre application a pour objectif de permettre à l'utilisateur de consulter les informations sur les films de l'univers. De plus, l'utilisateur doit pouvoir gérer son avancée sur le visionnage des films. En effet, il doit pouvoir ajouter les films qu'il a vu dans son profil afin de consulter les films qui lui restent à voir pour être à jour sur l'univers cinématographique.

A partir des exigences métiers données en début du projet, nous avons défini plusieurs fonctionnalités qui devaient être présentes pour assurer le bon fonctionnement de l'application. De plus, nous avons choisi de les catégoriser par priorité.

#### **Prioritaires**

- Un utilisateur doit pouvoir grâce à l'application ajouter un film et consulter les informations de ce dernier.
- Un utilisateur doit pouvoir consulter les informations de n'importe quel film de l'univers, qu'il soit vu ou non, et le noter.

### Importants

- Un utilisateur doit pouvoir catégoriser les films en fonction de la phase.
- Un utilisateur doit pouvoir ajouter un film vu dans son avancée
- Un utilisateur doit pouvoir choisir de faire apparaître ou non des éléments avant la génération du CV.

### Optionnels

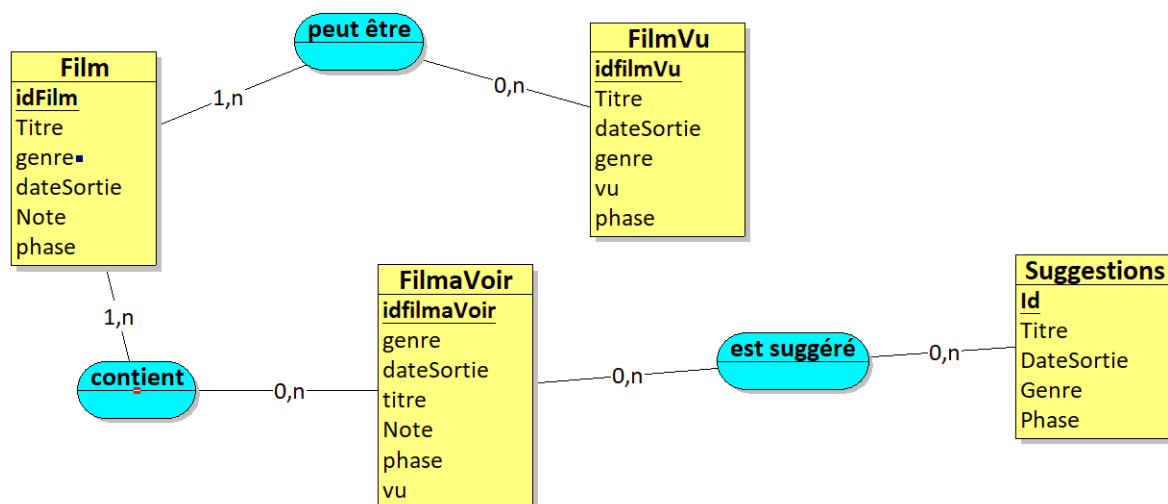
- Un utilisateur doit pouvoir consulter sa progression avec un pourcentage
- Un utilisateur doit pouvoir se voir suggérer un film en fonction de la note qu'il a mis à un film du même genre

## B. Analyse des besoins pour l'interface avec l'utilisateur

Cette application ayant pour but d'être utilisée, nous nous devons qu'elle se doit d'être le plus facilement utilisable et compréhensible. Nous avons donc choisi de créer une application nécessitant un minimum d'actions nécessaires pour chaque objectif afin de rendre l'application la plus agréable et simple possible. En effet, il est attendu qu'elle soit conçue pour être aussi ergonomique et conviviale que possible. De plus, la facilité et la rapidité d'un ajout d'un nouveau film de l'univers dans son avancée était indispensable. Nous avons donc décidé de présenter l'univers sur la page de l'application sous forme de mosaïque d'affiches cliquables des films du MCU.

## C. Schémas UML

### 1) Diagramme des classes



## 2) Appropriation de l'énoncé et modification

En effet en une phrase nous avons :

*“Des films sont classés dans 2 catégories que l'utilisateur peut gérer parmi lesquels il existe une ou plusieurs suggestions de visionnage”.*

Ainsi notre application permet ainsi à un utilisateur de gérer l'état d'un film organisé en phases comportant chacun

Les différentes relations entre les classes peuvent être schématisées de la manière suivante (Cf schéma coloré)mettant bien en avant les relations entre les différentes classes métiers.



### 3) Diagramme de séquence

Ce diagramme de séquence représente la connexion de l'utilisateur à l'application. Lorsqu'il souhaite ouvrir un CV, il sélectionne un CV connu par son titre que le module **App** transmet au module **DAL**. Ensuite, **DAL** récupère toutes les informations composant ce film. Et le trajet retour est effectué pour que l'utilisateur puisse finalement avoir accès au film qu'il a vu avec les éléments qui le caractérisent.

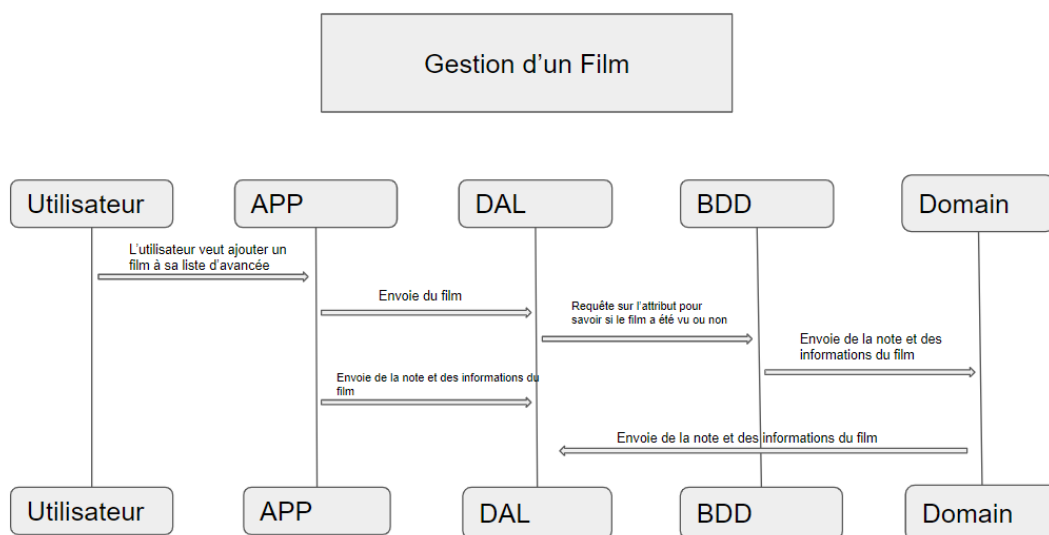


Diagramme de séquence d'utilisation de l'api

## D. Choix de conception et de production du code source

Durant ce projet nous avons fait au mieux pour respecter les différents principes connus. En effet, dès lors que le code n'était pas trop complexe, nous avons appliqué la méthode DRY. De plus, en respectant le concept KISS nous avons codé de la façon la plus simple possible avec les compétences que nous avons. En effet nous avons tenté de factoriser notre code et de le commenter. Nous avons bien entendu respecté la norme

camelCase. Enfin utiliser des interfaces nous a permis de mettre en place des injections de dépendance que nous n'avions jusqu'alors jamais utilisés.

## E. Répartition des tâches et planning

---

Pour mener à bien le projet, nous avons profité au début des séances de TD avec les différents intervenants pour cadrer le projet et organiser notre application.

Par la suite, nous avons réfléchi ensemble aux différentes tâches données et avons repris et réorganisé le trello pour bien définir nos objectifs. Le début du projet a débuté avant les congés de Noël : par conséquent, l'avancée s'est effectuée à distance. Nous avons utilisé l'outil collaboratif GITHUB pour le partage des morceaux du code source effectué par chaque membre de l'équipe. Dès qu'un membre du binôme effectuait une modification, il réalisait un *commit* en l'informant à l'autre membre afin qu'il *pull* la modification. Dès lors, nous avons utilisé cette méthode pour toutes les phases de code à distance. A partir de la rentrée, nous avons en permanence travaillé en binôme, à la manière d'un pair programming en effectuant un *commit* sur notre repository chaque modification. Notre lecteur pourra ainsi consulter les *commit* qui ont été réalisés pendant les phases de travail du projet.

En ce qui concerne la répartition des tâches, nous n'avons pas utilisé d'outil collaboratif. Nous nous disions simplement le travail que le membre du binôme devait réaliser tâche par tâche. Cette répartition a convenu parfaitement et nous avons ainsi pu réaliser l'API.

De plus, voici le planning de notre projet :

	29/11-05/12							06/12-12/12							13/12-19/12							20/12-26/12							27/12-02/01							03/01-09/01							10/01-16/01						
	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D							
I- Lancement du projet																																																	
Choix du projet et définition des fonctionnalités																																																	
Création du diagramme de classe																																																	
II- Réalisation du projet																																																	
1. Gestion de la page d'accueil																																																	
Ajout des différents films																																																	
Création de la mosaïque sur la page d'accueil																																																	
Redirection des images vers la page détails du film correspondant																																																	
2. Gestion des pages personnelles																																																	
Répertoire des films vus																																																	
Répertoire des films à voir																																																	
Gestion de la relation entre les deux pages																																																	
CSS																																																	
3. Tests																																																	
III- Phase finale																																																	
Test final																																																	
Rédaction du rapport																																																	
Rendu																																																	

Planning d'avancée du projet

## III. Résultats et tests

### A. Exigences métier respectées par le programme

Priorité	Description	Etat
Prioritaire	Un utilisateur doit pouvoir grâce à l'application ajouter un film et consulter les informations de ce dernier.	Respectée
Prioritaire	Un utilisateur doit pouvoir ajouter un film vu dans son avancée	Respectée



Prioritaire	Un utilisateur doit pouvoir consulter les informations de n'importe quel film de l'univers, qu'il soit vu ou non, et le noter.	Respectée
Important	Un utilisateur doit pouvoir catégoriser les films en fonction de la phase.	Respectée
Important	Un utilisateur doit pouvoir ajouter un film vu dans son avancée	Respectée
Optionnel	Un utilisateur doit pouvoir consulter sa progression avec un pourcentage	Pas réalisée
Optionnel	Un utilisateur doit pouvoir se voir suggérer un film en fonction de la note qu'il a mis à un film du même genre	A améliorer

## B. Exigences techniques

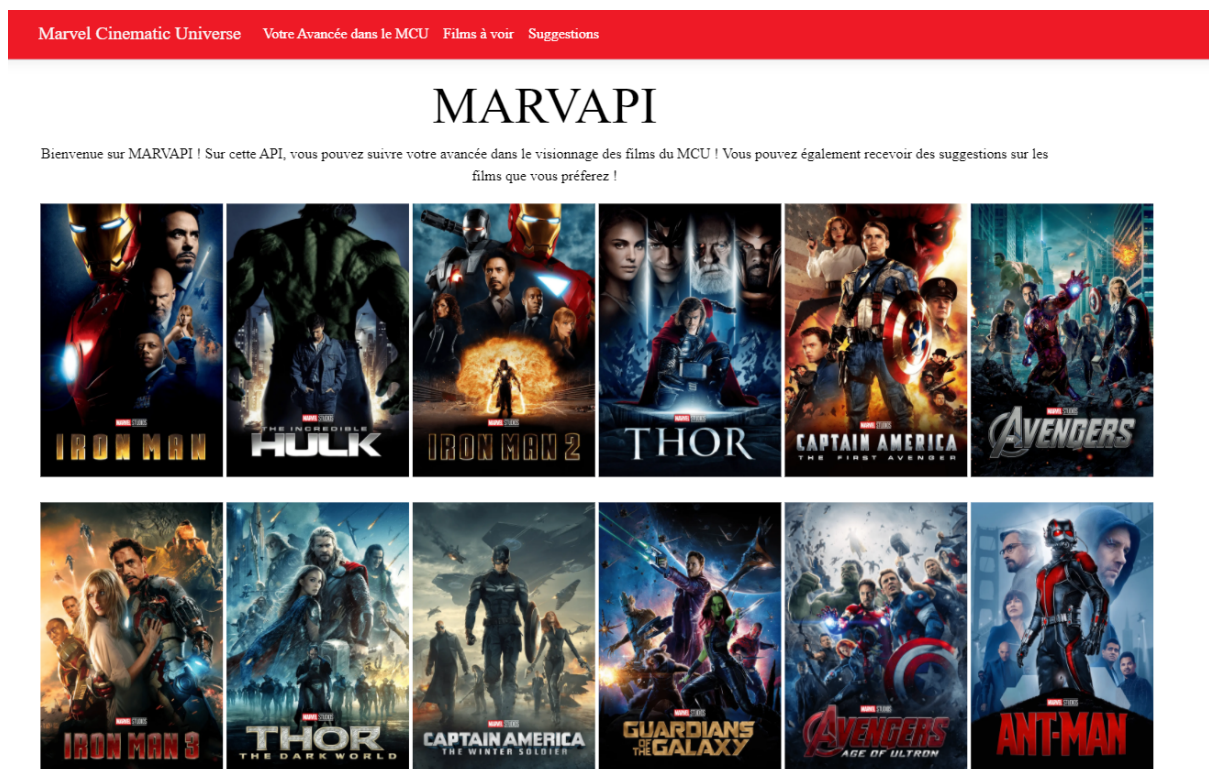
Code	Description
ET_01	L'application est réalisée à l'aide de la technologie ASP.NET Core MVC.
ET_02	Les données persistantes sont stockées dans une base de données relationnelle SQLite.
ET_03	L'interface entre les classes métier et la SGBDR exploite l'outil Entity Framework Core.
ET_04	L'application respecte autant que possible les grands principes de conception étudiés en cours : séparation des responsabilités, limitation de la duplication de code, KISS, YAGNI, etc.
ET_05	L'ensemble du code source respecte la convention camelCase.
ET_06	Les noms des classes, propriétés, méthodes, paramètres et variables sont choisis avec soin pour refléter leur rôle.
ET_07 ET_08	L'application dispose de tests automatisés de type "smoke tests" pour vérifier le comportement des contrôleurs

## C. Résultats du programme

Nous allons présenter dans cette partie les différentes fonctionnalités que nous pouvons faire avec notre API.

### 1) Consultation des films de l'univers

Lorsque l'utilisateur lance l'application, il est accueilli sur cette page d'accueil.



Cette page présente donc au niveau de la barre de navigation les différentes fonctionnalités que l'on propose à l'utilisateur. En effet, il peut donc consulter la liste des films du MCU qu'il a vu, les films qui lui restent à voir, ainsi que les suggestions en fonction des notes qu'il a attribuées aux films qu'il a vu.

Sur la page d'accueil, l'utilisateur se voit présenter les affiches de tous les films MARVEL (images libres de droit) qui sont sortis jusqu'à aujourd'hui. Ces affiches sont cliquables et

renvoient vers la page de renseignements du film en question. L'image suivante présente le résultat lorsqu'on clique sur un film.

[Marvel Cinematic Universe](#)
[Votre Avancée dans le MCU](#)
[Films à voir](#)
[Suggestions](#)

## Détails

Film

Titre	Captain America : First Avenger
Date de sortie	27/08/2011
Genre	Action Aventure
Phase	1
Resume	Captain America: First Avenger nous plonge dans les premières années de l'univers Marvel. Steve Rogers, frère et timide, se porte volontaire pour participer à un programme expérimental qui va le transformer en un Super Soldat connu sous le nom de Captain America. Allié à Bucky Barnes et Peggy Carter, il sera confronté à la diabolique organisation HYDRA dirigée par le redoutable Red Skull.

[Editer](#) | [Retour à la liste](#)

Il a également la possibilité d'éditer les informations du film.

## 2) Consultation des films vus

De plus, l'utilisateur peut consulter la liste des films qu'il a vu, c'est-à-dire caractériser son avancée dans le visionnage de l'univers. Au début, la liste est évidemment vide car nous considérons que l'utilisateur n'a vu aucun film au début.

[Marvel Cinematic Universe](#)
[Votre Avancée dans le MCU](#)
[Films à voir](#)
[Suggestions](#)

## Votre Avancée

Filtrer par phases:  [Filter](#)

Titre	Date de sortie	Genre	Note	Phase	Vu
-------	----------------	-------	------	-------	----

### 3) Sélection d'un film et ajout

L'image suivante présente donc les films qui restent à voir à l'utilisateur. En début de manipulation, l'utilisateur a donc dans sa liste de films à voir les 27 films du MCU. Il peut ainsi consulter les détails des films et les ajouter à son avancée. En effet, une fois le film vu, il peut l'ajouter dans la liste de film vus et le noter.

Marvel Cinematic Universe					
<a href="#">Votre Avancée dans le MCU</a> <a href="#">Films à voir</a> <a href="#">Suggestions</a>					
Films qui vous restent à voir					
Filtrer par phases <input type="text"/> <input type="button" value="Filter"/>					
Titre	Date de sortie	Genre	Phase	Vu	
Iron Man	30/04/2008	Action Aventure	1	0	<a href="#">Ajouter à votre avancée</a>   <a href="#">Details</a>
Shang-Chi et la légende des Dix Anneaux	01/09/2021	Action Aventure	4	0	<a href="#">Ajouter à votre avancée</a>   <a href="#">Details</a>
Black Widow	07/07/2021	Action Aventure	4	0	<a href="#">Ajouter à votre avancée</a>   <a href="#">Details</a>
Spider-Man : Far From Home	03/07/2019	Action Aventure	3	0	<a href="#">Ajouter à votre avancée</a>   <a href="#">Details</a>
Avengers : Endgame	24/04/2019	Action Aventure	3	0	<a href="#">Ajouter à votre avancée</a>   <a href="#">Details</a>
Captain Marvel	06/03/2019	Action Aventure	3	0	<a href="#">Ajouter à votre avancée</a>   <a href="#">Details</a>
Ant-Man et la Guêpe	18/07/2018	Action Aventure	3	0	<a href="#">Ajouter à votre avancée</a>   <a href="#">Details</a>
Avengers : Infinity War	25/04/2018	Action Aventure	3	0	<a href="#">Ajouter à votre avancée</a>   <a href="#">Details</a>
Black Panther	14/02/2018	Action Aventure	3	0	<a href="#">Ajouter à votre avancée</a>   <a href="#">Details</a>
Thor : Ragnarok	25/08/2017	Action Aventure	3	0	<a href="#">Ajouter à votre avancée</a>   <a href="#">Details</a>

Lorsque l'utilisateur clique sur "Ajouter à votre avancée", il est redirigé vers la page ci-dessous. Il doit donc noter le film et entrer la valeur 1 pour l'ajouter à sa liste. Par exemple, l'utilisateur a vu le film Iron Man et veut l'ajouter à sa liste en lui mettant un 7/10.

Marvel Cinematic Universe   Votre Avancée dans le MCU   Films à voir   Suggestions

Comment avez-vous trouvé ce film ? Mettez 1 dans "Vu" pour l'ajouter à votre avancée.

Note

7

Vu

1

Ajouter

Retour

#### 4) Gestion de l'avancée

Le film Iron Man s'est donc bien ajouté à sa liste de films vus et les informations qui y figurent. Il peut également modifier la note qu'il a mis au film, consulter les détails du film ou encore l'effacer de la liste. Si l'utilisateur veut le renvoyer dans la liste des films non vus, il lui suffit de l'éditer et entrer la valeur 0 à la variable "Vu". Bien évidemment, dès lors que l'utilisateur ajoute son film dans son avancée, le film est supprimé de la liste "Films à voir". Ainsi, il peut gérer son avancée dans l'univers jusqu'à avoir vu tous les films.

Marvel Cinematic Universe   Votre Avancée dans le MCU   Films à voir   Suggestions

### Votre Avancée

Filtrer par phases:

Titre	Date de sortie	Genre	Note	Phase	Vu	
Iron Man	30/04/2008	Action Aventure	7	1	1	<a href="#">Editer</a>   <a href="#">Details</a>   <a href="#">Effacer</a>

#### 5) Suggestions

Dans les suggestions, on retrouve tous les films ayant une note supérieure ou égale à 8.

Marvel Cinematic Universe Votre Avancée dans le MCU Films à voir Suggestions

## Suggestions

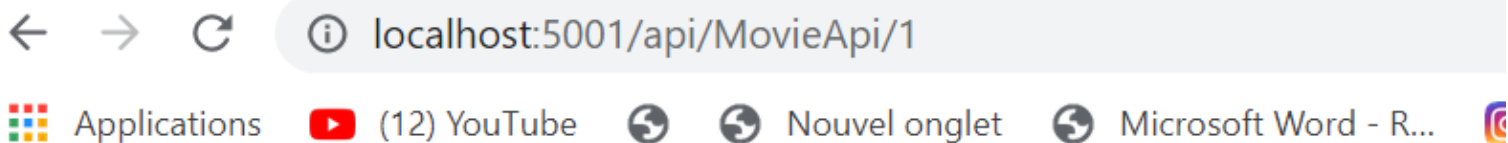
Retrouvez ici les films que vous seriez susceptibles d'aimer.

Titre	Date de sortie	Genre	Note	Phase	
Shang-Chi et la légende des Dix Anneaux	01/09/2021	Action Aventure	8	4	<a href="#">Details</a>
Black Widow	07/07/2021	Action Aventure	8	4	<a href="#">Details</a>
Avengers : Endgame	24/04/2019	Action Aventure	8	3	<a href="#">Details</a>

## D- Description des protocoles de tests et résultats

### 1) API

Le point d'accès `api/MovieApi/1` renvoie les informations sur le Movie identifié par {1} sous la forme d'un objet JSON contenant les champs Id, Titre, Image, Resume, Datedesortie,



```
{"id":1,"titre":"Iron Man","vu":2,"image":null,"resume":null,"datedesortie":"2008-04-30T00:00:00","genre":"Action Aventure","note":9,"phase":"1"}
```

Genre, Note et Phase.

### 2) SmokeTest

Nous avons essayé de réaliser des SmokeTest qui n'ont malheureusement pas abouti. Nous pensons que l'erreur était liée au chemin du test mais nous n'avons au final pas réussi à le modifier pour régler cela.

## IV. Bilan et perspectives

### A. Points positifs et négatifs

---

Ce projet nous a permis de bien comprendre et travailler sur la partie back-end d'une application web, et toutes les parties qui la composent. Nous avons globalement réussi à mettre en place toutes les fonctionnalités que nous voulions insérer à l'origine du projet. Le principal point positif de notre travail est que nous avons réussi à proposer le back-end d'une API web qui est fonctionnelle. Cependant, nous aurions pu évidemment proposer davantage de fonctionnalités qui auraient rendu l'API encore plus complexe.

### B. Evolutions possibles

---

Pour améliorer l'API réalisée, nous souhaiterions développer plusieurs fonctionnalités :

- Proposer des suggestions en fonction des moyennes des notes mises par les utilisateurs
- Ajouter une barre colorée avec un pourcentage de suggestion (rouge si le pourcentage est bas, orange s'il est moyen et vert s'il est bon voire excellent)
- Ajouter une durée totale de visionnage en fonction de l'ensemble des films vus