CPOA

Rapport d’analyse

FESTIVAL DE CANNES

Julien Giraud - Mélodie Guérin
21/12/2018

Table des matières

Annexes : maquettes des applications Java et Web.

JavaAppStoryBoard.pdf

WebAppStoryBoard.pdf

Introduction

Description

L'application que nous devons réaliser se fera en Web et en Java.

Elle concerne le Festival de Cannes et permettra de simplifier la gestion des VIP et le planning des projections des films, en interne.

Pour la gestion des VIP qui se fera par une application web, nous auront besoin de mettre en place une base de données qui répertoriera les fiches des différents VIP avec leurs attributs (photo, nationalité, type, coefficient d'importance, prise en charge, compagnon attitré, …).
Il faudra ne pouvoir consulter le site que par connexion, donc que la première page soit une page de connexion, et si celle-ci se fait que cela débloque les fonctionnalités de l'application.
Une fois connecter, l'utilisateur pourra créer de nouvelles fiches VIP, les modifier ou simplement les consulter, à partir d'une liste sur la droite, ou par recherche.

Pour le planning des projections, l'application se fera en Java. Il faudra également une base de données qui répertoriera tous les films ainsi que leurs attributs (nom du film, catégorie, durée, réalisateur, nombre de projections, …).
A partir de l'application il faudra pouvoir consulter le planning global (après avoir générer une première fois le planning), c'est-à-dire le calendrier des séances.
On pourra sélectionner une salle pour voir plus en détail les projections prévues pour cette dernière ou sélectionner un jour pour voir les projections prévues ce jour.
On pourra cocher une case pour ne voir que les séances libres et ajouter un film à ces dernières (cette option sera disponible quelque soit le planning affiché : planning général, planning en fonction de la salle, planning en fonction du jour ou en fonction des séances libres).
On pourra aussi rechercher un film pour voir ses attributs. De même, lorsqu'on cliquera sur une séance occupée, nous aurons un descriptif du film ainsi que la possibilité de supprimer la séance.
Sur la page du planning général il y aura la possibilité de réinitialiser le planning à partir de l'écran du planning principal.
Pour ajouter une séance, on choisi le film dans une liste, on peut les trier par catégorie, ainsi que choisir la durée de réservation de la salle.

Fiche du modèle Cannes

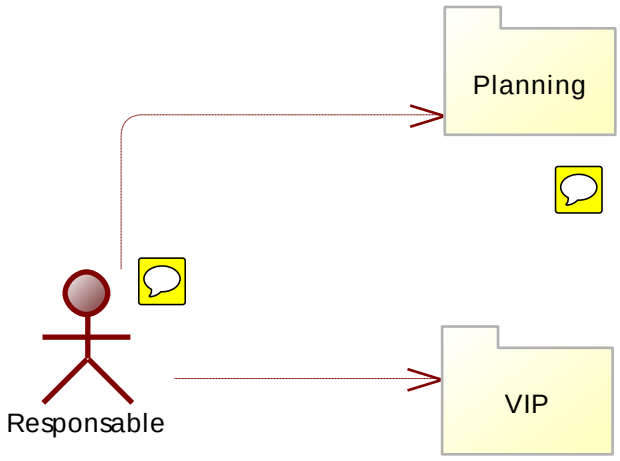
Nom	Cannes
Code	Cannes
Langage objet	Java
Commentaire	Modèle de l'application Java qui gère les plannings et de l'application Web qui gère les VIP (mais ne génère que le code Java)
Auteur	Julien GIRAUD, Mélodie GUERIN
Version	

Objets de niveau modèle

Liste des diagrammes

Nom	Code
packages	packages

Diagramme packages



Package Planning

Liste des diagrammes

Nom	Code
DiagrammeCasUtilisationPlanning	DiagrammeCasUtilisationPlanning
DiagrammeClassesPlanning	DiagrammeClassesPlanning
DiagrammeSequenceAjouterSeance	DiagrammeSequenceAjouterSeance

Diagramme DiagrammeCasUtilisationPlanning

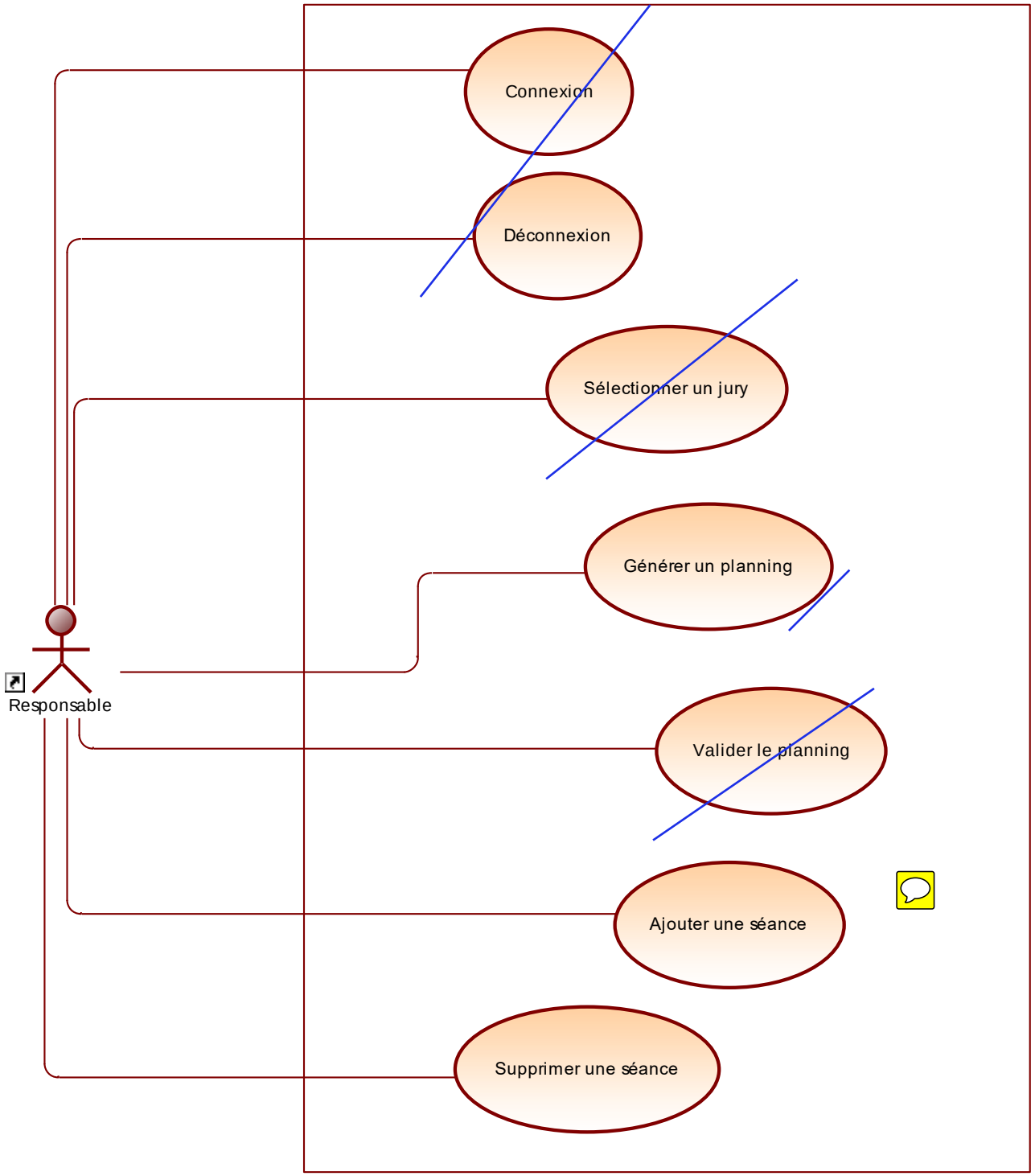
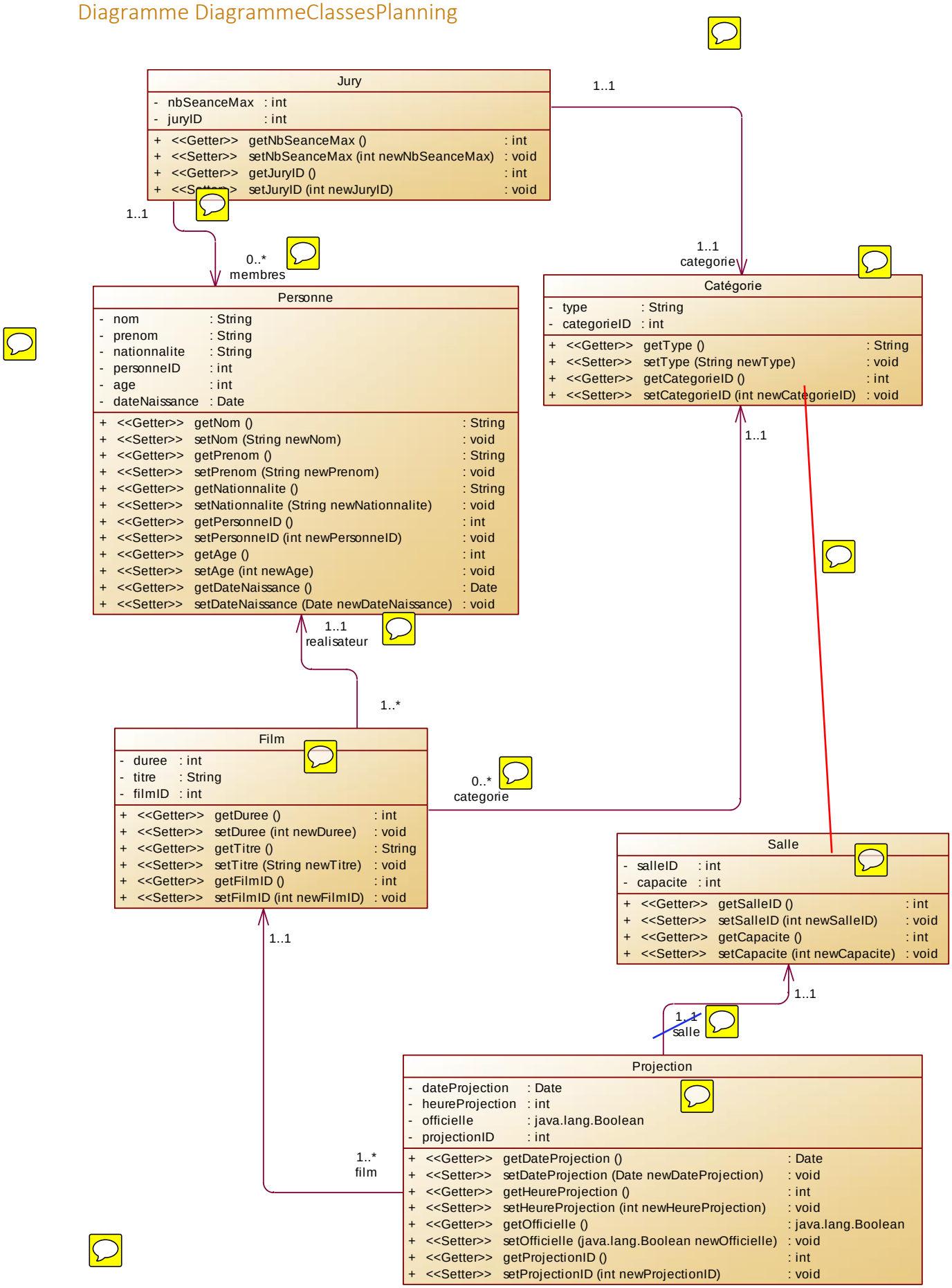


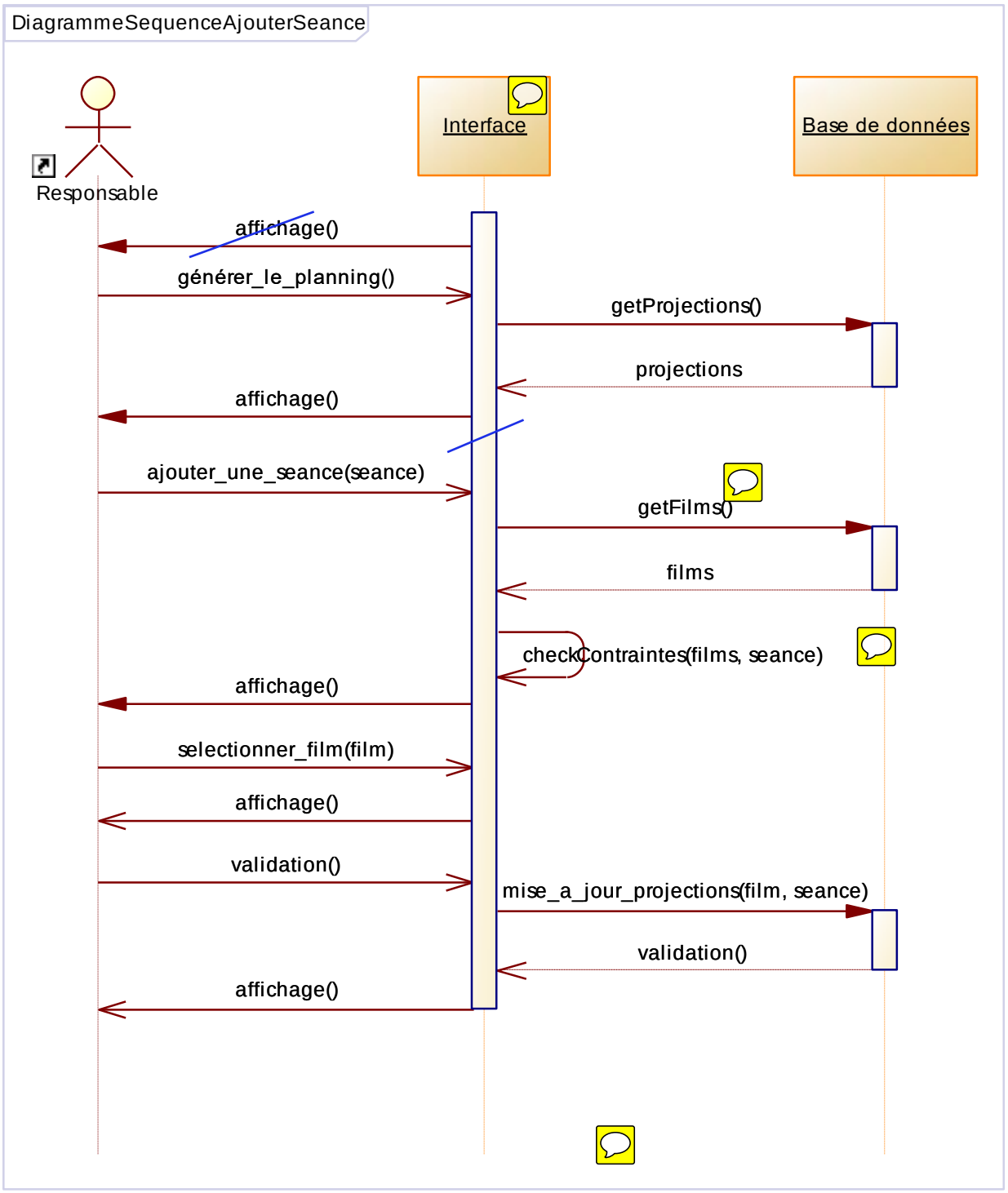
Diagramme DiagrammeClassesPlanning



Génération automatique de code : classe Jury

```
/*********************************************************************
 * Module:  Jury.java
 * Author:  p1704709
 * Purpose: Defines the Class Jury
 *********************************************************************/

package planning;

import java.util.*;

/** @pdOid ae40d9eb-2978-4ebf-a9e2-79f504d12ceb */
public class Jury {
   /** @pdOid f3564c24-7716-4410-9536-8e4e1b08c19c */
   private int nbSeanceMax;
   /** @pdOid 53d9d73e-0453-4c63-b7cf-9587efadc3b6 */
   private int juryID;

   /** @pdRoleInfo migr=no name=Categorie assc=categorie mult=1..1 */
   private Categorie categorie;
   /** @pdRoleInfo migr=no name=Personne assc=association6 coll=java.util.Collection impl=java.util.HashSet mult=0..* */
   private java.util.Collection<Personne> membres;


   /** @pdGenerated default getter */
   public java.util.Collection<Personne> getMembres() {
      if (membres == null)
         membres = new java.util.HashSet<Personne>();
      return membres;
   }

   /** @pdGenerated default iterator getter */
   public java.util.Iterator getIteratorMembres() {
      if (membres == null)
         membres = new java.util.HashSet<Personne>();
      return membres.iterator();
   }

   /** @pdGenerated default setter
     * @param newMembres */
   public void setMembres(java.util.Collection<Personne> newMembres) {
      removeAllMembres();
      for (java.util.Iterator iter = newMembres.iterator(); iter.hasNext();)
         addMembres((Personne)iter.next());
   }

   /** @pdGenerated default add
     * @param newPersonne */
   public void addMembres(Personne newPersonne) {
      if (newPersonne == null)
         return;
      if (this.membres == null)
         this.membres = new java.util.HashSet<Personne>();
      if (!this.membres.contains(newPersonne))
         this.membres.add(newPersonne);
   }

   /** @pdGenerated default remove
     * @param oldPersonne */
   public void removeMembres(Personne oldPersonne) {
      if (oldPersonne == null)
         return;
      if (this.membres != null)
         if (this.membres.contains(oldPersonne))
            this.membres.remove(oldPersonne);
   }

   /** @pdGenerated default removeAll */
   public void removeAllMembres() {
      if (membres != null)
         membres.clear();
   }

   /** @pdOid 08729810-c3a4-4797-90f8-45b0433c25a7 */
   public int getNbSeanceMax() {
      return nbSeanceMax;
   }

   /** @param newNbSeanceMax
    * @pdOid a8447a94-442c-4418-8b5b-25b4fe33bfe4 */
   public void setNbSeanceMax(int newNbSeanceMax) {
      nbSeanceMax = newNbSeanceMax;
   }

   /** @pdOid 499fdbda-6f67-4bff-90a6-aff2081dfe00 */
   public int getJuryID() {
      return juryID;
   }

   /** @param newJuryID
    * @pdOid 5002ca91-003a-4a04-aa29-6d7e35226208 */
   public void setJuryID(int newJuryID) {
      juryID = newJuryID;
   }

}
```

Diagramme DiagrammeSequenceAjouterSeance



Package VIP

Diagramme DiagrammeCasUtilisationVIP

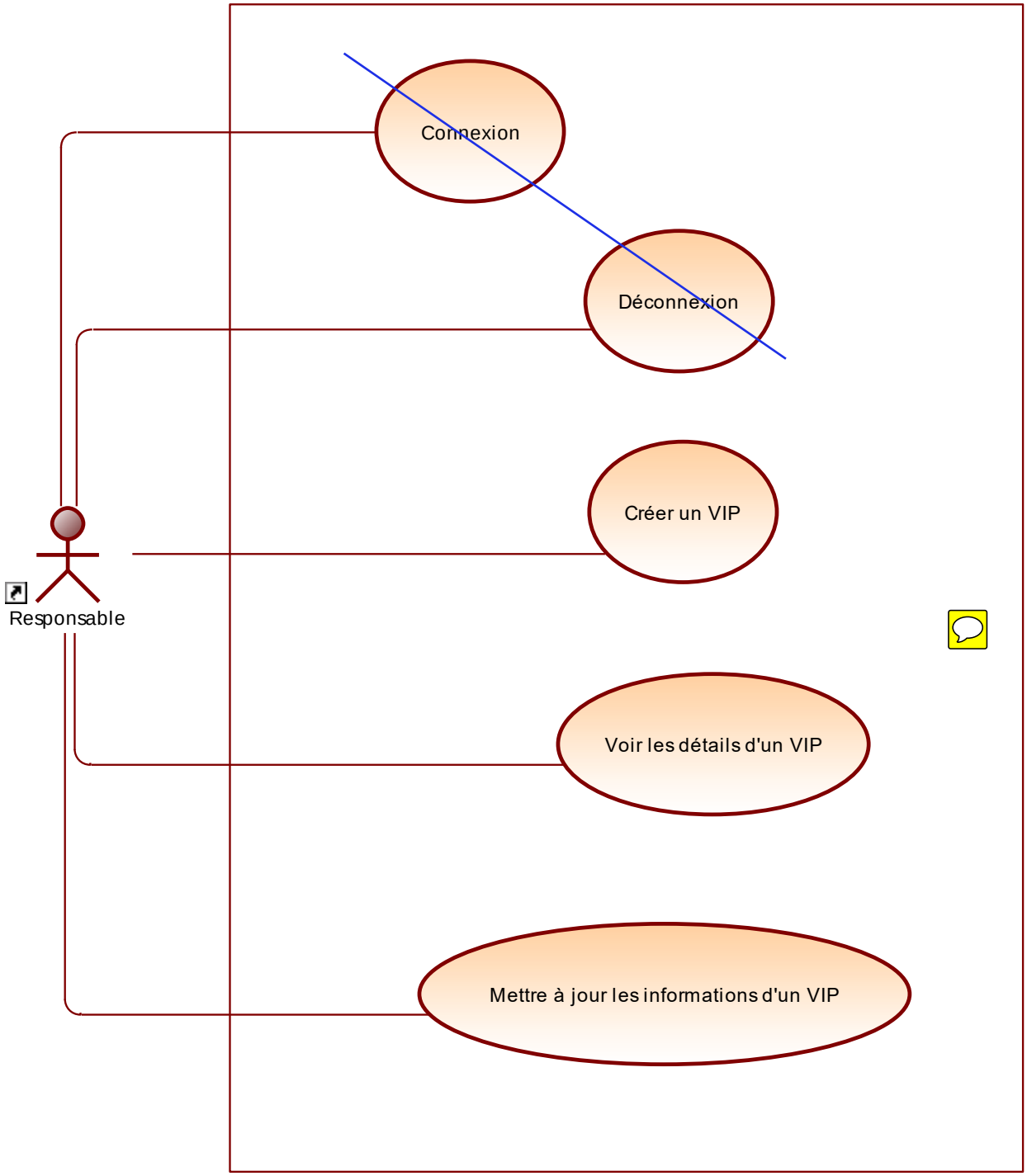


Diagramme DiagrammeClassesVIP

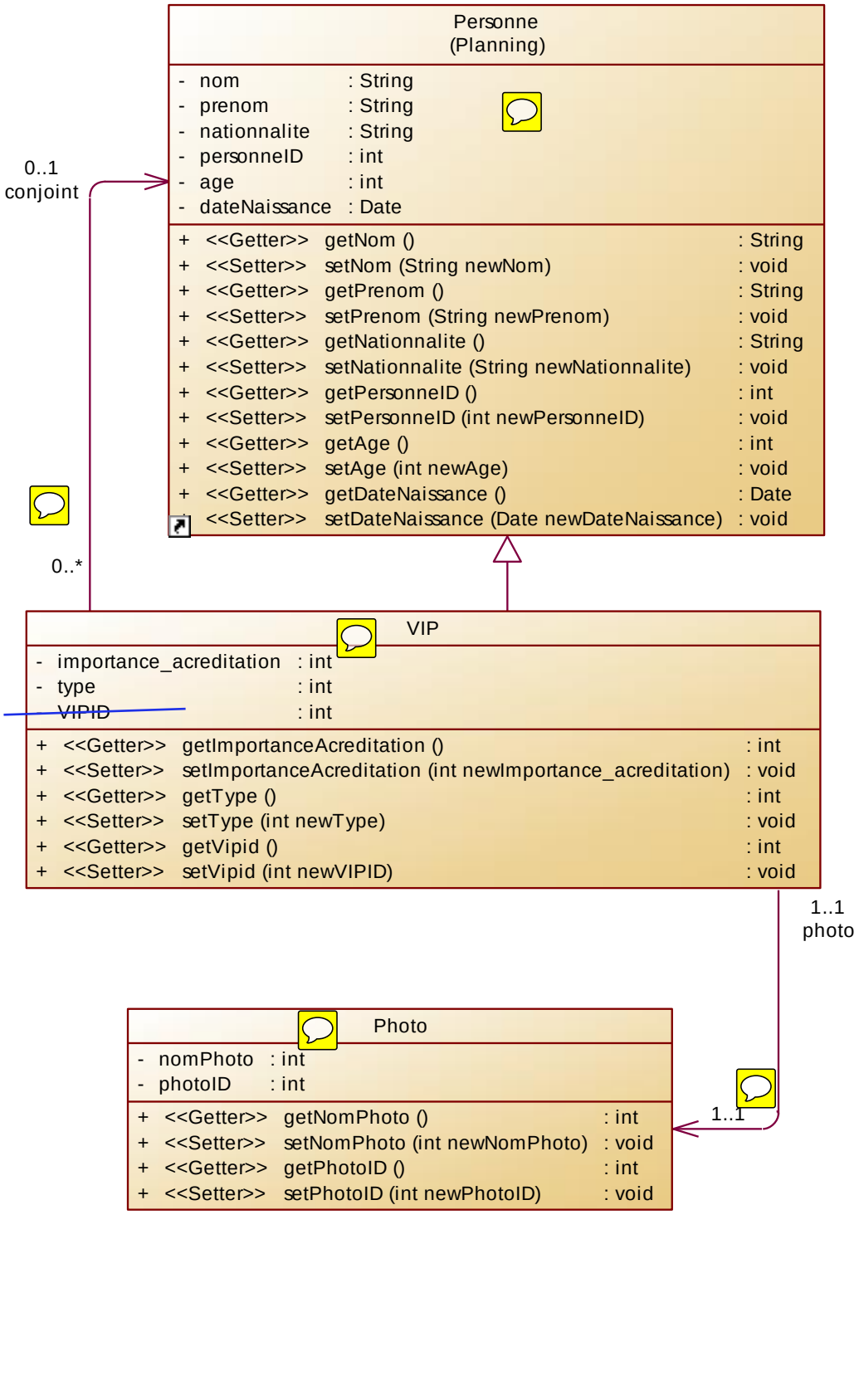
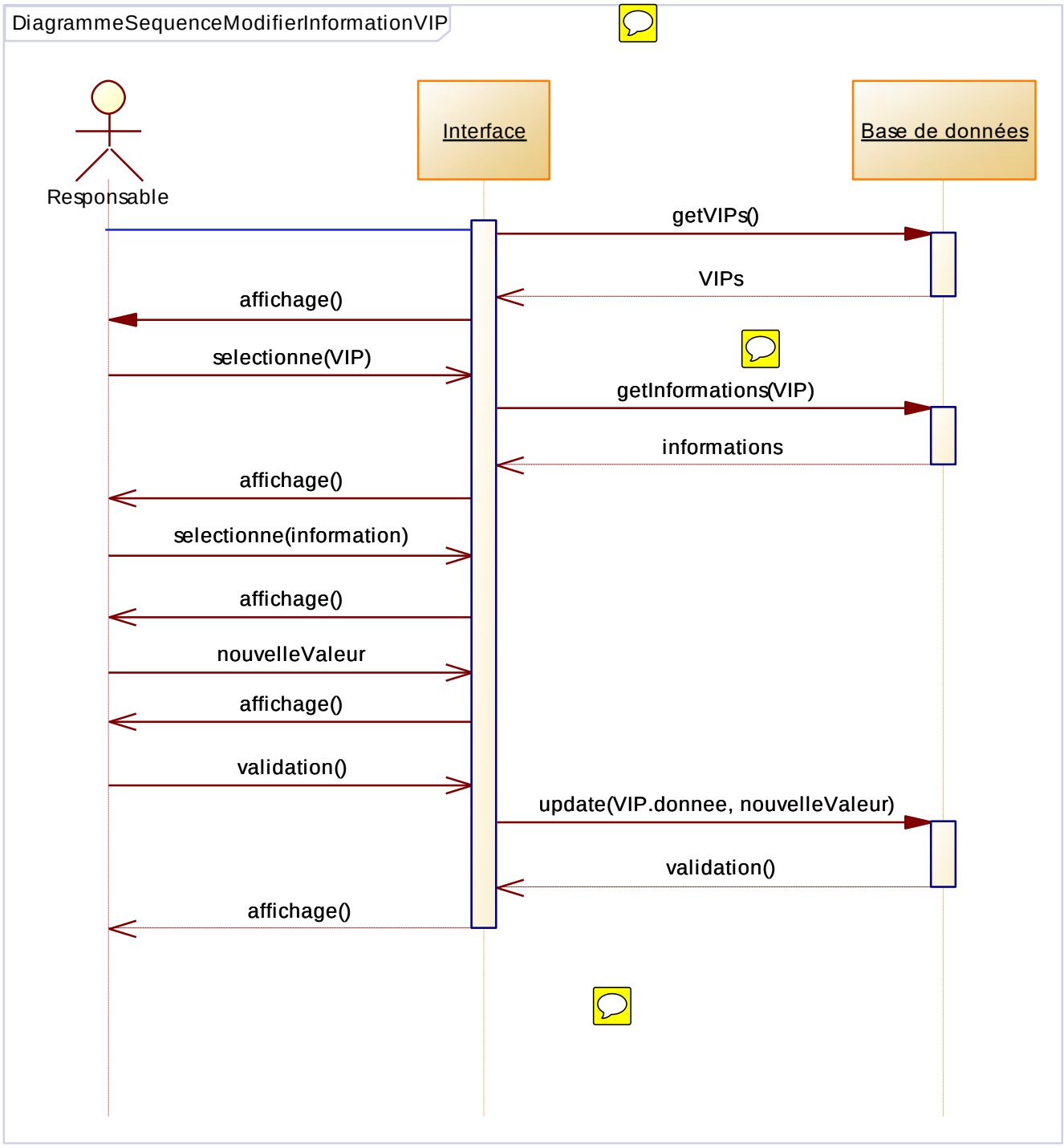
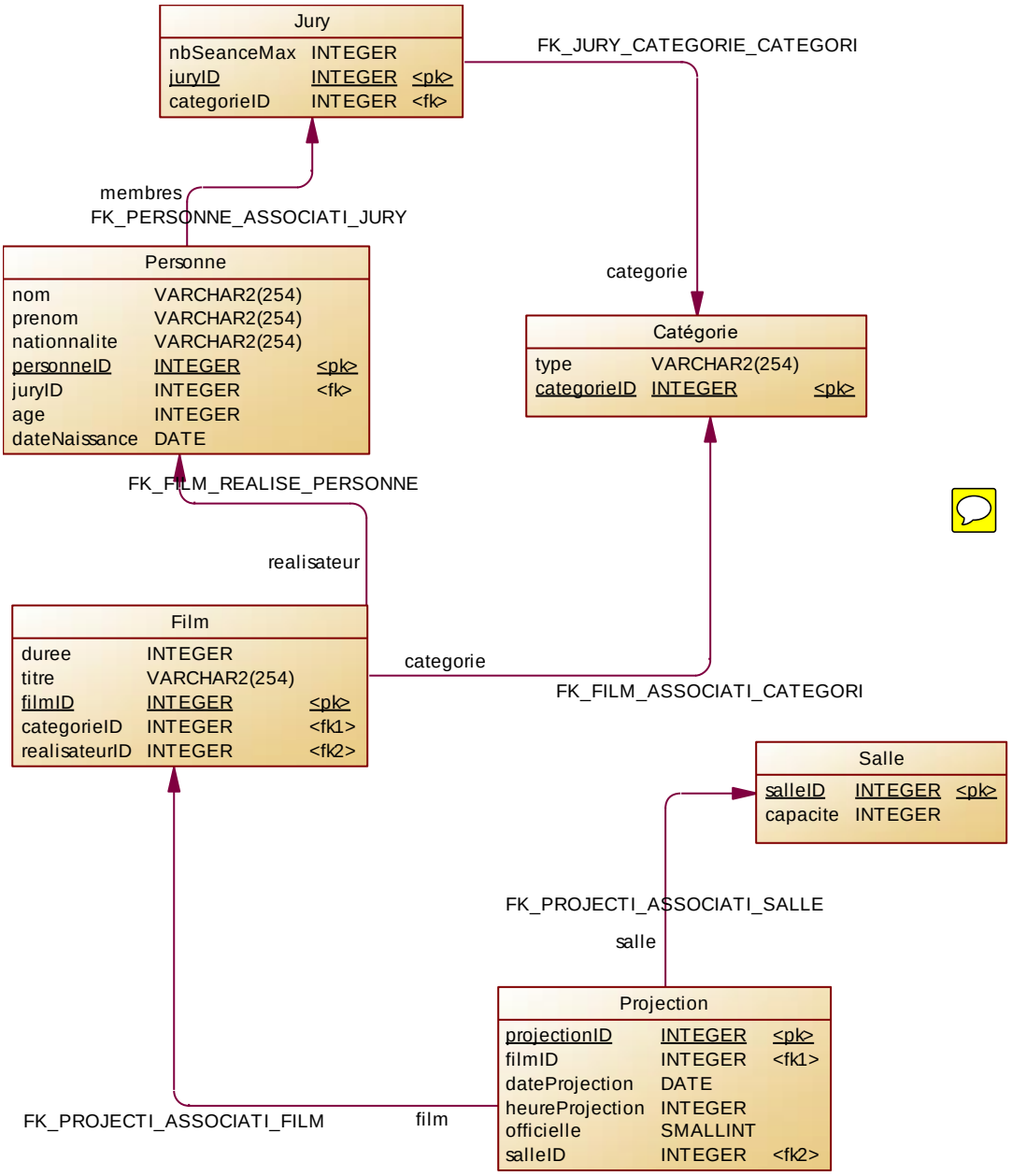


Diagramme DiagrammeSequenceModifierInformationVIP

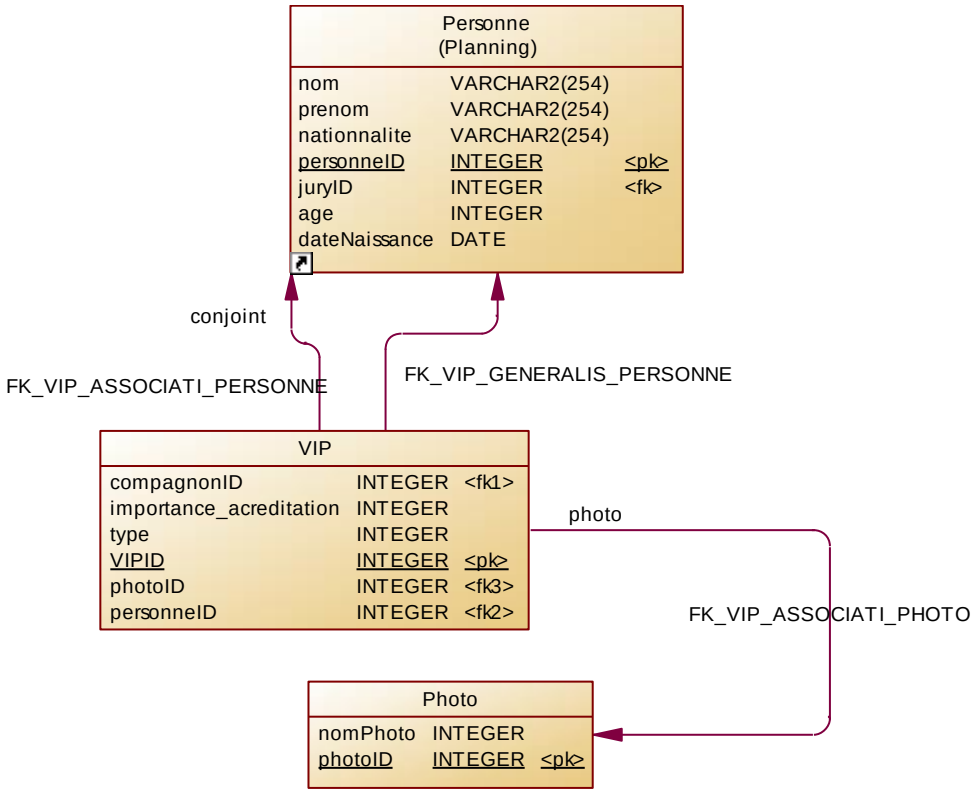


Modèle physique de données

Modèle du planning



Modèle des VIP



Génération automatique du code

Afin de gérer les problèmes de création de tables, nous avons rassemblé le code généré dans un seul script de création de la base de données, composé de 5 parties et une sixième pour la supprimer.

```
/*==============================================================*/
/* Création des tables                                          */
/*==============================================================*/
create table Categorie
(
    type                    VARCHAR(254),
    categorieID             INTEGER
);

create table Film
(
    duree                   INTEGER,
    titre                   VARCHAR(254),
    filmID                  INTEGER,
    categorieID             INTEGER,
    realisateurID           INTEGER
);

create table Jury
(
    nbSeanceMax             INTEGER,
    juryID                  INTEGER,
    categorieID             INTEGER
);

create table Personne
(
    nom                     VARCHAR(254),
    prenom                  VARCHAR(254),
    nationnalite            VARCHAR(254),
    personneID              INTEGER,
    age                     INTEGER,
    dateNaissance           DATE,
    juryID                  INTEGER
);

create table Photo
(
    nomPhoto                VARCHAR(254),
    photoID                 INTEGER
);

create table Projection
(
    projectionID            INTEGER,
    salleID                 INTEGER,
    filmID                  INTEGER,
    dateProjection          DATE,
    heureProjection         INTEGER,
    officielle              SMALLINT
);

create table Salle
(
    salleID                 INTEGER,
    capacite                INTEGER
);

create table VIP
(
    VIPID                   INTEGER,
    importanceAcreditation  INTEGER,
    typeVIP                 INTEGER,
    photoID                 INTEGER,
    compagnonID             INTEGER,
    personneID              INTEGER
);
```

```
/*==============================================================*/
/* Clés primaires des tables                                    */
/*==============================================================*/
alter table Categorie
    add constraint PK_CATEGORIE
        primary key (categorieID);

alter table Film
    add constraint PK_FILM
        primary key (filmID);

alter table Jury
    add constraint PK_JURY
        primary key (juryID);

alter table Personne
    add constraint PK_PERSONNE
        primary key (personneID);

alter table Photo
    add constraint PK_PHOTO
        primary key (photoID);

alter table Projection
    add constraint PK_PROJECTION
        primary key (projectionID);

alter table Salle
    add constraint PK_SALLE
        primary key (salleID);

alter table VIP
    add constraint PK_VIP
        primary key (VIPID);


/*==============================================================*/
/* Index des tables                                             */
/*==============================================================*/
alter table Film add index(
    categorieID,
    realisateurID
);

alter table Jury add index(
    categorieID
);

alter table Personne add index(
    juryID
);

alter table Projection add index(
    salleID,
    filmID
);

alter table VIP add index(
    photoID,
    compagnonID
);
```

```
/*==============================================================*/
/* Clés étrangères des tables                                   */
/*==============================================================*/
alter table Film
    add constraint FK_FILM_ASSOCIATI_CATEGORI foreign key (categorieID)
        references Categorie (categorieID);

alter table Film
    add constraint FK_FILM_REALISE_PERSONNE foreign key (realisateurID)
        references Personne (personneID);

alter table Jury
    add constraint FK_JURY_CATEGORIE_CATEGORI foreign key (categorieID)
        references Categorie (categorieID);

alter table Personne
   add constraint FK_PERSONNE_ASSOCIATI_JURY foreign key (juryID)
      references Jury (juryID);

alter table Projection
    add constraint FK_PROJECTI_ASSOCIATI_FILM foreign key (filmID)
        references Film (filmID);

alter table Projection
    add constraint FK_PROJECTI_ASSOCIATI_SALLE foreign key (salleID)
        references Salle (salleID);

alter table VIP
    add constraint FK_VIP_ASSOCIATI_PHOTO foreign key (photoID)
        references Photo (photoID);

alter table VIP
    add constraint FK_VIP_ASSOCIATI_PERSONNE foreign key (personneID)
        references Personne (personneID);

/*==============================================================*/
/* Autre contraintes                                            */
/*==============================================================*/
alter table VIP
    add constraint CHK_PERSONNE CHECK
    (
        typeVIP = 'acteur'
        OR typeVIP = 'realisateur'
        OR typeVIP = 'journaliste'
        OR typeVIP = 'people'
    );
```

```
/*==============================================================*/
/* Suppression des tables                                       */
/*==============================================================*/
alter table Film
    drop foreign key FK_FILM_ASSOCIATI_CATEGORI;

alter table Film
    drop foreign key FK_FILM_REALISE_PERSONNE;

alter table Jury
    drop foreign key FK_JURY_CATEGORIE_CATEGORI;

alter table Personne
   drop foreign key FK_PERSONNE_ASSOCIATI_JURY;

alter table Projection
    drop foreign key FK_PROJECTI_ASSOCIATI_FILM;

alter table Projection
    drop foreign key FK_PROJECTI_ASSOCIATI_SALLE;

alter table VIP
    drop foreign key FK_VIP_ASSOCIATI_PHOTO;

alter table VIP
    drop foreign key FK_VIP_ASSOCIATI_PERSONNE;

drop table Categorie, Film, Jury, Personne, Photo, Projection, Salle, VIP;
```