



RAPPORT DU PROJET RATEFLIX

RateFlix : Critique de film

Équipe n° 9

Albane CARTERON, Xavier DURUMAIN, Jules POTIRON, Eugénie ROCHERON : option Logiciels et Données
Lisa BENVENUTI, Ambre SANTANACH : option Cloud Système et Sécurité

Table des matières

I- Présentation de l'Infrastructure	2
A) srv-web	2
B) srv-bdd	2
C) srv-backup	3
D) srv-r-proxy	3
II- Fonctionnement de l'application WEB	4
III- Conception de l'application Java	6
IV- Conception de la base de données relationnelle	8
V- Architecture du projet	9
VI- Scénario de démonstration	11
A) Application web	11
B) Application Java	12
VII- Présentation de l'organisation	13
VIII- Table des figures	18

I- Présentation de l'Infrastructure

L'infrastructure du projet est constituée de 4 machines virtuelles (VM) sur un Vagrantfile. Chacune d'entre elles possède son adresse IP et ses scripts de provisionnement qui seront expliqués plus tard :

- srv-web (192.168.56.80) : install_sys.sh et install_web.sh.
- srv-bdd (192.168.56.81) : install_sys.sh, install_web.sh, install_bdd.sh et install_myadmin.sh.
- srv-backup (192.168.56.82) : install_sys.sh et get_bdd.sh.
- srv-r-proxy (192.168.56.83) : install_sys.sh, install_web.sh, et install_rproxy.sh.

Les scripts de provisionnement permettent la conception de chaque VM de manière unique. Le script install_sys.sh est essentiel car il permet la mise en place l'architecture de base d'une VM.

A) srv-web

Cette VM héberge le site web mis en place et elle se sert de 2 scripts. Dans install_web.sh on note que l'on récupère les différentes pages du site web dans un dossier et on les copie sur la VM.

```

32 echo "=> [2]: Configuration Apache2"
33 cp -r /vagrant/files/php/* /var/www/html
34
35 service apache2 restart

```

Figure 1 : Extrait du script install_web.sh

B) srv-bdd

Cette VM héberge la base de données et elle se sert de 4 scripts. Le script install_web.sh est nécessaire pour avoir accès à phpMyAdmin. Pour install_bdd.sh, les lignes de codes ci-dessous montrent la définition de variables pour la connexion à phpMyAdmin ainsi que la localisation d'un fichier sql contenant la création des 3 Tables nécessaires au fonctionnement du site web ainsi qu'un échantillon de données pour les Tables avant que la base de données soit étoffée.

```

10 # Utilisateur à créer (si un vide alors pas de création)
11 DBNAME="rateflix"
12 DBUSER="AmbreS"
13 DBPASSWD="network"
14 # Fichier sql à injecter (présent dans un sous répertoire)
15 DBFILE="/vagrant/files/creation_bdd.sql"

```

Figure 2 : Extrait du script install_bdd.sh : Identifiants et chemin du fichier sql

L'image ci-dessous montre plusieurs lignes de code utilisant les variables afin de configurer le service phpMyAdmin. Le pourcentage à la ligne 30 permet d'assurer ces droits à tous les types de serveurs.

```

26 echo "=> [2]: Configuration service"
27 if [ -n "$DBNAME" ] && [ -n "$DBUSER" ] && [ -n "$DBPASSWD" ]; then
28   mysql -e "CREATE DATABASE $DBNAME" \
29   >> $LOG_FILE 2>&1
30   mysql -e "grant all privileges on $DBNAME.* to '$DBUSER'@'%' identified by '$DBPASSWD'" \
31   >> $LOG_FILE 2>&1
32 fi

```

Figure 3 : Extrait du script install_bdd.sh : Configuration de la base de données

La ligne 33 assure la connexion au serveur depuis n'importe quelle adresse IP du réseau.

```

33 sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/mariadb.conf.d/50-server.cnf

```

Figure 4 : Extrait du script install_bdd.sh : connexion au serveur

Les lignes de code ci-dessous mettent à jour la base de données avec le fichier sql.

```

35 echo "=> [3]: Configuration BDD"
36 if [ -f "$DBFILE" ];then
37     mysql -u $DBUSER -p$DBPASSWORD $DBNAME < $DBFILE \
38     >> $LOG_FILE 2>&1
39 fi
  
```

Figure 5 : Extrait du script `install_bdd.sh` : mise à jour base de données

C) srv-backup

Cette VM a pour objectif de récupérer les données contenues sur la base de données. Afin d'assurer son fonctionnement, elle se sert de 2 scripts. Le script `get_bdd.sh` assure la récupération des informations présentes dans la base de données au moment du lancement du script et la sauvegarde du fichier obtenu dans un dossier spécifique.

```

3  ## Récupérer données BDD
4  DBUSER=AmbreS
5  DBPASSWORD=network
6  DBNAME=rateflix
7
8  BACKUP_DIR=/vagrant/files/backup
9  DATE=$(date +"%Y%m%d")
10 BACKUP_FILE=$BACKUP_DIR/backup_$DATE.sql
11
12 mysqldump -u $DBUSER -p$DBPASSWORD $DBNAME > $BACKUP_FILE
  
```

Figure 6 : Extrait du script `install_backup.sh`

D) srv-r-proxy

Cette VM a pour objectif de récupérer les données contenues sur la base de données. Afin d'assurer son fonctionnement, elle se sert de 3 scripts. Le script `install_rproxy.sh` permet la création d'un fichier `reverse_proxy.conf` et la modification du fichier `000-default.conf` dans la VM afin de mettre en place un proxy permettant l'accès au site web (utilisation de l'adresse IP de la VM du site web) à tous les appareils connectés au même réseau que l'ordinateur utilisé dans la salle Fourier. Le site web est alors accessible à l'adresse IP 192.168.4.205 une fois la VM mise en route.

```

16 # Création fichier rproxy.conf
17 sudo mkdir -p /etc/apache2/sites-available
18 sudo touch /etc/apache2/sites-available/reverse-proxy.conf
19
20 # Configuration port HTTP
21 sudo sed -i 's|</VirtualHost>||g' /etc/apache2/sites-available/000-default.conf
22 echo "<VirtualHost *:80>"
23     ServerName .srv-r-proxy
24     # Configuration pour HTTP
25     ProxyRequests Off
26     ProxyPreserveHost On
27
28     <Proxy *>
29     | Require all granted
30     </Proxy>
31
32     ProxyPass / http://192.168.56.80/
33     ProxyPassReverse / http://192.168.56.80/
34 </VirtualHost>" > /etc/apache2/sites-available/000-default.conf
  
```

Figure 7 : Extrait du script `install_rproxy.sh`

II- Fonctionnement de l'application WEB

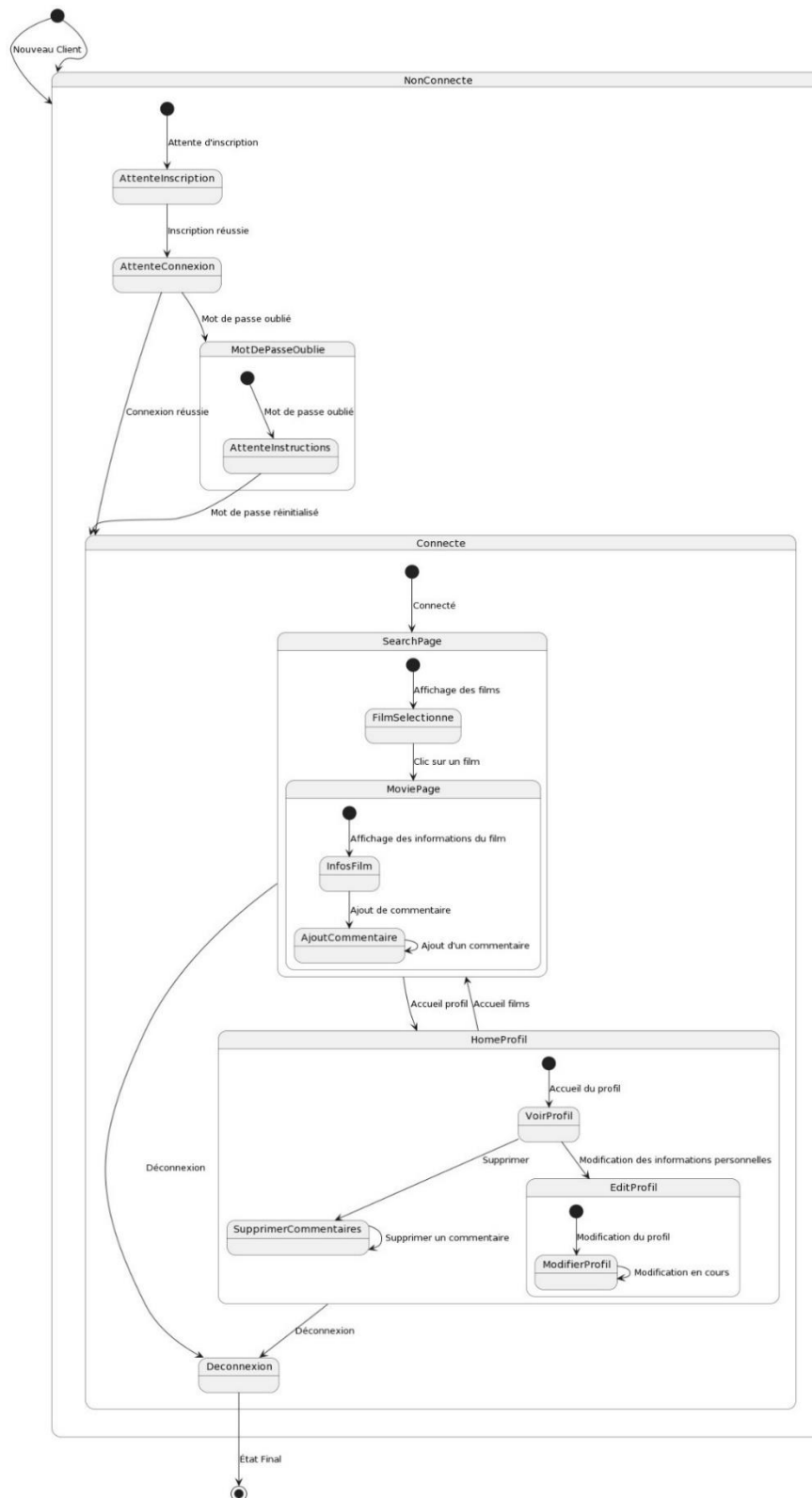


Figure 8 : Diagramme UML d'état-transition de Rateflix

L'application web se compose de sept pages, centrées sur l'expérience utilisateur, accessibles via une fenêtre de connexion.

Une fois la connexion réussie, un en-tête sur chaque page permet de naviguer dans le site. On y retrouve le nom du site RATEFLIX qui permet de revenir sur la page de recherche de film, un bouton "Déconnexion," un avatar et le pseudo pour accéder à la page du profil.

Les pages sont réparties comme suit :

1. Création de compte :

- Register : L'utilisateur remplit un formulaire avec son pseudo, adresse mail, mot de passe (crypté), et date de naissance. Validation par bouton, avec possibilité de se connecter si déjà inscrit.

2. Connexion :

- Login : Formulaire pour pseudo et mot de passe (masqué). Lien "mot de passe oublié" pour accéder à la page correspondante, et "Créer son compte" pour la création.

3. Mot de passe oublié :

- PasswordForgot : L'utilisateur entre son adresse mail, nouveau mot de passe et confirmation. Après confirmation, le mot de passe est mis à jour. Lien de retour vers la page de connexion.

4. Accueil site & recherche de film :

- SearchPage : Affiche tous les films avec une barre de recherche. La recherche instantanée affiche les résultats au fur et à mesure de la saisie. En cliquant sur une affiche, l'utilisateur est redirigé vers la page MoviePage du film sélectionné.

5. Profil du film & ajout de commentaires :

- MoviePage : Informations du film à gauche (affiche, titre, réalisateur, note globale). À droite, possibilité d'ajouter une note et un commentaire lié au film. En bas, les avis d'utilisateurs sans possibilité de modification ou suppression.

6. Accueil Profil & gestion des commentaires :

- HomeProfil : Accueil du profil utilisateur avec barre de navigation (Voir Profil, Modifier infos). Partie droite affiche le profil de l'utilisateur avec photo, pseudo, mail, date de naissance, et les avis qu'il a laissés. Il peut ici supprimer ses avis grâce à un bouton « poubelle ».

7. Modification informations personnelles utilisateurs :

- EditProfil : Page pour modifier les infos personnelles. Barre de navigation similaire à HomeProfil. Possibilité de changer pseudo, email, mot de passe et photo de profil (limite 2Mo). Bouton pour confirmer chaque modification.

III- Conception de l'application Java

L'application Java est destinée uniquement aux administrateurs. Elle leur permet de gérer tous les profils créés sur son site web et de gérer les films accessibles sur son site web.

L'application Java a été architecturée sous forme de 4 packages : vue, item, contrôleur, BDD.

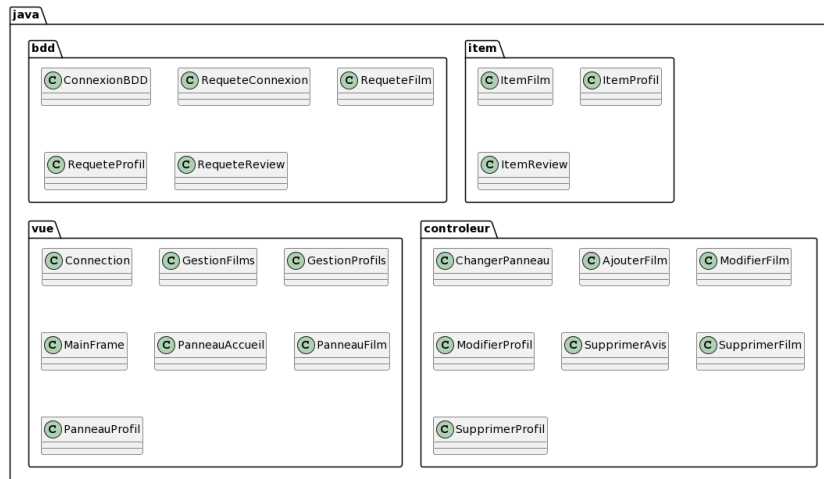


Figure 9 : Diagramme de classe global de l'application Java

Le package « item » regroupe les 3 classes représentant les objets qui seront utilisés dans toutes les autres classes.



Figure 10 : Diagramme de classe du package vue

Le package « vue » constitue la composante dédiée à l'Interface Homme Machine. La classe **MainFrame**, servant de **JFrame** principal, initie le lancement de l'application et affiche successivement tous les panneaux. Il est important de noter que cette classe demeure constamment ouverte, seules les interfaces des panneaux évoluant en fonction des actions entreprises seront modifiées.

Par la suite, six **JPanel** sont présents, avec le panneau principal étant le panneau d'accueil. Celui-ci offre l'accès aux deux fonctionnalités principales de l'application : la gestion des profils et la gestion des films.

Le panneau connexion donne accès à une interface permettant de se connecter à l'application.

Les panneaux de gestion des profils et gestion des films permettent d'afficher respectivement la liste complète des profils ou des films enregistrés sur le site internet. Il offre également la possibilité de rechercher un profil ou film en particulier à partir de son pseudo/ son titre. De plus, ce panneau est le lien direct pour accéder aux détails d'un profil ou un film.

Le panneau dédié aux profils offre une exploration approfondie des données liées à un utilisateur particulier, incluant son pseudonyme, son adresse e-mail, sa date de naissance et son avatar. De plus, la liste des avis émis par ce profil sur les films est accessible. Sur ce même panneau, il est envisageable d'effectuer des modifications ou de supprimer les informations du profil, ainsi que de supprimer chaque avis au besoin.

Quant au panneau Film relatif à un film, il permet une analyse détaillée des informations propres à celui-ci, telles que son titre, son producteur, sa date de sortie, sa description, sa moyenne et sa photo. De plus, la liste des avis associés à ce film est également disponible. Les fonctionnalités de modification ou de suppression des informations du film, ainsi que la suppression individuelle des avis, sont également accessibles sur ce panneau,

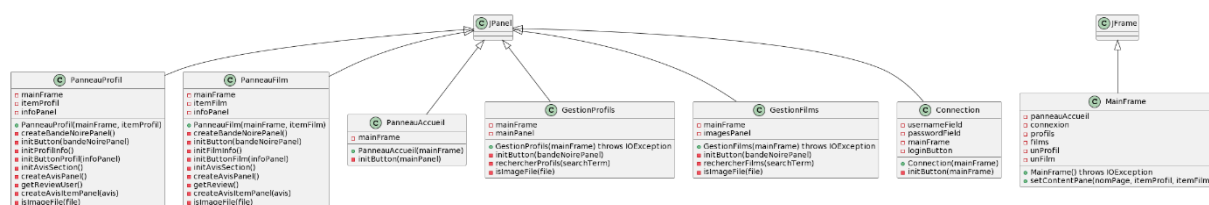


Figure 11 : Diagramme de classe package contrôleur

Le package "bdd" englobe l'ensemble des classes permettant d'interagir avec la base de données. La classe ConnectionBDD facilite la connexion à la base de données, tandis que les autres classes sont dédiées à la gestion des requêtes.

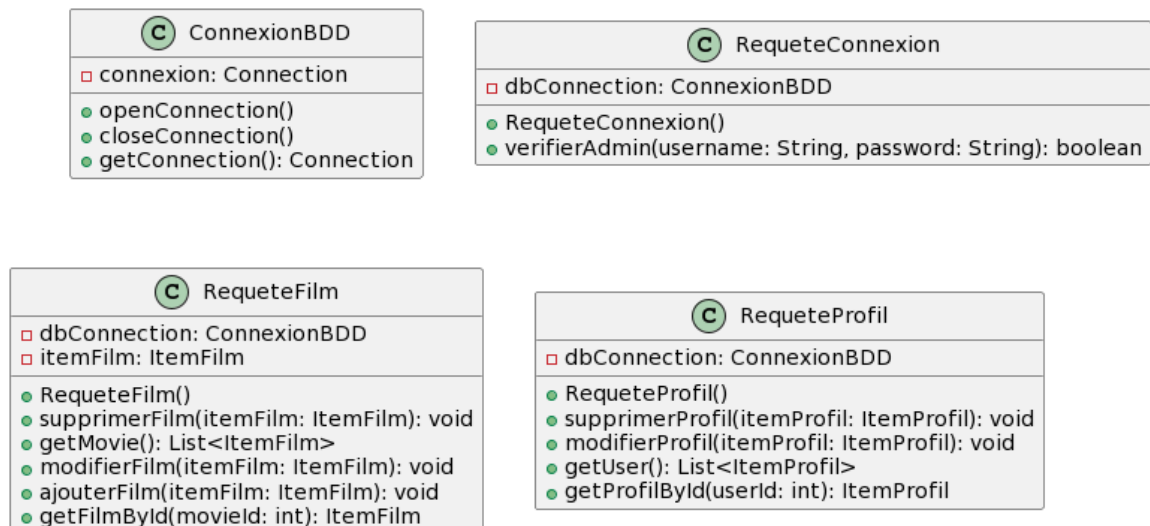


Figure 12 : Diagramme de classe package bdd

IV- Conception de la base de données relationnelle



Figure 13 : Modèle Conceptuel des Données (MCD)

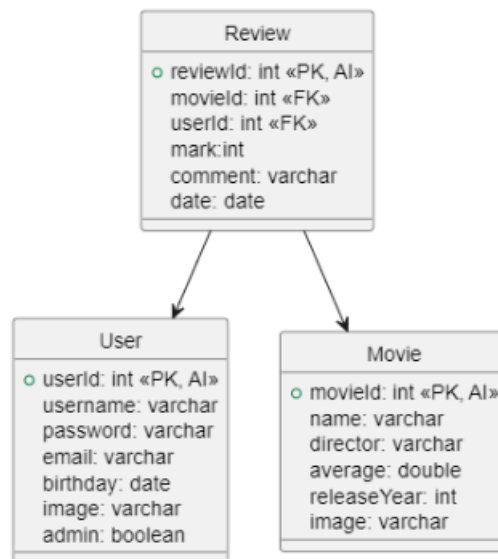


Figure 14 : Modèle Logique des Données (MLD)

Notre base de données est essentielle au bon fonctionnement de nos deux applications et à leur bonne communication.

Elle est conçue pour gérer efficacement les informations liées aux utilisateurs, aux films, et aux avis laissés par les utilisateurs. En se basant sur le MCD et le MLD fournis ci-dessus, voici une description détaillée de chaque table et de leur rôle.

Table "User"

La table "User" stocke les informations des utilisateurs de l'application. Chaque utilisateur est identifié par un ID unique (**userId**). Les informations stockées comprennent le pseudo (**username**), le mot de passe crypté (**password**), l'adresse e-mail (**email**), la date de naissance (**birthday**), le lien vers l'image de profil (**image**), et le statut administrateur (**admin**). Cette table permet de gérer les profils des utilisateurs et de les associer à leurs activités sur le site. La table permet de récupérer les informations personnelles des utilisateurs, facilitant l'identification et la personnalisation de l'expérience utilisateur. La colonne du booléen **admin** permet de déterminer si un utilisateur a des droits d'administration. Si c'est le cas, il peut accéder à l'application bureau.

Table "Movie"

La table "Movie" stocke les détails des films enregistrés sur le site. Chaque film est identifié par un ID unique (**movieId**). Les informations stockées comprennent le titre du film (**name**), le réalisateur (**director**), la moyenne des notes attribuées par les utilisateurs (**average**), l'année de sortie (**releaseYear**), et le lien vers l'image du film (**image**). Cette table permet de gérer les informations liées aux films présents sur le site. La colonne **average** permet de calculer la moyenne des notes attribuées par les utilisateurs.

Table "Review"

La table "Review" enregistre les avis laissés par les utilisateurs sur les films. Chaque avis est identifié par un ID unique (**reviewId**). Les informations stockées comprennent l'ID du film associé (**movieId**), l'ID de l'utilisateur qui a laissé l'avis (**userId**), la note attribuée (**mark**), le commentaire (**comment**), et la date de l'avis (**date**). Cette table permet de suivre les avis (évaluations et commentaires) des utilisateurs pour chaque film. Les clés étrangères (**movieId** et **userId**) lient les avis aux films et aux utilisateurs correspondants.

En résumé, la base de données est conçue pour permettre la gestion des profils d'utilisateurs, des informations sur les films, et des avis laissés par les utilisateurs. Ces données sont cruciales pour les fonctionnalités clés des deux applications, notamment la création de compte, la recherche de films, la visualisation et la gestion des profils, des films et des avis.

V- Architecture du projet

Notre projet est architecturé en 3 parties :

L'application Web : Destinée aux utilisateurs, l'application Web est développée en utilisant des technologies telles que HTML, CSS, PHP et JavaScript. Son rôle principal est d'offrir une interface conviviale pour les utilisateurs finaux. Elle interagit avec la base de données de l'infrastructure pour fournir des fonctionnalités dynamiques et personnalisées.

L'application Java : Conçue spécifiquement pour les administrateurs de Rateflix, l'application Java est écrite en utilisant le langage Java et tire parti de la bibliothèque Swing pour une interface graphique utilisateur (GUI). Cette application est déployée localement et communique avec la base de données à travers le serveur WAMP. Son objectif est de fournir aux administrateurs un ensemble d'outils et de fonctionnalités avancées pour gérer efficacement les aspects administratifs de Rateflix.

L'infrastructure : L'infrastructure est la base du système. Elle est conçue pour héberger l'application Web et être accessible aux utilisateurs. Elle gère également la communication entre l'application Web et la base de données. À ceci s'ajoute une sauvegarde de la base de données et des fichiers web ainsi qu'un reverse proxy pour permettre la connexion à l'application Web à tous les utilisateurs connectés au réseau wifi de l'ordinateur de l'école.

VI- Scénario de démonstration

A) Application web

Scénario de l'Utilisateur Mr. Tintin sur Rateflix:

1. Création du Compte :

- M.Tintin se rend sur Rateflix et clique sur le bouton "S'inscrire".
- Il saisit son pseudo, son adresse e-mail, son mot de passe et confirme le mot de passe.
- Le système vérifie la validité des informations et crée son compte.

2. Connexion au Compte :

- M.Tintin accède à la page de connexion et saisit son pseudo et son mot de passe.
- Le système vérifie les informations et le connecte à son compte.

3. Mot de Passe Oublié :

- En cas d'oubli de mot de passe, M.Tintin clique sur "Mot de passe oublié".
- Il entre son adresse email, son nouveau mot de passe et la confirmation de son mot de passe. Si l'adresse mail est vérifiée, le mot de passe est mis à jour.

4. Recherche du Film "Mamamia" :

- M.Tintin utilise la barre de recherche pour trouver le film "Mamamia".
- Il consulte la note moyenne du film et les commentaires des autres utilisateurs.

5. Dépôt d'un Avis sur "Avatar" :

- Sur la page d'accueil, M.Tintin recherche "Avatar" à l'aide de la barre de recherche.
- Ayant vu le film la veille, il dépose une note et un avis.

6. Promotion de sa Chaîne YouTube :

- M.Tintin décide de laisser un commentaire sous le film "Titanic" pour promouvoir sa chaîne YouTube.
- Il poste un commentaire avec le texte "Aller voir ma chaine Youtube @Tintin". (Cet avis est un mauvais commentaire et l'administrateur devra le supprimer dans l'application Java. Rf.Scénario Application Java).

7. Dépôt d'un Avis sur "American Sniper" :

- Retour à la page d'accueil, M.Tintin recherche et sélectionne "American Sniper".
- Il laisse une note et un commentaire, puis réalise qu'il n'a pas vu le film.

8. Suppression du Commentaire sur "American Sniper" :

- M.Tintin va dans son profil, trouve le commentaire sous "American Sniper" et le supprime en cliquant sur la poubelle.

9. Modification du Profil :

- M.Tintin accède à la section "Editer profil".
- Il modifie sa photo, sa date de naissance et son pseudo.

10. Vérification de la Suppression du Commentaire :

- M.Tintin retourne sur la page de "American Sniper" pour s'assurer que son commentaire a bien été supprimé.

11. Déconnexion :

- M.Tintin se déconnecte de son compte, terminant ainsi sa session sur Rateflix.

Ce scénario reflète les différentes étapes que M.Tintin pourrait suivre lors de son utilisation de l'application web Rateflix, y compris des actions telles que la recherche de films, le dépôt d'avis, la modification de son profil, et la gestion de ses commentaires.

B) Application Java

Ouvrir l'application via la classe « Mainframe » tout en s'assurant qu'à la ligne 19, l'état 'CONNEXION' soit renseigné :

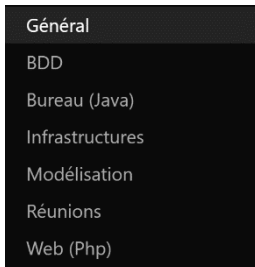
- Se connecter avec les coordonnées de l'administrateur : username – 'admin' / password : 'Admin'
- Cliquez sur « Gestion des profils »
- Cherchez un profil dans la barre de recherche. Par exemple, entrez les lettres « alb ». Enfin, cliquez sur le bouton « Rechercher ».
- Le profil d'Albane s'affiche. Cliquez sur son nom d'utilisateur pour accéder à son profil.
- Modifiez le pseudo « Albane » par « Carteron ». Validez votre changement. Vous revenez alors sur la page « Gestion des profils ».
- Cliquez sur le pseudo de Xavier pour accéder à son profil. Tentez de supprimer un de ses avis en appuyant sur le bouton « Supprimer » associé à celui-ci. Puis supprimez complètement son profil en cliquant sur le bouton « Supprimer » situé à droite des informations personnelles. Vous revenez alors sur la page « Gestion des profils ».
- Dans la barre noire du haut de l'application, cliquez sur l'emplacement du titre « Gestion des films » pour accéder à la gestion des films.
- Cherchez un film dans la barre de recherche. Par exemple, entrez les lettres « har ». Enfin, cliquez sur le bouton « Rechercher ».
- Le film « Harry Potter » s'affiche. Cliquez sur le titre du film pour accéder au film.
- Modifiez la date de sortie du film « 2001 » par « 2002 ». Validez votre changement. Vous revenez alors sur la page « Gestion des films ».
- Accédez au film « Fight club ».
- Finalement, revenez au menu « Gestion des films » en cliquant sur le bouton « Retour ».
- Puis, accédez au film « American sniper ».
- Supprimez un avis en cliquant sur le bouton « Supprimer » associé à celui-ci. Vous revenez alors sur la page « Gestion des films ».
- Accédez au film « Interstellar » et supprimez-le. Vous revenez alors sur la page « Gestion des films ».
- Ajoutez un film en cliquant sur le bouton « Ajouter un film ». Rentrez comme titre « Interstellar », comme producteur « Christopher Nolan », comme date de sortie « 2014 » comme image « Interstellar.jpg » et comme description ce que vous souhaitez. Cliquez sur « Ok ». Assurez-vous maintenant de l'ajout du film sur le menu « Gestion des films »

Vous avez vu toutes les fonctionnalités de l'application Java en tant qu'administrateur. Vous pouvez désormais fermer l'application.

VII- Présentation de l'organisation

Concernant l'organisation au sein de l'équipe, nous avons créé tout un environnement à la fois pour pouvoir communiquer correctement mais également pour pouvoir travailler en groupe efficacement.

Teams :



Teams nous a beaucoup servi pour parler dans différents canaux spécifiques comme on peut le voir sur l'image.

Nous nous en sommes également servi pour ajouter les fichiers communs tel que les rapports, les schémas, les plannings, ...

Figure 15 : Canaux Teams

GitHub :

Nous avons utilisé GitHub afin de versionner notre travail et de le mettre en commun au sein du groupe. Certains l'avait déjà utilisé auparavant pour des projets personnels, c'était la première fois qu'on réalisait un projet GitHub de groupe. Nous avons mis du temps à comprendre le fonctionnement de l'environnement mais nous y avons consacré le temps qu'il fallait pour être plus efficace par la suite.

Nous avons créé 3 branches :

- Bureau (application Java)
- Web (application Web)
- Infra (infrastructure)

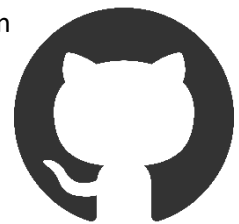


Figure 16 : Logo GitHub

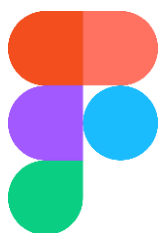
Visual Studio Code :



Visual Studio Code a été l'éditeur de texte qui a été privilégié pour la partie Web ainsi que pour l'infrastructure. Il était relié directement au GitHub pour avoir les mises à jour de chacun.

Figure 17 : Logo Visual Studio Code

FIGMA :



Nous avons utilisé FIGMA pour le design de la partie WEB. C'est un outil très puissant qui nous a permis de faire le design comme on le souhaite puis de l'implémenter dans le code en HTML et CSS.

FIGMA nous a aidé à faire l'architecture du code, cependant nous avons dû la réadapter ensuite pour qu'elle soit en responsive. C'était la première fois que nous l'utilisons donc nous avons découvert les fonctionnalités au fur et à mesure.

Figure 18 : Logo Figma

**IntelliJ :**

Le logiciel en Java a été codé à l'aide de l'éditeur IntelliJ car plus pratique pour lier la base de données. Il était également relié à GitHub.

Figure 19 : Logo IntelliJ

JIRA :

Nous avons testé l'outil JIRA pour voir les sprints en cours de chacun. Nous avons tous commencé à l'utiliser mais petit à petit, à cause des problèmes qui n'étaient pas liés directement aux sprints, nous avons arrêté de le mettre à jour.

C'était tout de même intéressant d'avoir une première approche de cet outil qui peut être très pratique pour des projets de plus longue durée et de plus grande envergure.

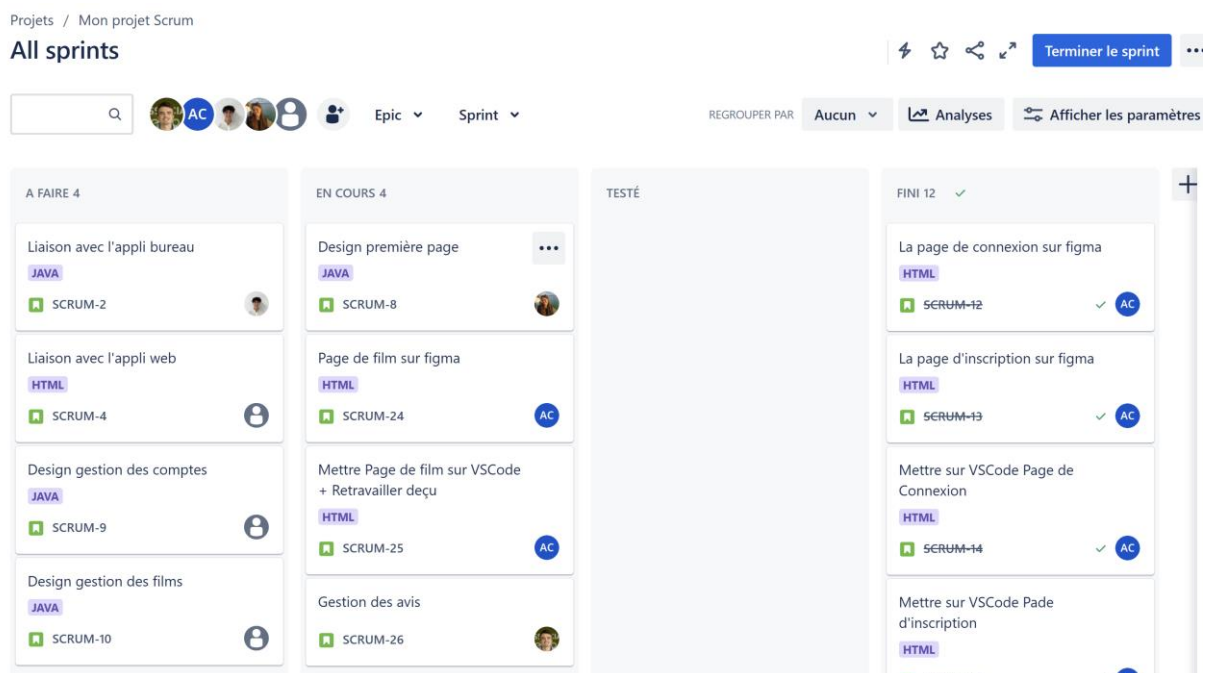


Figure 20 : Capture d'écran de Jira

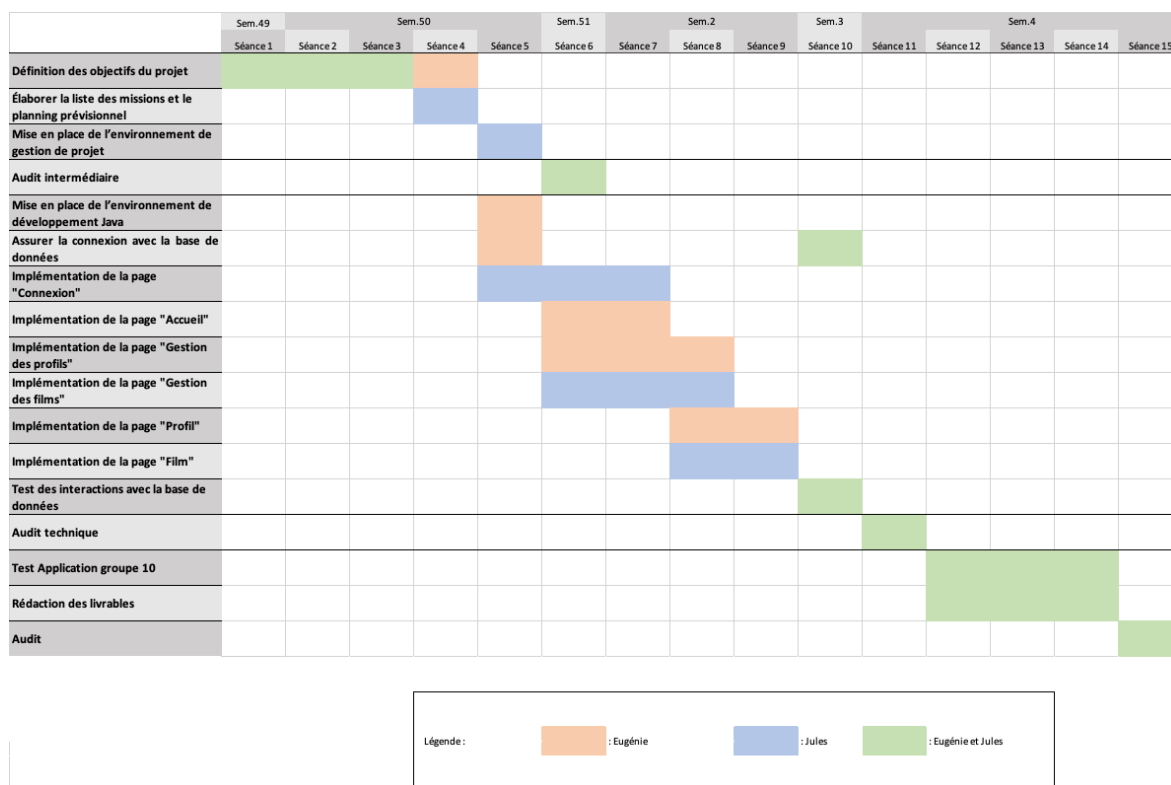


Figure 21 : Planning effectif partie Java



Figure 22 : Planning effectif partie Web

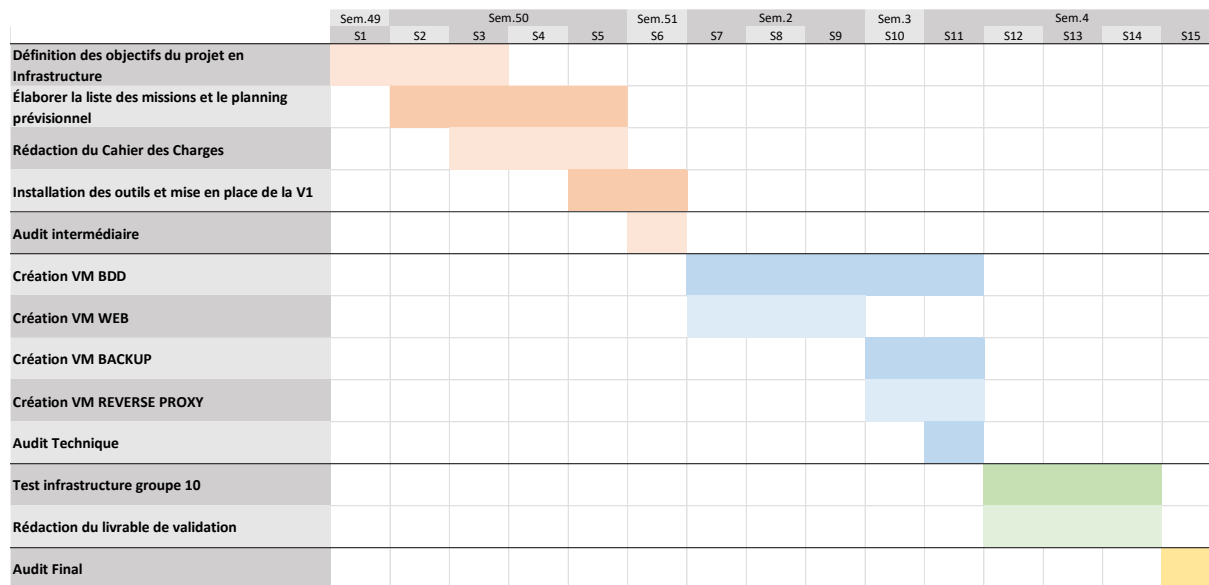


Figure 23 : Planning effectif Infrastructure

RateFlix : Critique de films

Groupe 9

Membres du groupe : LISA BENVENUTI, Albane CARTERON, Xavier DURUMAIN, Jules POTIRON, Eugénie ROCHERON, Ambre SANTANACH

DIAGRAMME DE GANTT

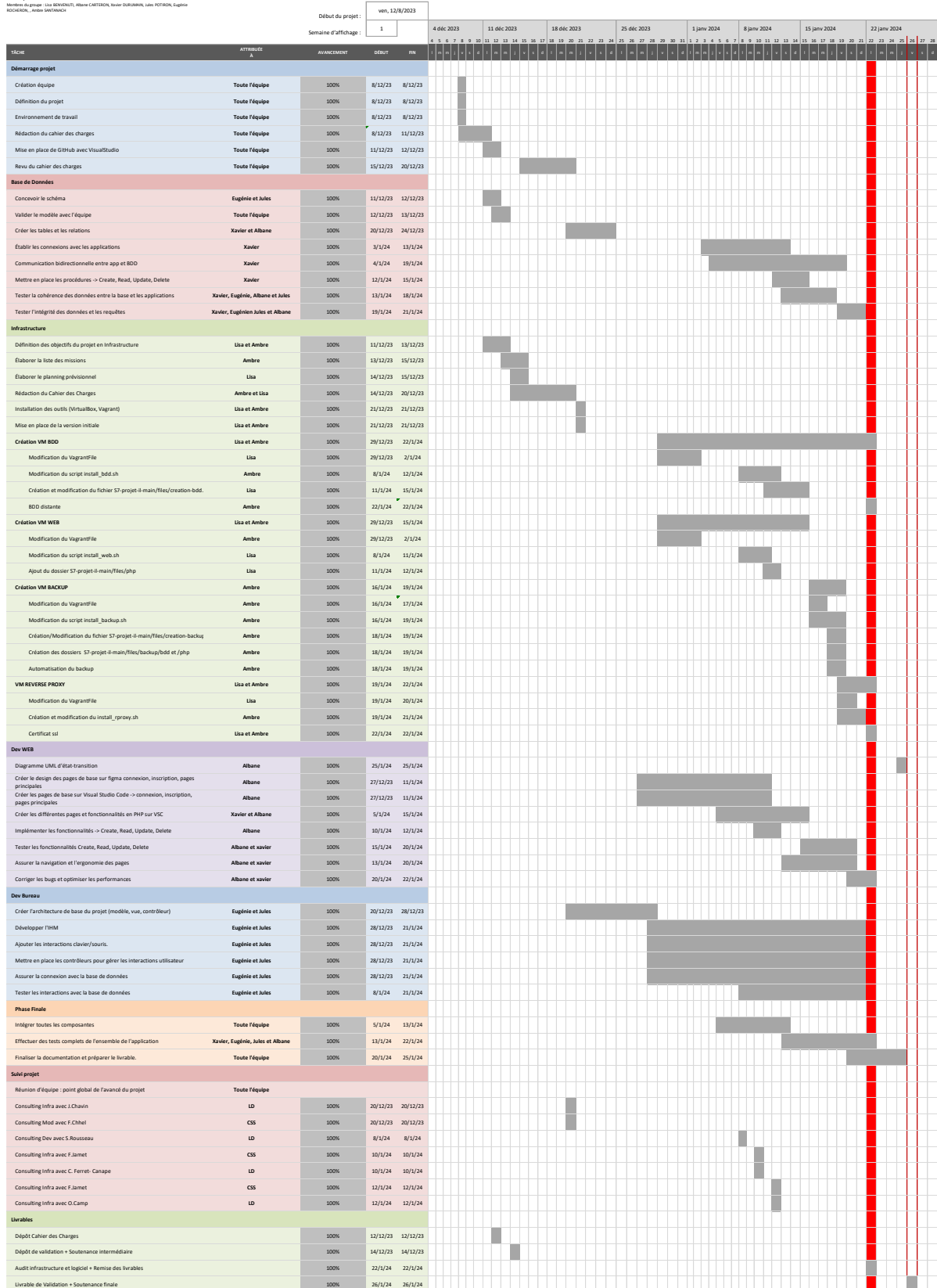
<https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html>

Figure 24 : Diagramme de Grantt effectif

VIII- Table des figures

Figure 1 : Extrait du script install_web.sh.....	2
Figure 2 : Extrait du script install_bdd.sh : Identifiants et chemin du fichier sql	2
Figure 3 : Extrait du script install_bdd.sh : Configuration de la base de données	2
Figure 4 : Extrait du script install_bdd.sh : connexion au serveur.....	2
Figure 5 : Extrait du script install_bdd.sh : mise à jour base de données	3
Figure 6 : Extrait du script install_backup.sh.....	3
Figure 7 : Extrait du script install_rproxy.sh	3
Figure 8 : Diagramme UML d'état-transition de Rateflix	4
Figure 9 : Diagramme de classe global de l'application Java.....	6
Figure 10 : Diagramme de classe du package vue	6
Figure 11 : Diagramme de classe package contrôleur.....	7
Figure 12 : Diagramme de classe package bdd.....	7
Figure 13 : Modèle Conceptuel des Données (MCD)	8
Figure 14 : Modèle Logique des Données (MLD)	8
Figure 15 : Canaux Teams.....	13
Figure 16 : Logo GitHub	13
Figure 17 : Logo Visual Studio Code	13
Figure 18 : Logo Figma.....	13
Figure 19 : Logo IntelliJ	14
Figure 20 : Capture d'écran de Jira.....	14
Figure 21 : Planning effectif partie Java	15
Figure 22 : Planning effectif partie Web.....	16
Figure 23 : Planning effectif Infrastructure	16
Figure 24 : Diagramme de Grantt effectif	17