CPOA

Projet VIP

Mehdi LABOURDETTE

Simon PASQUEREAU

G6S3

Table des matières

[Conception 2](#_Toc485904138)

[Modélisation UML du système d’information 2](#_Toc485904139)

[Modélisation UML des processus métiers 3](#_Toc485904140)

[Base de données 4](#_Toc485904141)

[Schéma de la base de données 4](#_Toc485904142)

[Scripts 4](#_Toc485904143)

[Développement 5](#_Toc485904144)

[Organisation de l’application et couche présentation 5](#_Toc485904145)

[Affichage des VIP 6](#_Toc485904146)

[Affichage des films 7](#_Toc485904147)

[Ajout d’un VIP 8](#_Toc485904148)

[Ajout d’un film 9](#_Toc485904149)

[Ajout d’une photo 10](#_Toc485904150)

[Affichage des évenements 10](#_Toc485904151)

[Couche d’accès aux données de la base 11](#_Toc485904152)

[Couche métier 11](#_Toc485904153)

[Bilan conduite du projet 11](#_Toc485904154)

[Découpage du projet 11](#_Toc485904155)

[Temps estimé et temps réel et explications 11](#_Toc485904156)

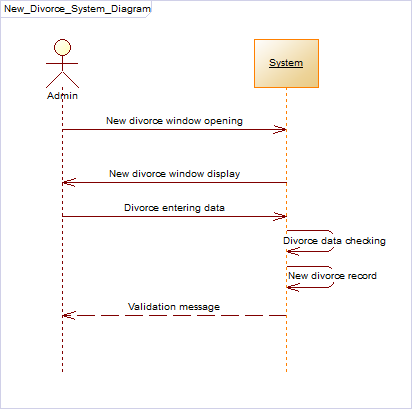
[Conclusion générale 12](#_Toc485904157)

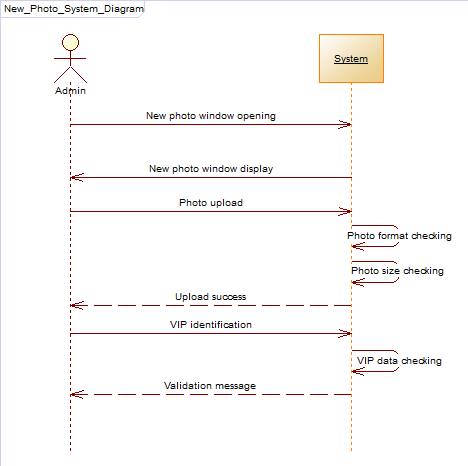
[Bilan technique 12](#_Toc485904158)

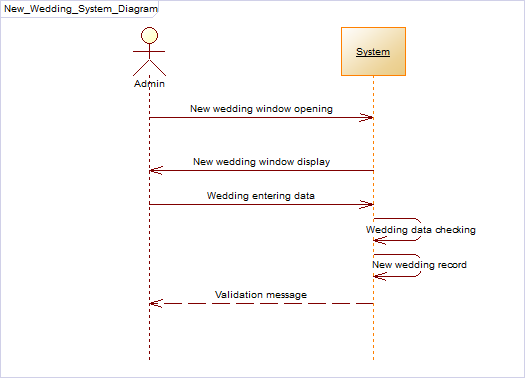
[Remarques personnelles 12](#_Toc485904159)

# Conception

## Modélisation UML du système d’information

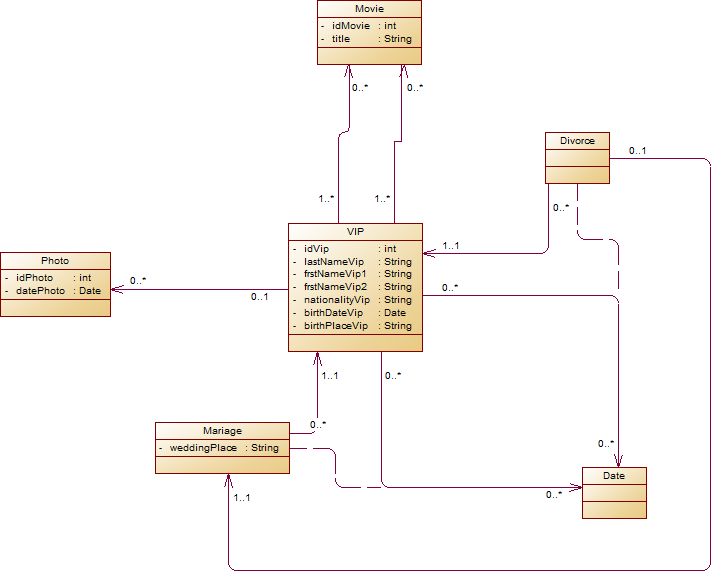






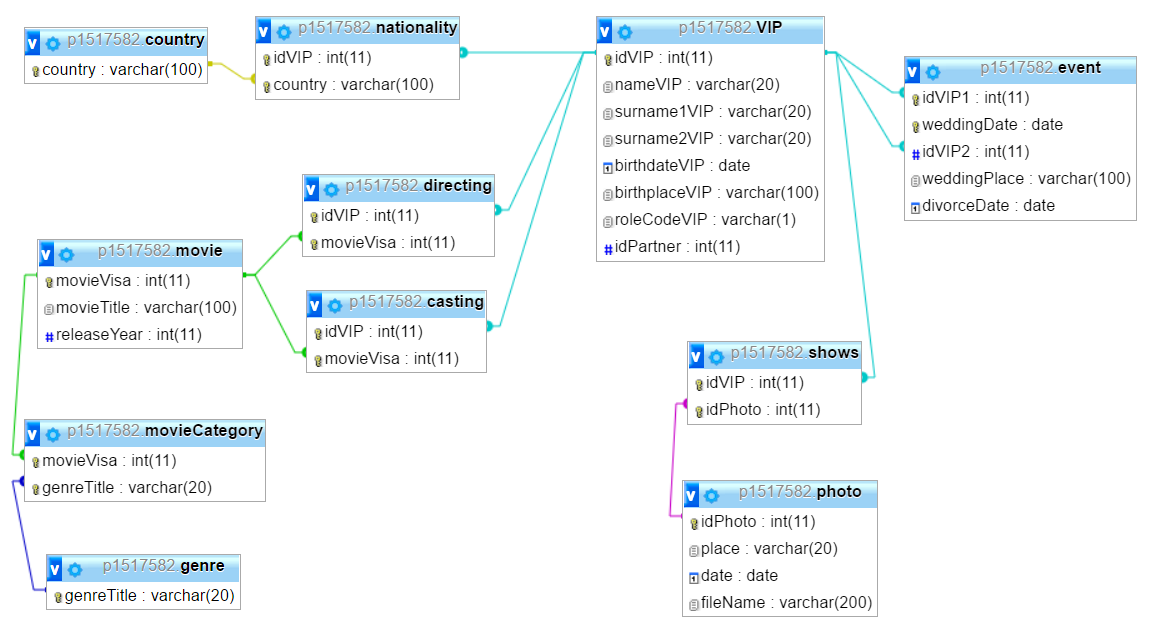
Le diagramme ne figure pas pour la création d’un VIP mais il demeure très semblable.

## Modélisation UML des processus métiers



# Base de données

## Schéma de la base de données



Nous avons décidé qu’il est possible qu’un VIP ait plusieurs nationalités, comme c’est le cas dans la réalité (Angelina Jolie a 2 nationalités par exemple). Nous avons donc choisi de créer une entité relation « nationality » qui lie un VIP à un pays.

De la même façon, nous avons décidé qu’un film pouvait appartenir à plusieurs genres (comedy et drama par exemple pour une comédie dramatique). Il y a donc une entité relation « movieCategory » qui lie un film à un genre.

Pour le rôle du VIP, nous avons choisi de le définir avec un attribut de type caractère et prenant les valeurs A pour un acteur, D (director) pour un réalisateur, B (both) pour un acteur/réalisateur et N (none) si le VIP n’a pas de rôle particulier. La valeur de cet attribut est gérée par des triggers. Ainsi, l’administrateur n’a pas à s’en soucier.

Pour toutes les clés primaires des tables VIP, movie, event, l’auto-incrémentation est mise en place pour plus de simplicité.

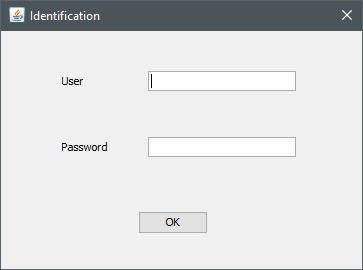
## Scripts

Tous les scripts sont présents dans le dossier du projet.

# Développement

## Organisation de l’application et couche présentation

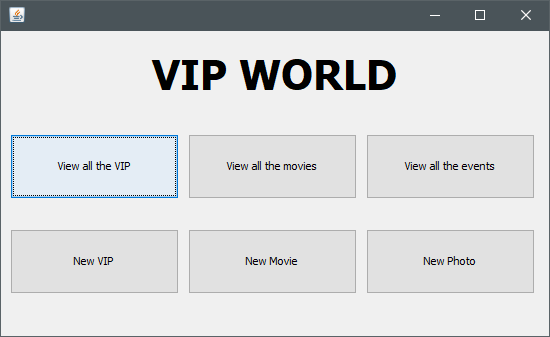
Nous avons choisi de développer notre application en anglais (variables, fonctions, interface, etc.).



Le premier écran (figure ci-dessus) est l’écran de connexion afin de se connecter en tant qu’administrateur de la base de données.

Nous avons choisi de développer un écran d’accueil principal permettant d’accéder à différentes fonctions (figure ci-dessous) :

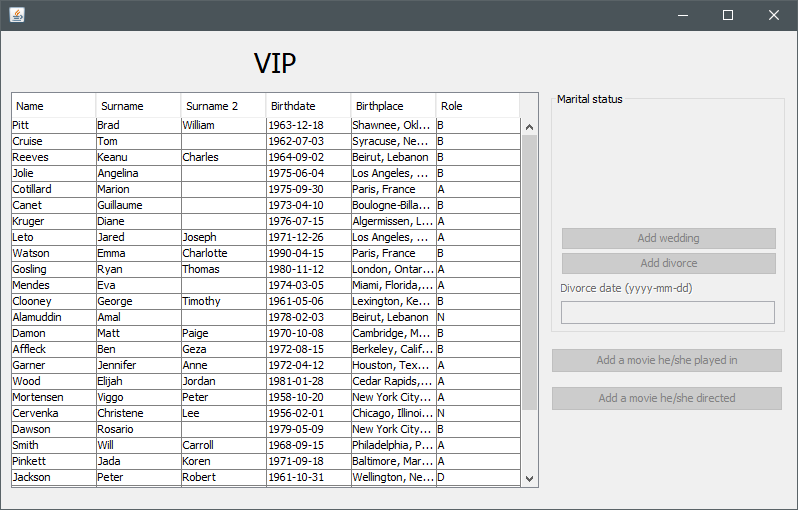
* Afficher la liste des VIP ;
* Afficher la liste des films ;
* Afficher la liste des évènements ;
* Ajouter un VIP, un film, un évènement, une photo.



Certaines de ses actions n’étaient pas explicitement demandées, mais elles semblent être des fonctionnalités plutôt importantes pour l’administrateur, nous avons donc choisi de les développer.

### Affichage des VIP

Cette fenêtre affiche sous forme de tableau les VIP présents dans la base de données. Lorsqu’un VIP est sélectionné, un encadré sur la droite de la fenêtre permet de voir son statut matrimonial. Pour le VIP sélectionné, il est aussi possible d’ajouter un mariage, un divorce, un film dans lequel il a joué ou qu’il a dirigé. Ces fonctionnalités ne figuraient pas au cahier des charges mais rendent l’application facile à utiliser pour n’importe qui.

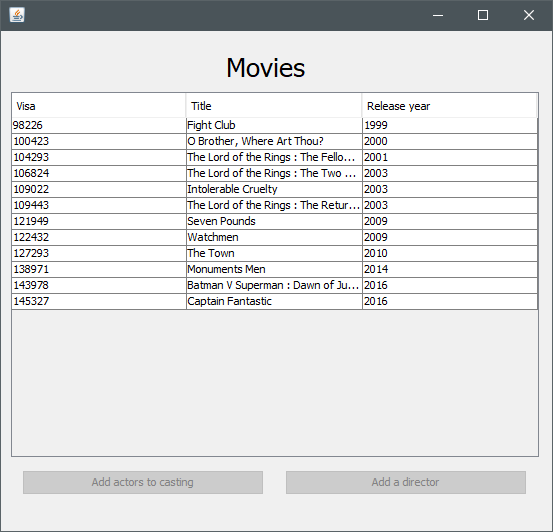


Contrôle de saisie :

Pour un VIP marié, il n’est pas possible de saisir un mariage et pour un VIP non marié, il n’est pas possible de saisir un divore, les boutons restent indisponibles en fonction des cas.

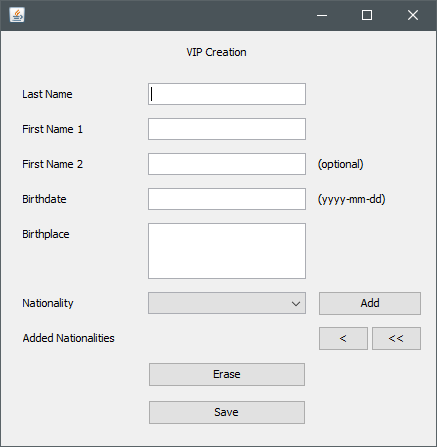
### Affichage des films

De façon analogue, cette fenêtre (figure ci-dessous) affiche sous forme de tableau les films présents dans la base de données. Lorsqu’un film est sélectionné, il est possible d’ajouter des acteurs ou des réalisateurs.



### Ajout d’un VIP

L’administrateur doit rentrer les informations suivantes :



Le second prénom est optionnel et la date doit être renseignée au format anglo-saxon. Tous les autres champs sont obligatoires !

Concernant les nationalités, l’utilisateur choisit dans la JComboBox et clique sur Add. Le pays est ainsi ajouté à une ArrayList qui sera utilisée pour l’insertion dans la base. En cliquant sur « < », l’utilisateur supprime de la liste le dernier pays qu’il vient d’ajouter et « << » permet d’effacer tous les pays de la liste.

En cliquant sur Erase, on supprime toutes les informations qui ont été remplies.

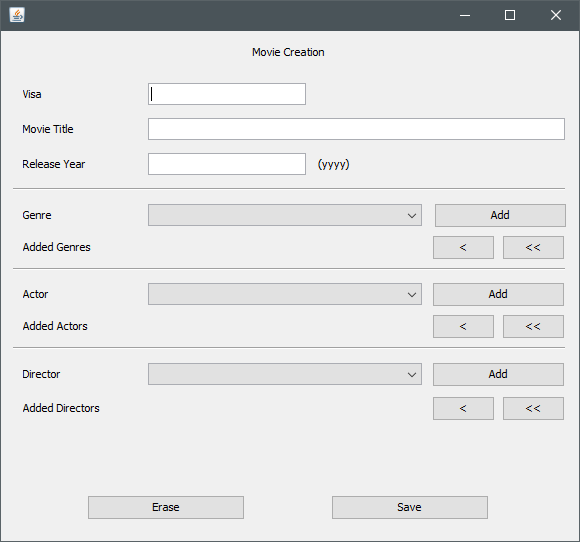
En cliquant sur Save, on insère le VIP dans la base et on crée les lignes de la table nationality avec le numéro de VIP correspondant et la ou les nationalités sélectionnées.

Contrôle de saisie :

Pour la saisie des dates, nous avons utilisé le format anglo-saxon. Ainsi, les seuls formats disponibles pour le 5 avril 2017 sont : « 2017-04-05 », « 2017-4-05 », « 2017-04-5 » ou « 2017-4-5 ». Tous les autres formats entraineront l’affichage d’un message d’erreur demandant la rectification. Il est également impossible d’ajouter une date future.

### Ajout d’un film

L’administrateur doit rentrer les informations suivantes :



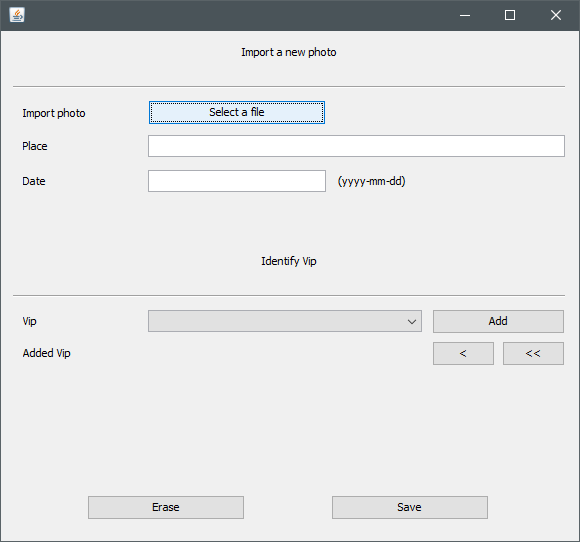
Pour les genres, les acteurs et les directeurs, le fonctionnement est exactement le même que pour l’ajout d’une nationalité pour un VIP. L’utilisateur doit saisir obligatoirement les 3 champs du haut et au moins un genre. Les acteurs et réalisateurs à ajouter au film ne sont pas obligatoires.

Contrôle de saisie :

Le Visa ne doit contenir que des entiers. L’année doit être un nombre à 4 chiffres. Si ces conditions ne sont pas remplies, un message d’erreur s’affiche demandant la rectification.

### Ajout d’une photo

L’administrateur doit rentrer les informations suivantes :



Même principes que sur les autres fenêtres. L’identification sur une photo est faite à l’ajout de la photo et non à postériori.

Contrôle de saisie :

La date doit être renseignée au bon format (voir plus haut). Les 3 champs du haut sont obligatoires. Si il manque un champ ou si le format de la date n’est pas correct, un massage d’erreur s’affiche demandant la rectification.

### Affichage des évenements

Cette fenêtre affiche juste sous forme de tableau l’ensemble des évènements renseignés. Il n’est pas possible d’effectuer plus d’actions depuis cette fenêtre.

## Couche d’accès aux données de la base

## Couche métier

# Bilan conduite du projet

## Découpage du projet

La première partie de ce projet a été consacrée à la modélisation, présentée en première partie. Nous avons profité des deux premières séances afin d’effectuer la modélisation et afin de décider quels choix d’implémentation nous allions faire, car plusieurs solutions étaient envisageables (par exmple pour le nombre de nationalités ou le nombre de genres pour un film).

Dans un second temps, nous avons créé les classes métiers du projet (VIP.java, Movie.java, Photo.java, Event.java). Les classes country et genre n’ont pas été créées car dans la base, elles ne contiennent qu’un seul attribue, à savoir le libellé du genre ou le nom du pays. Il était inutile d’en faire des classes java, ces valeurs pouvant être récupérées et stockées très facilement grâce aux DAO. Nous avons aussi créé les tables dans la base correspondantes à notre modélisation, afin que les tables et les classes java soient en adéquation.

La troisième partie a été consacrée au développement de l’application elle-même. Nous avons créé les interfaces que nous souhaitions et avons développé les DAO, les modèles de données en conséquence.

La partie finale, qui s’est aussi un peu déroulées en même temps que la précédente, a été consacré au développement du site.

Remarques sur la répartition du travail :

Ayant tous deux des compétences à peu près équivalentes en java, bases de données et PHP, nous avons réparti le travail de façon très équitable, ce qui constitue un point très positif de ce travail de groupe !

## Temps estimé et temps réel et explications

Nous envisagions dans un premier temps de réaliser ce projet sur le temps du module de CPOA réel, ce qui était relativement ambitieux.

Cependant, comme nous avons décidé de développer des fonctionnalités qui n’étaient pas explicitement demandées au cahier des charges, la charge de travail a été un peu plus important que prévue. La rédaction de ce rapport s’ajoute également au temps de travail réalisé en dehors du module de CPOA. Au total, ce projet nous aura demandé une quinzaine d’heures de travail supplémentaire par rapport au module de CPOA.

# Conclusion générale

## Bilan technique

Nous nous sommes retrouvé face à des problématiques déjà relativement maitrisées, que ce soit en java ou en PHP, et ce grâce aux cours que nous avions déjà réalisés dans ces matières. Ainsi, nous pouvons dire que la difficulté n’était pas très élevée. Nous avons cherché à réutiliser ce que nous avions déjà vu, afin de ne pas « réinventer la roue » pour chaque implémentation. Le seul point nouveau que nous avons abordé lors de ce module a été l’ajout de la fonctionnalité de transfert de photo grâce au protocole FTP. Cette tâche nous a donné un peu de fil à retordre, mais nous avons réussi à la compléter.

Globalement, nous pensons avoir fourni une application fonctionnelle et avec un code plutôt qualitatif. Il y a certainement des choses à revoir, mais dans l’ensemble, nous avons toujours cherché à appliquer les principes qui nous ont été inculqués à l’IUT, pour fournir un code clair et maintenable, autant que faire se peut.

## Remarques personnelles

Nous avons apprécié ce module dans son ensemble, car il se rapproche des tâches réelles d’un analyste-programmeur. Nous avons retiré de la satisfaction à fournir un projet fonctionnel et répondant à l’essentiel des points du cahier des charges.

Nous aurions pu répondre mieux à ce cahier des charges, mais en raison du temps passé à développer des fonctionnalités non demandées, nous avons manqué de temps pour remplir des objectifs qui étaient peut-être plus essentiels, car demandés par le client. Ce point nous amènera à réfléchir par la suite à toujours se conformer au cahier des charges, quitte à mettre notre créativité ou nos envies de côté.

Parmi les points que nous aurions pu améliorer avec plus de temps ou une meilleure organisation dans le travail :

* Compléter les vérifications sur les saisies de données par l’administrateur, et notamment concernant les dates ;
* Rendre le site web un peu plus esthétique et complet ;
* Améliorer le visuel et l’ergonomie de l’application java.