

# RAPPORT PE :

# Note2Film

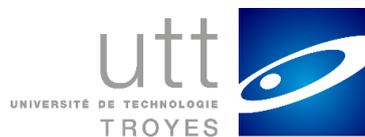
---

**SEMESTRE A22**

---

**Étudiant : Mr HALLEY Louis**

**Tuteure : Mme JARKASS Iman**



---

# Sommaire

Remerciements.....	3
I. Introduction.....	3
II. Concept de l'application.....	3
III. Charte graphique.....	5
IV. Mockup.....	6
V. Technique & développement.....	9
VI. Communication.....	25
VII. Enseignements tirés.....	27
VIII. Conclusion.....	28
Annexes.....	29

---

# Remerciements

J'aimerais prendre le temps de remercier les personnes et organisations clés du projet, sans qui rien n'aurait pu être possible :

- Mme JARKASS, professeure tuteure du projet qui a assuré mon suivi durant le semestre.
- Mr VIAL, premier professeur à avoir accepté de suivre mon projet (n'étant pas spécialisé en ISI ou en RT, il n'a finalement pas pu accompagner le projet).
- L'UTT Net Group, pour m'avoir fourni une licence de développeur Apple.

L'ensemble de la communauté ayant pris le temps d'installer Note2Film et d'effectuer ses retours.

## I. Introduction

Dès mes premiers mois à l'UTT, j'ai envisagé de réaliser un Projet Etudiant. Je trouvais particulièrement passionnant le fait de pouvoir obtenir du temps à consacrer à son projet qui plus est avec le soutien de l'UTT.

J'ai alors commencé à chercher une idée, un concept, qui me motiverait à travailler de nouvelles compétences tout en répondant à un besoin que j'ai personnellement et que d'autres ont peut-être aussi.

C'est alors, que le 17 juillet 2022, en plein stage ST05, m'est venue une idée qui me semblait parfaitement en adéquation avec un Projet Etudiant. Je ne le savais pas encore, mais « Note2Film » était née.

Ce rapport a pour but de retracer mon semestre d'automne 2022 et tout le parcours que j'ai réalisé afin d'arriver au produit final qu'est aujourd'hui cette application.

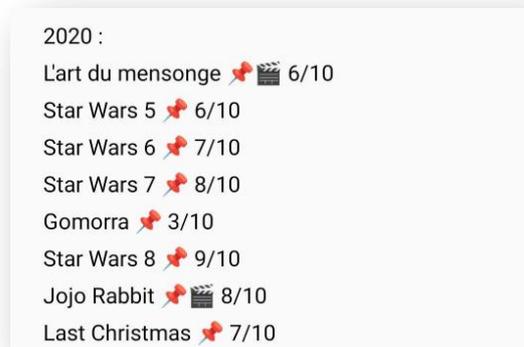
## II. Concept de l'application

Depuis 2018, j'ai pris l'habitude de conserver une trace des films que je regarde (au cinéma et en dehors), ainsi que de leur attribuer une note. Je trouvais ça intéressant d'avoir un souvenir de toutes ces heures passées devant le 7<sup>ème</sup> art.

---

Jusqu'à présent, ce répertoire restait sans vie dans l'application « Notes » de mon téléphone, sous forme d'une liste.

Les informations enregistrées sont sommaires (seule une note ainsi que la présence d'un « Clap » si le film a été vu au cinéma). Les données ne sont pas exploitées, aucune possibilité de recherche ou de tri.



*Figure 1 : un extrait de ces notes.*

Même si j'étais convaincu qu'enregistrer ses films pouvaient avoir une utilité, la manière de procéder n'apportait aucun retour sur ces informations.

L'idée m'est donc venue de créer une application mobile permettant justement d'enregistrer ces films et séries. En passant par l'informatique, ces notes pourraient prendre une toute autre allure : l'aspect de tri serait possible, un plus grand nombre d'informations pourrait être enregistré (parfois même automatiquement) et le rendu final serait bien plus joli qu'une simple liste.

C'est sur cette base que j'ai commencé à imaginer les fonctionnalités présentes dans l'application :

- Possibilité d'enregistrer ses films et séries visionnés, en leur attribuant une note, un commentaire et en spécifiant la date et le lieu de visionnage.
- Possibilité d'enregistrer ses envies futures, les films et séries que l'on aimerait voir afin de pouvoir se balader dans cette liste parallèle les soirs où l'on cherche quelque chose à regarder.
- Un retour sur ses statistiques de visionnage avec par exemple le temps passé au cinéma sur une année précise, le nombre de films vus sur cette même année, etc...

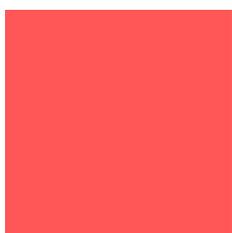
---

## III. Charte graphique

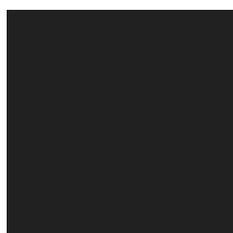
Après quelques semaines de réflexion pendant le mois d'août, je décide de retenir le nom « Note2Film » pour le projet.

Une fois le concept précisé, j'ai pu entrer en quête d'une direction artistique et notamment l'identité visuelle de l'application, qui permettrait à cette dernière d'être reconnue.

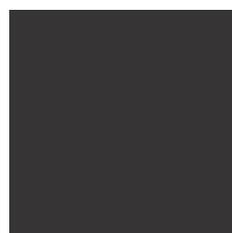
### 1) THÈME & COULEURS



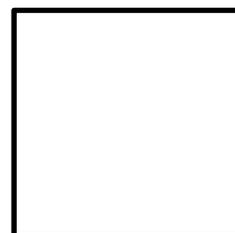
#FF5757



#212121



#373435



#FFFFFF

### 2) LOGOS



### 3) POLICES D'ÉCRITURE

TT Interfaces Black :

**Police de titre**

---

TT Interfaces :

Police de rédaction

## IV. Mockup

Afin de préparer au mieux le projet Note2Film, j'ai décidé d'établir un Mockup avant le début du développement. Le principe de ce dernier est de créer l'aspect visuel de chacune des pages de son application afin de pouvoir se concentrer uniquement sur le développement d'ordre technique par la suite.

J'ai donc débuté la conception de ces pages afin d'avoir une première idée du rendu final.

*Ci-dessous quelques exemples, le Mockup est retrouvable en intégralité sur le site du projet : <https://note2film.louis-halley.fr/#gallery>*

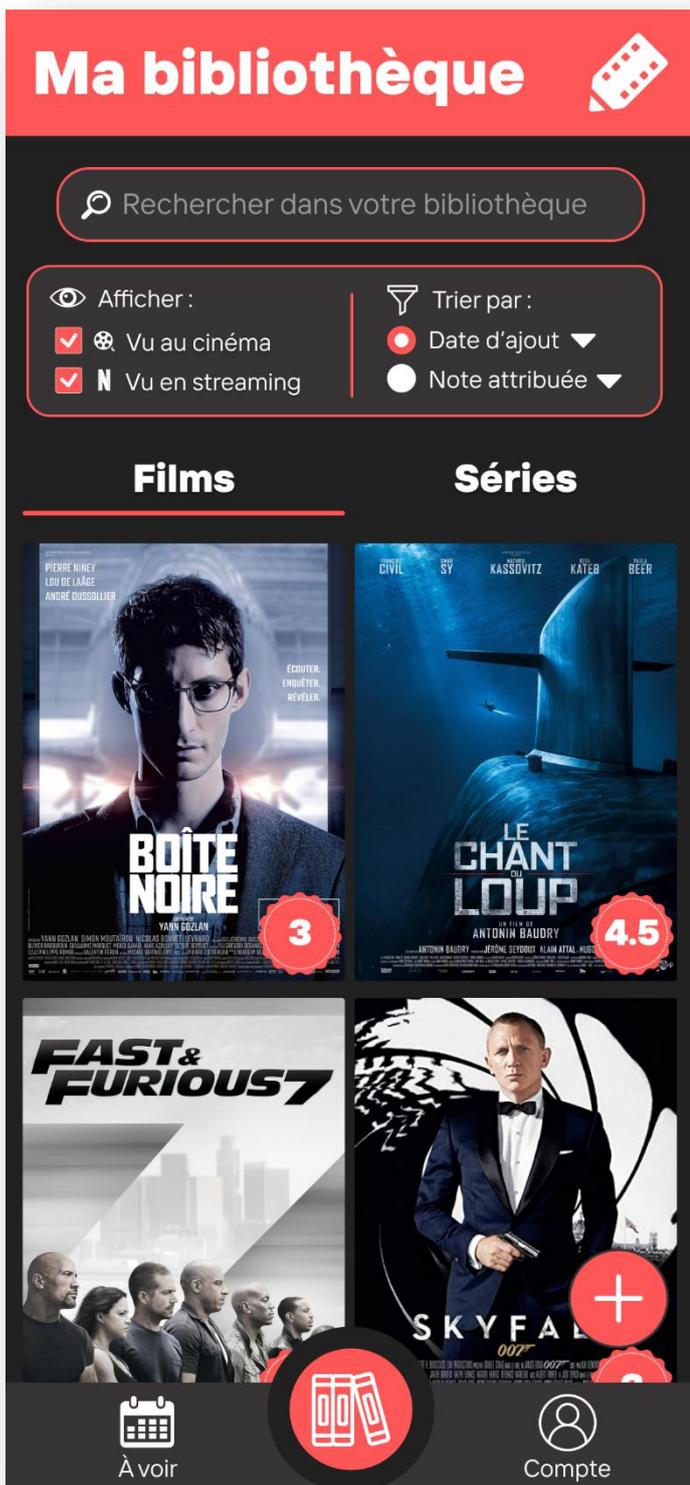


Figure 2 : Mockup de l'écran « Ma bibliothèque ».



Figure 3 : Mockup de l'écran « Détails d'un film enregistré ».

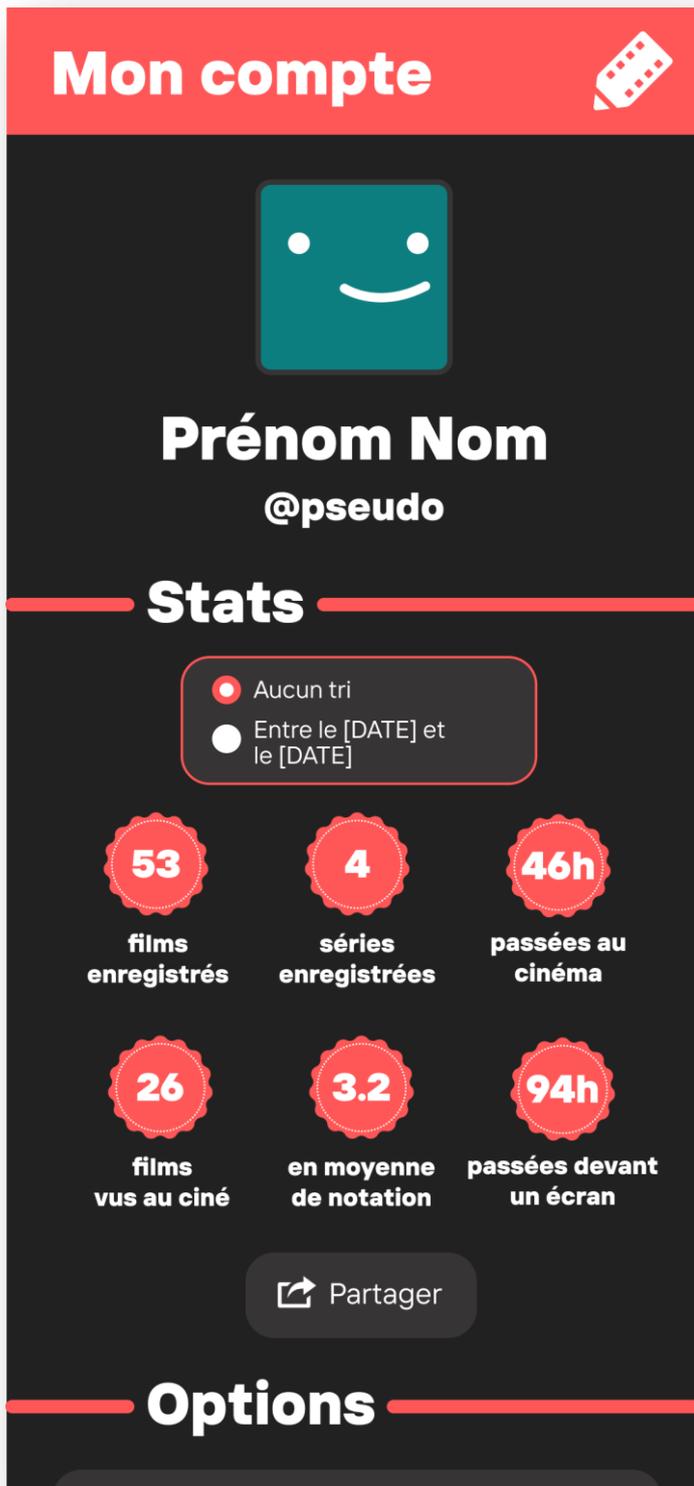


Figure 4 : Mockup de l'écran « Mon compte ».

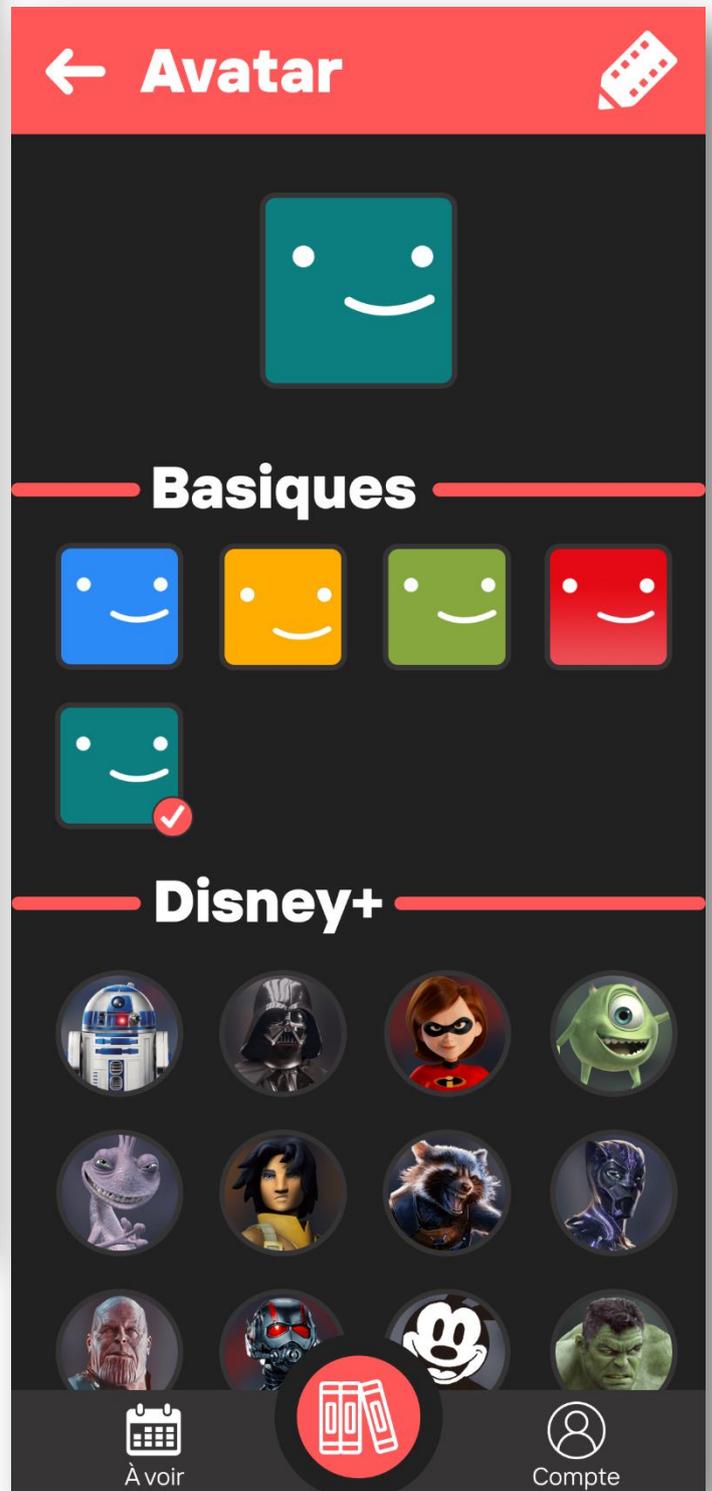


Figure 5 : Mockup de l'écran « Changer d'avatar ».

# V. Technique & développement

## 1) SUPPORT DE DÉVELOPPEMENT (REACT NATIVE)

Concernant la partie développement, j'ai décidé de concevoir ce projet sous forme d'une application mobile. En effet, je n'avais jusqu'alors pas pu approfondir ce domaine pourtant si important dans le monde de demain. Au début des vacances d'été, j'avais pu découvrir le « React Native » en créant ma première application test.

De plus, une application mobile m'est apparue comme permettant une meilleure ergonomie et serait plus appropriée qu'un site Internet ou un logiciel d'ordinateur.

Je suis parti sur un développement basé sur le Framework « React Native », ce dernier utilisant notamment du JavaScript qui est un langage de programmation avec lequel j'ai déjà pu travailler. Ce Framework permet de créer, avec un seul et même code, une application compatible sur Android et iOS. A quelques ajustements près, cela permet donc d'élargir son panel de clients potentiels.

L'un des avantages du « React Native » est son accessibilité. En effet, une énorme communauté gravite autour et propose de nombreuses solutions afin de compléter l'outil et d'y ajouter des extensions. De plus, son utilisation peut se rapprocher du développement Web. Les pages étant créées à base de « div » (figure 6) auxquelles est attribuée une feuille de style (figure 7) afin de les mettre en forme.

```
1 import { Text, View, StatusBar } from 'react-native';
2 import * as React from 'react';
3 import { styles } from '../styles/main-style';
4
5 export const WrongVersionScreen = () => {
6
7   return (
8     <View style={styles.container_erreur}>
9       <StatusBar backgroundColor={'#FF5757'} barStyle={'light-content'} />
10      <View style={styles.viewregular}><Text style={styles.textregular}>Votre application n'est plus à jour. Merci de passer
11        à notre version la plus récente.{'\n\n'}En cas de problème, contactez un développeur.</Text></View>
12    </View>
13  );
14 }
```

---

Figure 6 : exemple d'une page simplifiée de l'application. Dans cette dernière, les lignes 1 à 3 contiennent les dépendances et les lignes 8 à 11 contiennent le code de la page (se rapprochant de l'HTML).

```
1 import { StyleSheet, Text, View } from 'react-native';
2
3 export const styles = StyleSheet.create({
4
5   container_erreur: {
6     flex: 1,
7     backgroundColor: '#FF5757',
8     alignItems: 'center',
9     justifyContent: 'center',
10    color: 'white',
11  },
12
13 });
```

Figure 7 : exemple d'une feuille de style simplifiée (se rapprochant du CSS).

## 2) INFRASTRUCTURE RÉSEAU

L'un des points très intéressants dans Note2Film me semble être son côté réseau. En effet, j'ai décidé de ne pas partir sur une application qui ne fonctionnerait qu'en « local » mais bien connectée à un serveur et un cloud. Cela permettrait d'amener une nouvelle dimension très enrichissante au projet en me demandant de travailler de nouvelles compétences.

Pour comprendre la suite de ce rapport, il est donc important de pouvoir imaginer à quoi ressemble l'infrastructure réseau de Note2Film et les discussions qui ont lieu entre les différents nœuds de cette dernière. C'est ce qui est expliqué dans la figure 8 ci-dessous.

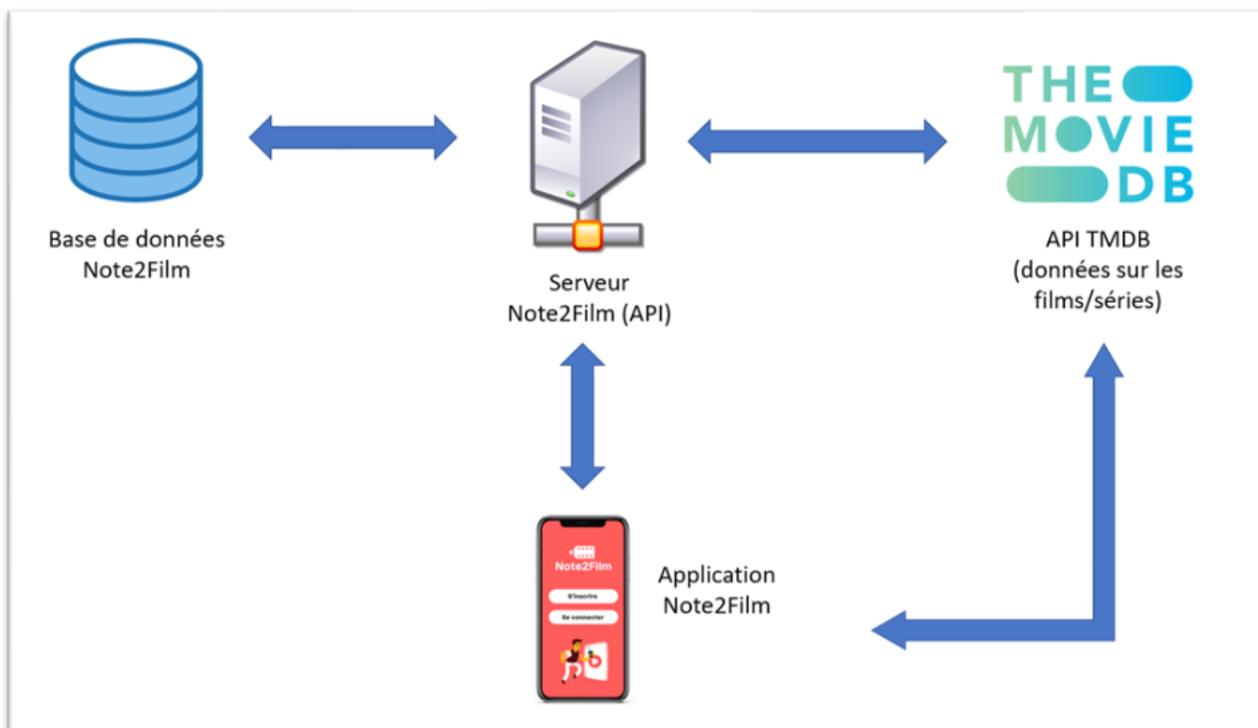


Figure 8 : schéma de l'infrastructure réseau de Note2Film.

Dans ce schéma, vous voyez donc ces différents nœuds qui interviennent dans le processus de l'application. La plupart d'entre eux ont été développés par mes soins (application, serveur, base de données), et d'autres proviennent d'outils préexistants qui permettent la faisabilité du projet (API « The Movie Database »).

Chacune de ces étapes va être expliquée en détails dans la suite de ce rapport.

### 3) API THE MOVIE DATABASE (TMDB)

Afin de concevoir l'application, j'avais nécessairement besoin de données sur les films et séries. En effet, il était inimaginable de demander à l'utilisateur d'entrer manuellement les informations sur le film désiré, sachant que le but même du projet était d'automatiser au maximum la prise de notes.

J'ai donc débuté mes recherches afin d'obtenir un accès à une API ressourçant des données sur les films et les séries telles que leur date de sortie, leur synopsis, leur affiche, etc... Ma première idée était de se tourner vers des sites de rédaction d'articles de cinéma, comme par exemple « AlloCiné » (<https://www.allocine.fr/>), mais ces

---

derniers n'avaient pas, pour la plupart, d'API ou n'en laissaient pas, de manière courante, un accès public.

J'ai néanmoins fini par trouver un des leaders du milieu : « The Movie Database ». Après avoir effectué une demande pour avoir accès à leurs ressources, j'ai pu obtenir une clé d'API.



« The Movie Database » est une base de données communautaire regroupant les films et les séries du monde entier. Le site propose de nombreuses informations sur chacun de ces éléments et notamment des illustrations comme des images de fond qui me seraient très utiles pour embellir l'application.

Comme vu dans la partie précédente, cette API est indispensable et me servira dans divers nœuds du processus. Lors de la recherche du film, de sa sauvegarde, etc... tout devra passer par « The Movie Database ».

#### 4) BASE DE DONNÉES (MONGODB)

Afin de stocker les données des utilisateurs (comptes, informations d'authentification, films/séries, etc...), l'application allait avoir besoin d'une base de données. J'ai donc décidé d'avoir recouru à MongoDB, outil avec lequel j'ai déjà pu travailler par le passé sous différents langages (JavaScript, Python et PHP).

Cette base de données sera agencée en divers sous-dossiers :

- **Stockage des comptes (« accounts »)**

Les données d'un compte regroupent toutes les informations personnelles d'un utilisateur. Elles contiennent :

##### **UID : chaîne de caractères (string)**

Un UID est une chaîne de 32 caractères générés aléatoirement à la création du compte. Il permet de rendre chaque utilisateur unique.

##### **first\_name : chaîne de caractères (string)**

Prénom de l'utilisateur.

---

**last\_name : chaîne de caractères (string)**

Nom de l'utilisateur.

**pseudo : chaîne de caractères (string)**

**email : chaîne de caractères (string)**

**password : chaîne de caractères (string)**

Mot de passe de l'utilisateur stocké de manière sécurisé sous forme d'un Hash (ce point sera abordé en détails dans la partie « Authentification »).

**avatar : entier (int)**

Numéro de l'avatar choisi par l'utilisateur.

**join : entier (int)**

Cette variable correspond à la date d'inscription de l'utilisateur. Elle est stockée sous forme de « timestamp » (nombre de secondes écoulées depuis le 1<sup>er</sup> janvier 1970).

**level : entier (int)**

Niveau du compte, par défaut 0 (utile pour une future version premium/l'utilisation de comptes administrateurs).

```
_id: ObjectId("635e79b6c6450bff254ac561")
uid: "l9vdope334uptdwqv"
first_name: "John"
last_name: "Smith"
pseudo: "lecineaste"
email: "monemail@gmail.com"
password: "$2b$10$22rw1oLOT6DgD2dPri3ozeUdq7.qIokpxToDqGqEAmHIXF9RevXqi"
join: 1667135926
avatar: 75
level: 0
__v: 0
```

Figure 9 : capture d'écran d'un compte stocké dans la base de données.

---

- **Stockage de la bibliothèque (fs\_biblio)**

Le dossier « fs\_biblio » comprend toutes les informations sur les films et séries enregistrés par un utilisateur. Ces derniers sont structurés comme ci-dessous :

**UID : chaîne de caractères (string)**

Compte ayant ajouté le film/série en question.

**tmdb\_id : entier (int)**

ID de l'élément, entier (quasi) unique qui permet son identification sur le site « The Movie Database ».

**type : chaîne de caractères (string)**

Cette variable peut prendre 2 valeurs (« tv » ou « movie ») et correspond à la catégorie de l'élément. Cette information est conservée car les ID de « The Movie Database » (représentés par tmdb\_id) ne sont pas réellement uniques et peuvent se retrouver attribués dans une autre catégorie.

*Exemple : l'ID 19995 correspond au film Avatar et à la série Three Women and A Half.*

**title : chaîne de caractères (string)**

**poster : chaîne de caractères (string)**

**runtime : entier (int)**

**release : chaîne de caractères (string)**

Ces informations sont sauvegardées directement depuis l'API TMDb afin de ne pas avoir à les redemander à chaque fois (cela permet une meilleure optimisation de l'application). Ils correspondent, dans l'ordre, au titre du film/série, à son affiche, à sa durée et à sa date de sortie.

**nb\_views : entier (int)**

Nombre de fois que l'élément a été vu par l'utilisateur (et donc ajouté à sa bibliothèque).

**viewing : tableau d'objets (array of object)**

Tableau contenant tous les visionnages du film/série en question. Chaque objet est structuré comme ci-dessous :

---

**comment : chaîne de caractères (string)**

Commentaire laissé par l'utilisateur.

**mark : entier (int)**

Note (sur 5).

**timestamp : entier (int)**

Date à laquelle l'utilisateur a vu le film/série (sauvegardée sous forme de timestamp).

**place : chaîne de caractères (string)**

Lieu de visionnage (cinéma, plateforme, autre).

**created : entier (int)**

Date à laquelle le film/série est ajouté à la bibliothèque (sauvegardée sous forme de timestamp).

Ces informations sont spécifiques à un visionnage et peuvent donc être présentes à de nombreuses reprises si l'utilisateur a vu plusieurs fois le même film/série.

```
_id: ObjectId("635d32a1e9b2489d4672c3cb")
uid: "19t1epuw1u4m5r161"
tmdb_id: 282035
imdb_id: "null"
type: "movie"
title: "La Momie"
poster: "/uH0cfHEo6i9ovafAlhBdf1MHEt5.jpg"
runtime: 110
release: "2017-06-06"
nb_views: 1
viewing: Array
  0: Object
    comment: ""
    mark: 3
    timestamp: 1609484400
    place: "autre"
    created: 1667052193
    _id: ObjectId("635d32a1e9b2489d4672c3cc")
  __v: 0
```

Figure 10 : capture d'écran d'un film stocké dans la base de données.

Ainsi, la structuration d'une base de données nécessite une habile réflexion, que ce soit en termes d'organisation et d'optimisation. Elle est indispensable à de nombreuses échelles, comme par exemple lors de l'authentification que j'aurai l'occasion d'évoquer plus tard dans ce rapport.

## 5) SERVEUR NOTE2FILM (API)

---

La seconde étape la plus importante après le développement de l'application mobile est celle du serveur.

Contrairement à ce que l'on pourrait penser, il est impossible de faire discuter l'application mobile Note2Film ainsi que la base de données en direct. En théorie, tout ce qui est présent dans l'application Note2Film est modifiable par l'utilisateur lui-même. Cela signifie que si c'était l'application qui avait accès à la base de données, n'importe quel utilisateur pourrait s'y connecter, la consulter, et y enregistrer ce qu'il veut sans passer par des contrôles. Cela poserait donc évidemment des soucis de sécurité et de fiabilité.

Il faut donc un intermédiaire, qui se charge de vérifier les demandes de l'utilisateur et d'en donner suite si nécessaire.

Durant le semestre, j'ai donc dû développer, parallèlement à l'application, un serveur programmé lui-aussi en JavaScript.

Ce dernier utilise le module Express (<https://expressjs.com/>) qui permet la circulation de requête entre un « backend » (le serveur) et un « frontend » (l'application).

Par exemple, si l'application consulte l'URL suivante : <https://api-n2f.louis-halley.fr/version>, le serveur lui retourne la version actuelle de l'application et les versions acceptées (figure 11).



```
JSON  Données brutes  En-têtes
Enregistrer Copier Tout réduire Tout développer Filtre le JSON
success: true
version: "1.0.2"
versions_ok:
  0: "1.0.2"
versions_ok_ios:
  0: "1.0.2"
```

Figure 11 : réponse du serveur Note2Film quant aux versions acceptées.

Ces requêtes peuvent être de 2 types :

- Requête GET : l'application consulte le serveur et attend une réponse de sa part afin d'obtenir une information (par exemple la version).
- Requête POST : l'application envoie des informations au serveur et attend une réponse de sa part.

Les requêtes POST sont les plus utilisées dans Note2Film. Elles permettent de créer un véritable échange entre l'application et le serveur. Pour prendre un exemple concret, si un utilisateur souhaite sauvegarder un film dans sa bibliothèque, le serveur doit contrôler l'existence du film, sa validité ainsi que la bonne et due forme des informations données comme le lieu, la date de visionnage, etc...

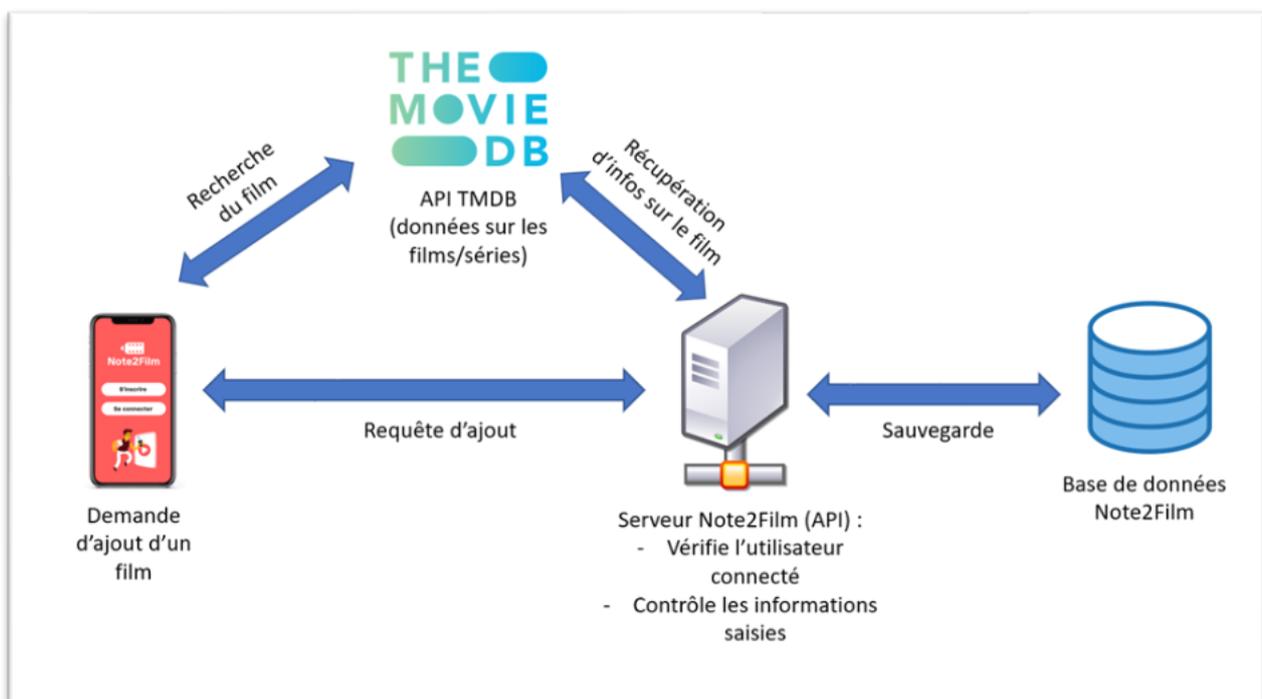


Figure 12 : exemple d'une requête d'ajout de film dans la bibliothèque d'un utilisateur.

Au total, le serveur de Note2Film doit être disposé à exécuter environ 20 requêtes, auxquelles s'ajoutent toutes les vérifications des données envoyées par l'utilisateur.

## 6) AUTHENTIFICATION

Au début du projet, je me suis demandé s'il était préférable que les données de films et de séries soient stockées directement sur le téléphone de l'utilisateur ou sur une base de données. Très vite, j'ai compris qu'il était plus intéressant de travailler à travers un

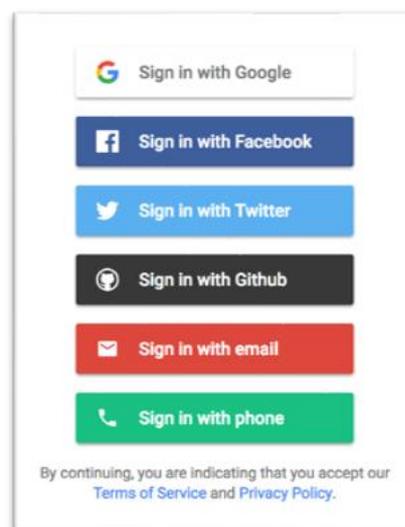
---

« Cloud » et que ce serait une meilleure solution pour les utilisateurs (changement de téléphone, perte des données, ...).

Même si ce choix serait plus enrichissant en termes de découvertes, il apportait aussi son lot de contraintes. Ainsi, établir un système d'authentification, pour que chaque utilisateur ait accès à son espace sur la base de données, devenait obligatoire.

Aujourd'hui, de nombreux systèmes clés en main existent afin de faciliter au maximum cette étape qui peut s'avérer longue et fastidieuse.

Ci-contre, l'exemple de « Firebase Authentication », un outil développé par Google permettant à l'utilisateur d'utiliser un de ses comptes déjà existant pour l'identifier.



L'inconvénient de ce genre d'outils est que j'aurais du apprendre à utiliser l'outil en lui-même plutôt que la technique qu'il y a derrière. C'est pour cette raison que j'ai préféré créer mon propre système d'authentification. Ainsi, lors d'une inscription par exemple, l'utilisateur renseigne ses informations personnelles qui sont ensuite communiquées au serveur. Ce dernier se charge de les vérifier et surtout de sécuriser le mot de passe avant de le sauvegarder dans la base de données.

- **Sécurisation du mot de passe**

De nos jours, laisser les mots de passe « en clair » dans la base de données représente une faille de sécurité critique. C'est pour cela que ces derniers sont chiffrés par le biais de formules mathématiques permettant d'obtenir un « hash » (figure 13).

## Fonction de hachage cryptographique en action

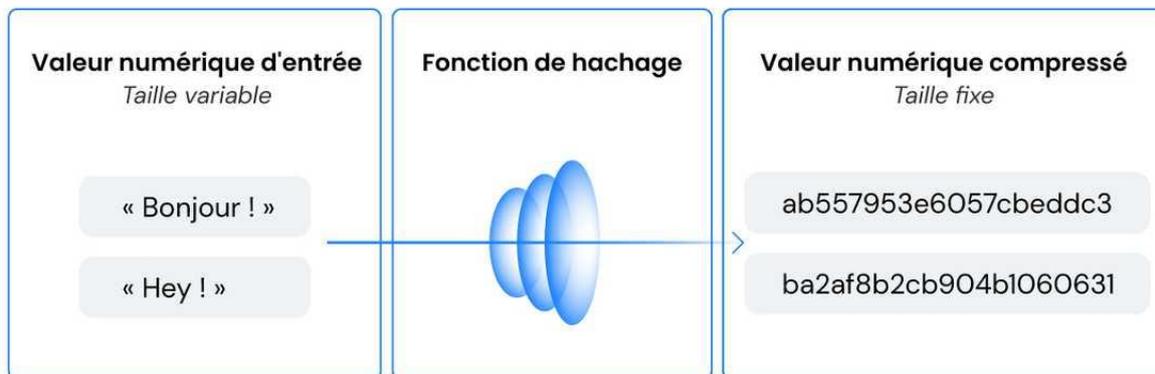


Figure 13 : explication du hachage cryptographique.

Cette technologie permettant de sécuriser une chaîne de caractères repose sur le fait que le processus inverse est en théorie impossible. En cas d'infiltration dans la base de données, le pirate n'a alors accès qu'à ce « hash » ne lui permettant pas de se connecter au compte de l'utilisateur.

*A contrario*, lorsque l'utilisateur tentera de se connecter, le mot de passe entré sera lui-même passé dans la fonction de hachage et il suffira alors de comparer les « hashes » et de vérifier qu'ils soient identiques.

- **Jeton d'authentification (aussi appelé « token »)**

Maintenant que le mot de passe est sécurisé, un autre problème se pose devant nous ; il faut trouver un moyen de conserver l'identification d'un utilisateur entre les différentes requêtes. En effet, il est inconcevable de demander à ce dernier d'entrer son mot de passe à chaque fois qu'il tente de réaliser une action sur son compte.

C'est là qu'entre en jeu les jetons d'authentification. Lors de la première connexion de l'utilisateur, une chaîne de caractères est générée aléatoirement et est stockée à 2 endroits : sur le serveur et sur le téléphone de l'utilisateur vérifié. Ainsi, lorsque l'utilisateur tentera d'effectuer une requête, le jeton stocké dans son téléphone sera communiqué au serveur qui le comparera avec celui stocké dans ce dernier. Si ces derniers correspondent, le serveur peut alors exécuter la requête demandée.

```
_id: ObjectId("636779ae389466a3599c816a")
uid: "la54u6x61siyy3mu1"
token: "78e9a57c90646fe4a2cf09913de63f8caef02"
timestamp: 1667725742
__v: 0
```

Figure 14 : capture d'écran (en partie) d'un jeton d'authentification stocké dans la base de données.

## 7) SYSTÈME DE MISE À JOUR

Un autre point fondamental de l'application, est la conception de son système de mise à jour. En effet, contrairement à un site Internet où tout le code est placé sur un serveur et peut donc être modifié régulièrement, une application mobile nécessite d'être installée et exécutée localement. Cela signifie, que sans un bon système de mise à jour, rien ne préviendrait l'utilisateur ou ne l'encouragerait à faire la dernière mise à jour publiée.

Cette vérification, ayant lieu au lancement de l'application, a évolué au fur et à mesure des versions :

- Première variante (1.0.0) : l'application demande au serveur la version actuelle et si cette dernière n'est pas égale à celle de l'application, elle se bloque et l'encourage à effectuer la mise à jour.
- Seconde variante (1.0.1) : l'application demande au serveur la liste des versions acceptées (renvoyées sous forme de tableau) et l'application vérifie que sa version actuelle est au minimum incluse dans ce tableau.
- Dernière variante (1.0.2) : en fonction de l'environnement où est utilisée l'application (iOS ou Android), cette dernière demande les versions acceptées lui correspondant. Cela permet de ne publier une nouvelle version que pour un de ces 2 supports, en cas de bug isolé.

## 8) PUBLICATION SUR LES MAGASINS D'APPLICATION (STORES)

Une fois la première version terminée, la nouvelle étape consistait à publier l'application sur les Stores. Aujourd'hui, 2 leaders existent dans ce domaine :

- Le Google Play Store, permettant de diffuser son application sur les appareils de type Android. Après l'achat d'une licence à 20€, l'application est contrôlée et publiée.
- L'Apple Store, permettant de diffuser son application sur les appareils de type iPhone. La licence coûte 100€/an et sur ces appareils aucun autre moyen d'installation n'est possible.

Après avoir acheté la licence de développeur Google Play Store, j'ai pu soumettre la première version de l'application.

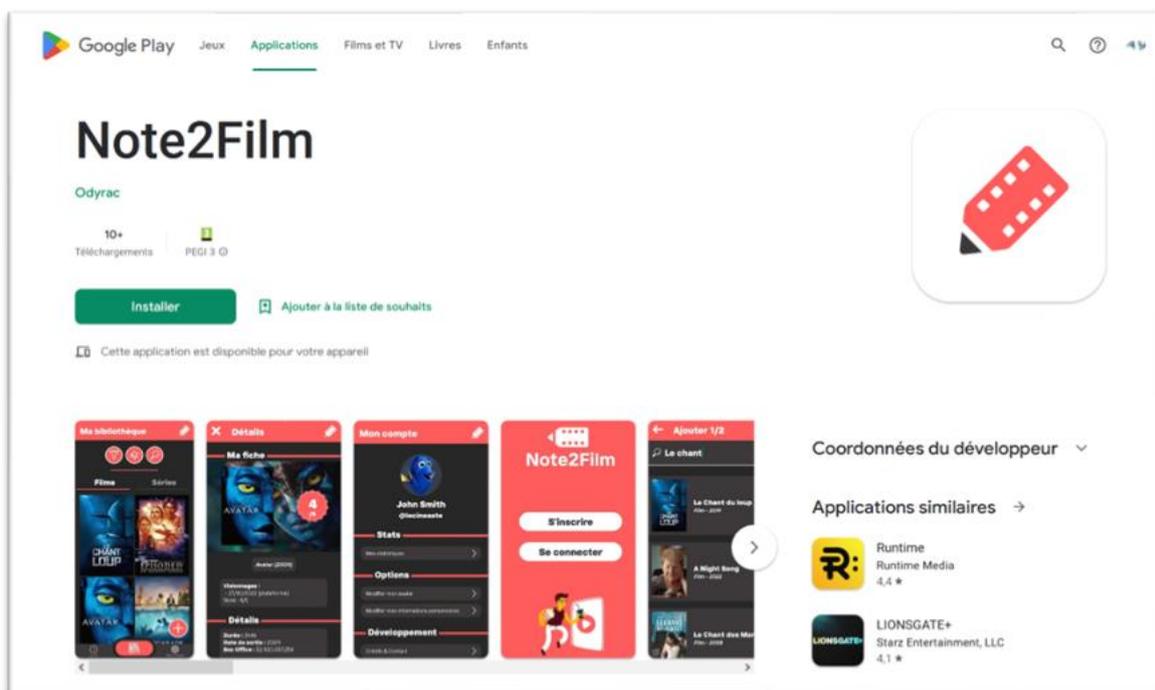


Figure 15 : fiche du store complétée avant la sortie de l'application.

Après plusieurs jours de vérification, cette première version a finalement pu être publiée.

Concernant les iPhones et l'Apple Store, le coût de son utilisation a empêché sa sortie durant longtemps. Heureusement, grâce au soutien de l'UTT Net Group (UNG), la seconde version de l'application a fini par y être publiée. Ces derniers possédant déjà une licence pour les besoins de l'UTT, ils m'y ont gentiment laissé l'accès.



## 9) ORGANISATION ET BUGS RENCONTRÉS

Durant le développement de l'application et du serveur, ce sont plusieurs dizaines voire centaines de bugs qui m'ont freiné et permis, parfois avec prise de tête, d'en apprendre plus sur le développement du projet.

Afin d'éviter de se retrouver noyé sous le travail, une bonne organisation était primordiale. C'est pour cela que très tôt dans le projet, j'ai décidé d'adopter l'outil « Trello » (<https://trello.com/>). Ce dernier fournit un tableau en ligne permettant de ranger toutes les tâches à exécuter.

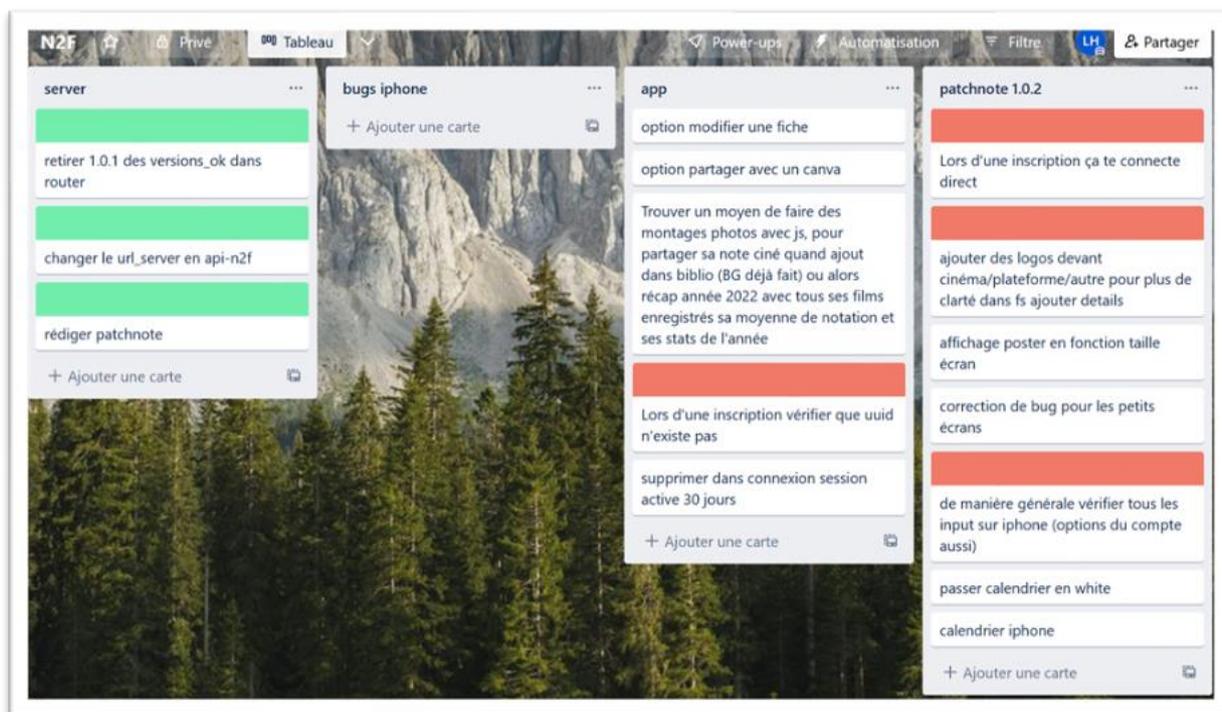


Figure 16 : capture d'écran du Trello de Note2Film.

Cela a permis d'organiser mon travail et, de manière quotidienne, de noter la moindre idée qui me venait en tête.

Parmi ces nombreux bugs, certains se sont fait remarqués :

- **Utilisation de modules non-compatibles**

Même si le React Native offrait un large choix d'outils, il pouvait arriver que j'aie besoin d'en ajouter.

Figure 17 : exemple du module « DateTimePicker » permettant la sélection d'une date (pour choisir le moment du visionnage).

Il fallait alors être très vigilant et vérifier la compatibilité du module en fonction des différentes plateformes. L'exemple s'est d'ailleurs présenté avec le module « DateTimePicker » (figure 17), qui ne fonctionnait pas sur les iPhones lors de la première publication.

Ici, il eut suffi d'une mise à jour afin de corriger le bug, mais dans certains cas, cela a pu être plus problématique.



Ainsi, lors de la phase de développement, un module s'est installé par mégarde au détour d'une mauvaise manipulation. Ce dernier n'était pas bien établi dans le projet. Cela signifie qu'aucune erreur ne s'affichait réellement et que l'application s'exécutait normalement dans l'environnement de développement, mais, dès lors, qu'une version commerciale était créée (celle qui serait publiée sur les Stores), l'application plantait. Sans message d'erreur, c'est près d'une semaine qu'il m'a fallu afin de découvrir le problème, en devant notamment revenir à des sauvegardes précédentes du code afin de finir par isoler le problème.

- **Affichage des images**

Les images sont un élément central de l'application. Présentes à travers les avatars et les affiches de films/séries, un affichage rapide et de qualité est nécessaire. Très vite, j'ai été confronté à des soucis, ces dernières n'apparaissant pas toujours correctement. Après quelques recherches, j'ai fini par comprendre que les images d'une application React Native étaient liées au « cache » de l'application et étaient donc limitées en termes de quantité. C'était très intéressant de pouvoir réfléchir au problème sous le côté matériel et non logiciel du projet. Afin d'y remédier, un système de pagination dans la bibliothèque permet de limiter le nombre d'images sur un même écran (figure 18).

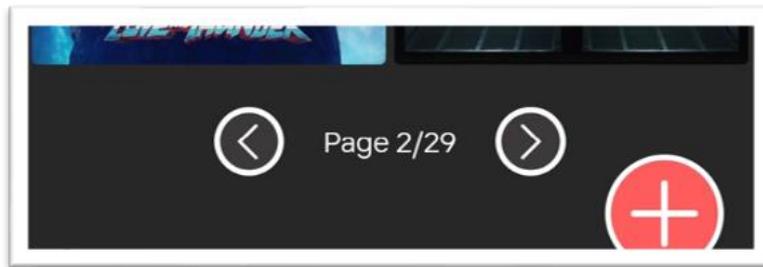


Figure 18 : capture d'écran du système de page, limitant ainsi le nombre de films/séries affichés sur l'écran.

- **Optimisation pour tout type de téléphone**

Publier son application sur autant d'appareils a eu l'avantage de toucher un maximum de monde. Néanmoins, cela signifie aussi que l'application devait être « responsive » et rester fonctionnelle sur des environnements parfois totalement différents.

J'ai donc dû optimiser l'application sous différents critères :

- Taille de l'écran : en fonction des générations de téléphone, les tailles d'écran varient énormément et il faut s'assurer que toutes les zones restent accessibles, que rien ne se chevauche, etc...
- Compatibilité des iPhones : même si le React Native offre la possibilité d'user du même code pour des plateformes différentes, certains ajustements et correctifs sont à déployer. Pour prendre un exemple, lors de la première version publiée sur iPhone, les claviers cachaient les entrées de texte et empêchaient l'utilisateur de voir ce qu'il saisissait.

Ces différents bugs ont pu être corrigés à travers de nombreuses mises à jour publiées à la suite de la première sortie de l'application ; les retours des utilisateurs permettant de les découvrir.

## 10) SITE WEB

Afin de préparer le côté communication (que je détaillerai par la suite), j'ai aussi pu développer un site Internet, permettant de regrouper les informations essentielles sur le projet. Cela a permis d'offrir une synthèse aux personnes s'intéressant à l'aventure Note2Film.

Les 2 sites développés sont :

- Un site Internet de présentation :

<https://note2film.louis-halley.fr/>

- Un site Internet répertoriant les liens importants du projet :

<https://note2film.louis-halley.fr/linktree/>

## VI. Communication

Un aspect important dans le projet était le côté communication. En effet, même si je savais que Note2Film répondrait à l'un de mes besoins personnels, j'étais persuadé qu'il pourrait en convenir de même pour d'autres.

Comme vu précédemment, j'ai d'ailleurs pu développer un site Internet au projet ressourçant ses informations essentielles notamment dans l'optique de le faire connaître.

Très tôt, j'ai pris la décision de promouvoir le projet autour de moi notamment en lui créant un compte Instagram.

Ce dernier a permis de se hisser dans le cercle des élèves de l'UTT et de créer la base d'une communauté qui suivait les dernières nouvelles.

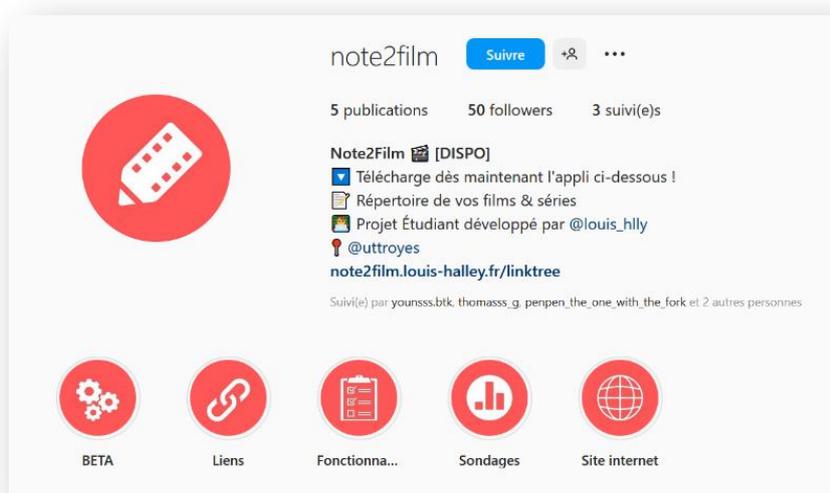


Figure 19 : capture d'écran du compte Instagram @note2film (<https://www.instagram.com/note2film/>).

Chaque « post » était travaillé afin de respecter la charte graphique établie précédemment et conserver au maximum un certain professionnalisme. Ainsi, à travers les « stories » (courtes vidéos verticales), j'ai pu réaliser des sondages mais aussi des démonstrations de certaines fonctionnalités afin de faire monter l'enthousiasme avant la sortie officielle (figure 20).

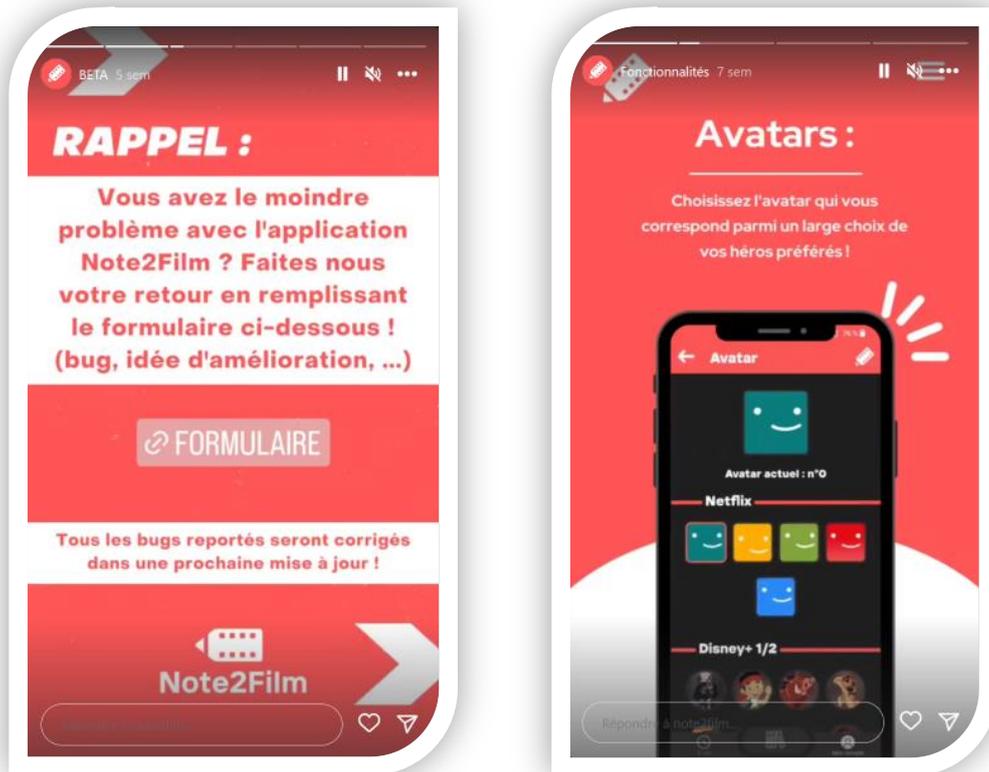


Figure 20 : exemple de « stories » publiées.

Afin de marquer la rupture entre la période de développement et la sortie officielle, j'ai aussi pu travailler sur 2 spots publicitaires mettant en avant les différentes fonctionnalités de l'application. Vous pouvez les retrouver ci-dessous :

<https://www.instagram.com/p/CknRkxgv95i/>

<https://www.instagram.com/p/Clgicj0umsq/>

La vidéo d'annonce de sortie de l'application a ainsi été vue plus de 700 fois !

---

Ce compte Instagram s'est avéré d'autant plus utile au moment de la sortie de l'application. Cela a permis à la communauté de faire ses retours, de prévenir de bugs et de soumettre des idées.

Cet aspect communication fut très enrichissant car il demandait le développement de nouvelles compétences, autre que technique, afin d'assurer au projet une bonne sortie et un nombre d'utilisateurs conséquent.

## VII. Enseignements tirés

Travailler pendant un semestre entier sur ce projet a donc été très enrichissant pour moi.

En effet, cela m'a permis d'en apprendre plus sur différents aspects :

- Technique : j'ai dû approfondir de nombreuses compétences telles que le réseau, l'utilisation d'une base de données, le développement Web, le design d'une charte graphique, etc...
- Organisation : même si je n'avais pas à proprement parler de deadline durant le semestre, j'ai réfléchi très tôt sur les délais que je comptais m'imposer afin de finir le projet dans les temps. Cela était d'autant plus important que certaines périodes seraient plus chargées que d'autres, lors des examens par exemple, et qu'il serait alors essentiel d'avoir pris de l'avance.
- Mentalité personnelle : c'est un souci que j'ai déjà pu remarquer dans mes autres projets mais qui s'est encore plus ressenti ici. Passer autant de temps sur quelque chose qui nous tient à cœur a pour conséquence d'avoir des difficultés à décrocher. Certes la motivation reste intacte mais cela peut s'avérer parfois risqué lorsque d'autres matières requièrent notre attention et que notre esprit reste bloqué sur le projet. Cela est encore accentué lorsque le projet est public et que la communauté effectue ses retours en direct : on veut toujours améliorer ce qu'on leur propose sans jamais les décevoir.

---

## VIII. Conclusion

Je suis extrêmement fier du travail accompli sur le semestre. Aujourd'hui, Note2Film accueille une quarantaine d'utilisateurs et déjà plus de 600 films et séries ont été enregistrés !

Ce projet m'a permis de gagner en expérience mais aussi d'apprendre à me connaître un peu plus. Il m'a aussi offert la possibilité d'échanger avec des personnes ayant la même passion que moi que ce soit celle du cinéma ou de la programmation.

J'ai la conviction certaine que je pourrai continuer à travailler dessus dans les prochains mois afin d'améliorer l'application et de proposer de nouvelles fonctionnalités à la communauté. Les différents retours obtenus jusqu'à présent étant très motivants pour la suite.

---

# Annexes

Figure 1 : extrait de notes.....	4
Figure 2 : Mockup 1/4.....	7
Figure 3 : Mockup 2/4.....	7
Figure 4 : Mockup 3/4.....	8
Figure 5 : Mockup 4/4.....	8
Figure 6 : code d'une page simplifiée.....	9
Figure 7 : feuille de style.....	10
Figure 8 : infrastructure réseau.....	11
Figure 9 : exemple d'un compte.....	13
Figure 10 : exemple d'un film.....	15
Figure 11 : réponse du serveur.....	16
Figure 12 : exemple d'une requête.....	17
Figure 13 : hachage cryptographique.....	19
Figure 14 : jeton d'authentification.....	20
Figure 15 : fiche PlayStore.....	21
Figure 16 : Trello.....	22
Figure 17 : module de date.....	23
Figure 18 : système de pagination.....	24
Figure 19 : compte Instagram.....	25
Figure 20 : exemple de stories.....	26