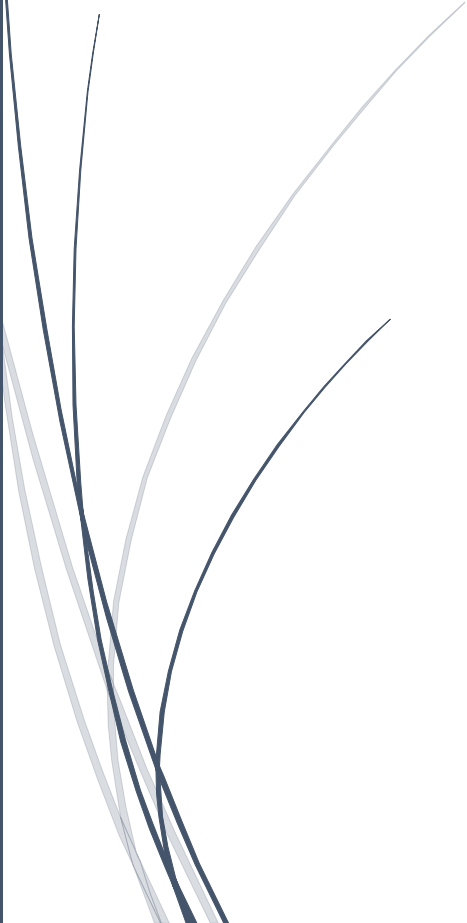


A dark blue vertical bar on the left side of the page. A blue arrow points to the right from this bar, containing the date.

17/06/2016

Rapport de projet CPOA

VOICELA application de gestion des Vip's

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

RAPPORT DE PROJET CPOA /
TOUMI Nacereddine, MAISONHAUTE Noémie
CLIENT : M.PILLOT

Table des matières

| | |
|--|----|
| PRESENTATION DU SUJET..... | 3 |
| PARTIE CONCEPTION..... | 4 |
| 1. Diagrammes de modélisation UML..... | 4 |
| a. Diagramme de Classe..... | 4 |
| b. Diagramme de Classe dégradé (efficace et utile) | 5 |
| Modele Relationnel..... | 5 |
| Diagramme de classe dégradé..... | 5 |
| c. Diagramme de Cas d'Utilisation..... | 6 |
| d. Diagramme de Séquence (Processus métier) | 7 |
| <i>Enregistrement d'une photo</i> | 7 |
| <i>Enregistrement d'une divorce</i> | 8 |
| <i>Enregistrement d'un mariage</i> | 9 |
| 2. Script SQL | 10 |
| a. Table Casting..... | 10 |
| Création de la table..... | 10 |
| Alimentation de la table..... | 10 |
| b. Table Evénement | 11 |
| Création de la table..... | 11 |
| Alimentation de la table..... | 12 |
| Triggers MySQL | 13 |
| c. Table FILM..... | 15 |
| Création de la table..... | 15 |
| Alimentation de la table..... | 16 |
| Trigger MySQL..... | 17 |
| d. Table Genre..... | 17 |
| Création de la table..... | 17 |
| Alimentation de la table..... | 18 |
| e. Table Pays | 18 |
| Création de la table..... | 18 |
| Alimentation de la table..... | 18 |
| f. Table PHOTOVIP..... | 19 |
| Création de la table..... | 19 |

| | |
|---|-----------|
| Alimentation de la table..... | 20 |
| g. Table REALISATEUR..... | 20 |
| Création de la table..... | 20 |
| Alimentation de la table..... | 21 |
| h. Table VIP | 21 |
| Création de la table..... | 21 |
| Alimentation de la table..... | 22 |
| Trigger MySQL..... | 23 |
| PARTIE DEVELOPPEMENT..... | 25 |
| Couche présentation..... | 25 |
| Connexion | 25 |
| Onglet EVENEMENT | 26 |
| Onglet PHOTO | 28 |
| Couche accès aux données de la base | 29 |
| Méthode d'intégration incrémentale et tests réalisés..... | 29 |
| Couche métier..... | 29 |
| BILAN CONDUITE DE PROJET..... | 32 |
| Découpage du projet, temps estimés/réel et explications | 32 |
| Gestion de version avec github | 33 |
| CONCLUSION GENERALE..... | 34 |
| Bilan technique | 34 |
| Bilan personnel | 35 |
| ANNEXE..... | 36 |

PRESENTATION DU SUJET

Le sujet consiste à réaliser une application n-niveaux architecturée de la façon suivante :

- une base de données relationnelle qui implémente le système d'Information de l'application
- une application client écrite en java qui offre des fonctionnalités d'administration de la base de données.
- une application client de type web qui permet d'interroger la base de données

Nous devons répondre aux demandes du journal VOICELA qui est spécialisé dans la diffusion d'information « people ». Le site web qu'on nous demande permettra de suivre la carrière du VIP, ainsi que sa vie privée. L'application permet de rajouter des données sur un VIP dans la base.

B. Phase de conception – présentation du sujet:

Le journal VOICELA est spécialisé dans la diffusion d'informations "people". Afin de permettre à ses lecteurs et à ses lectrices d'être informés au mieux sur la vie des V.I.P. (Very Important Person), la rédaction souhaite mettre en place un système d'information les concernant. Un VIP est une personne publique connue, voire célèbre très rarement issu du monde des sciences mais gravitant plutôt dans la nébuleuse du show-biz, des médias, de la mode ou encore de la politique. Cependant dans le cadre du projet proposé, on ne considérera que les personnes du monde du cinéma, à savoir les **acteurs et les réalisateurs**.

Afin de répondre aux attentes de sa clientèle toujours avide de spectaculaire, la rédaction du journal souhaite publier un site Web qui permettrait

- de connaître pour **chaque VIP du cinéma**, un certain nombre d'informations: **nom, prénoms (3 max.), nationalité, date et lieu de naissance, et son âge**.
- de connaître un certain nombre d'événements liés à la vie d'un VIP en particulier
 - **son ou ses mariages et avec pour chacun d'eux la date et le lieu de la cérémonie**
 - **la date de son ou de ses divorces si le VIP est divorcé ou bien s'il s'est remarié**
- de visualiser **les photos** sur lesquelles apparaît tel ou tel VIP afin d'agrémenter les publications

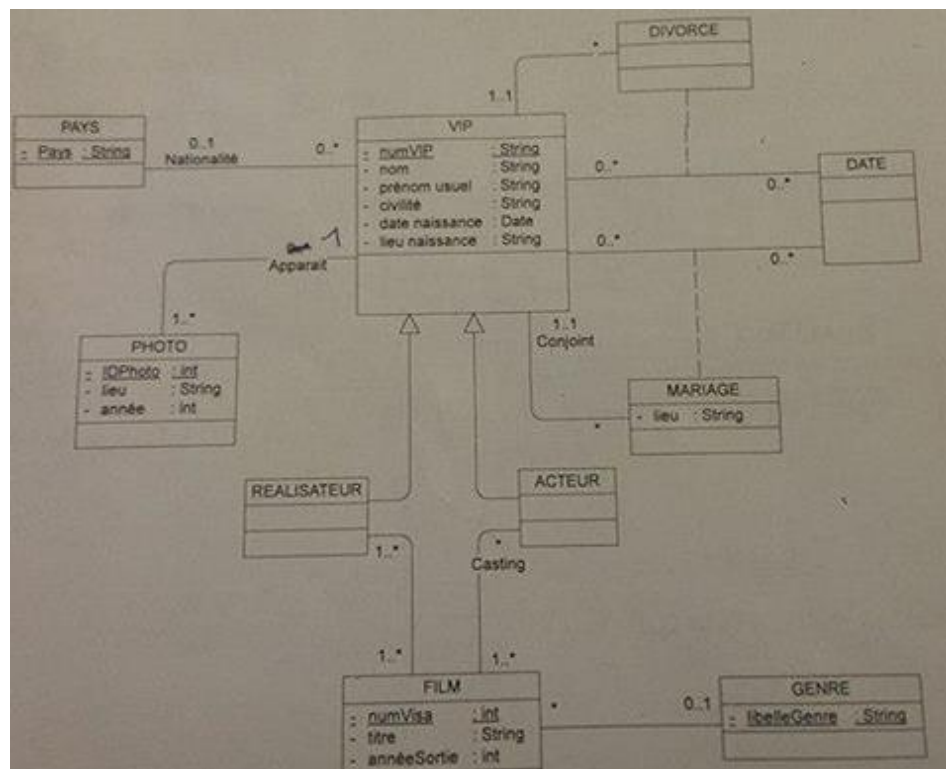
Le conjoint d'un VIP n'est pas forcément un VIP mais par soucis de simplification on considérera qu'il le devient forcément. Enfin il est possible qu'un VIP se marie plusieurs fois avec le même VIP (par exemple les mariages du couple mythique Liz Taylor et Richard Burton).

Le système d'information à mettre en place doit permettre également de suivre la carrière des VIP. Ainsi pour un comédien, il est important de connaître les films dans lesquels il a déjà tourné avec pour chaque film l'année de sortie du film, le titre du film, le genre (aventure, comédie-dramatique, road-movie, comique, etc.), et l'identité du réalisateur qui est lui-même un VIP. De même pour un réalisateur on aimerait connaître sa filmographie avec pour chaque film réalisé, toutes les informations sur le film plus la liste des acteurs principaux qui y participent. Enfin le système d'information doit également recenser les photos sur lesquelles apparaissent les VIP et qui viendront bien évidemment agrémenter le site Web.

PARTIE CONCEPTION

1. Diagrammes de modélisation UML

a. Diagramme de Classe



b. Diagramme de Classe dégradé (efficace et utile)

Modele Relationnel

PAYS (nomPays)GENRE (libelleGenre)

FILM (numVisa, titre, année, #genre)

VIP (numVip, nomVip, prenomVip, civilite, dateNaissance, lieuNaissance, codeRole, #pays, codeStatut)

PHOTOVIP (#numVip, numeroSenquentId, date, lieu)

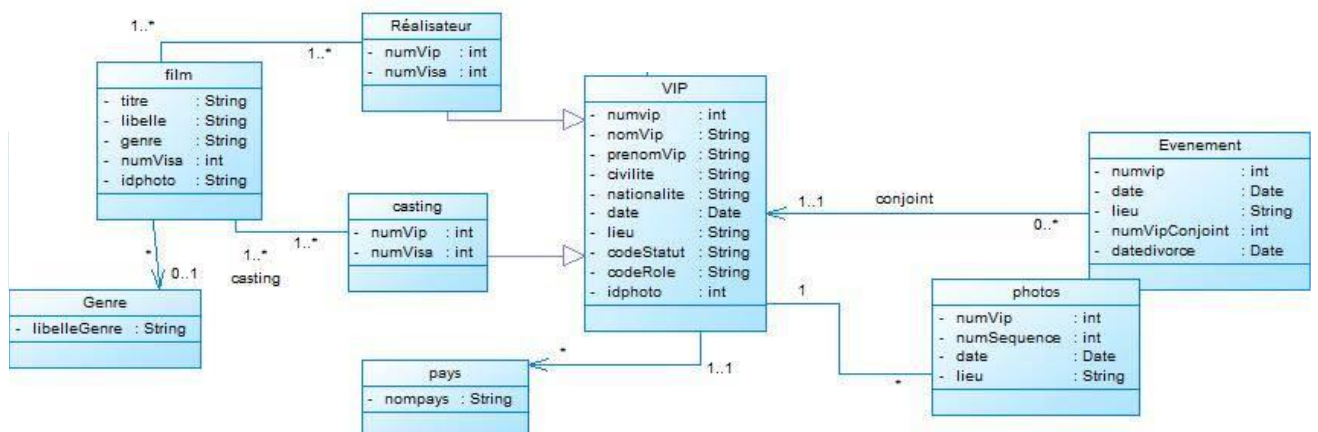
EVENEMENT (#numVip, dateMariage, numVipConjoint, lieuMariage, dateDivorce)

REALISATEUR (#numVip, #numVisa)

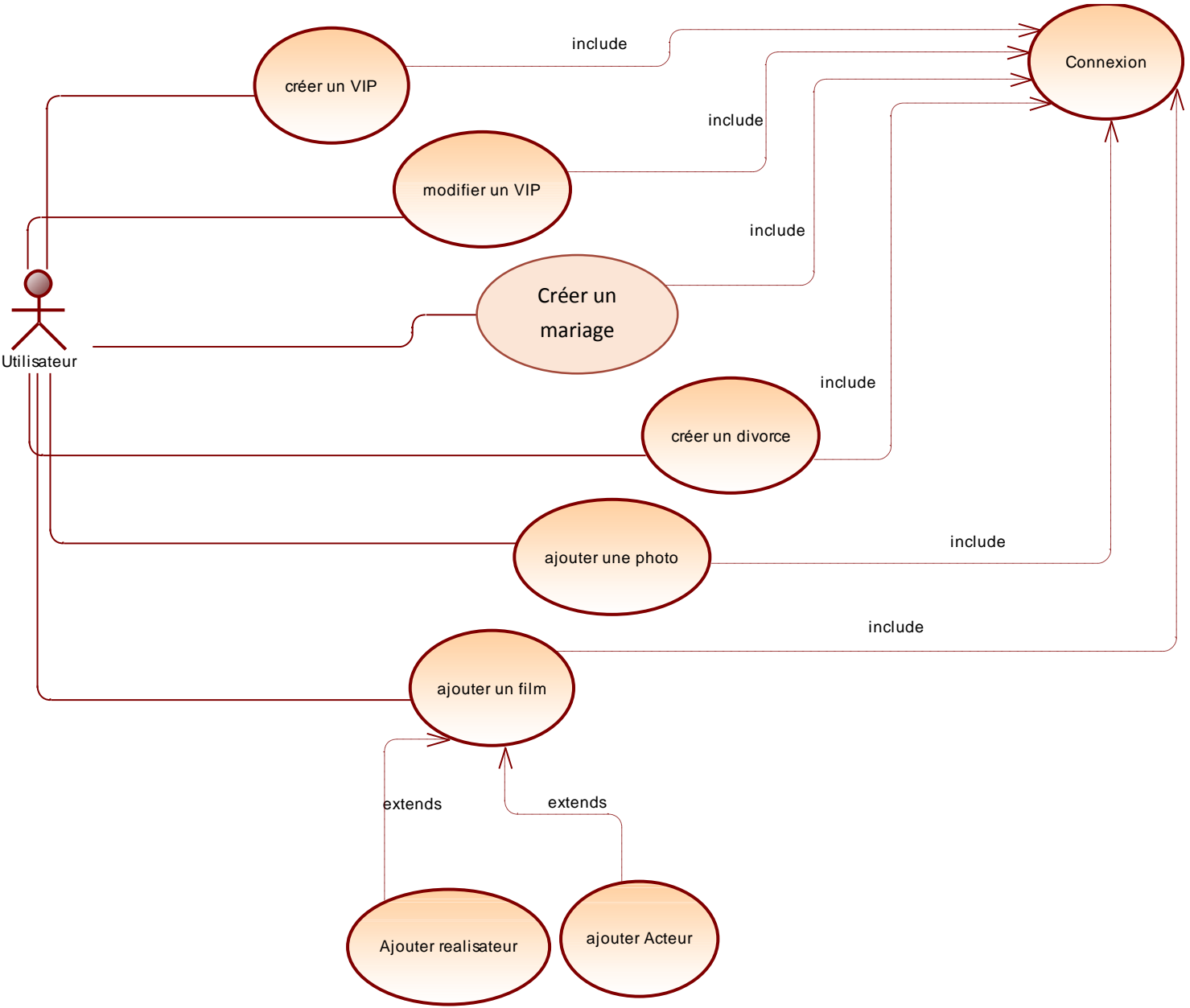
CASTING (#numVip, #numVisa)

Diagramme de classe dégradé

Ceci est le diagramme de classe dégradé de notre BD que nous avons utilisé pour la réalisation de notre Application JAVA

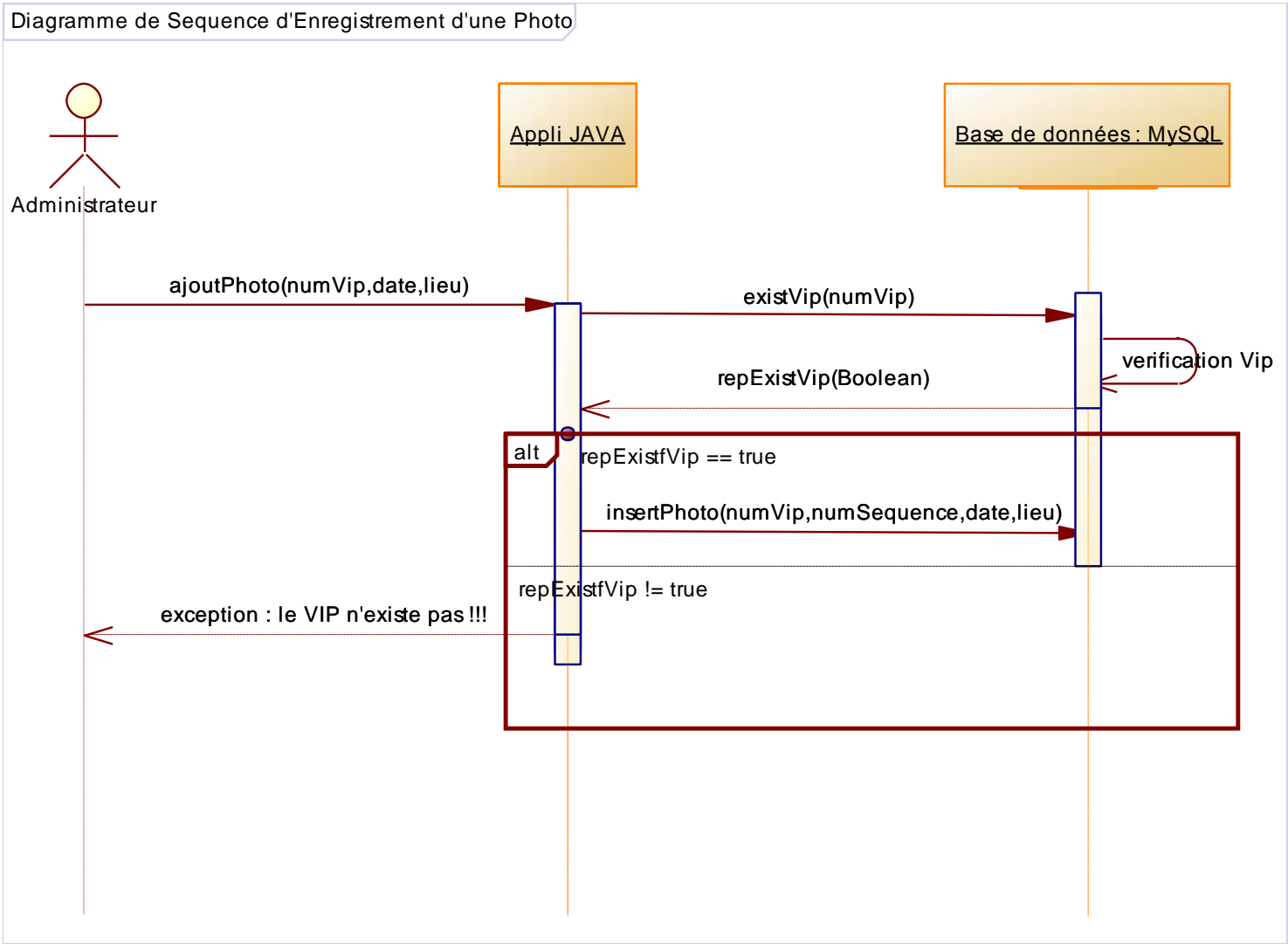


c. Diagramme de Cas d'Utilisation

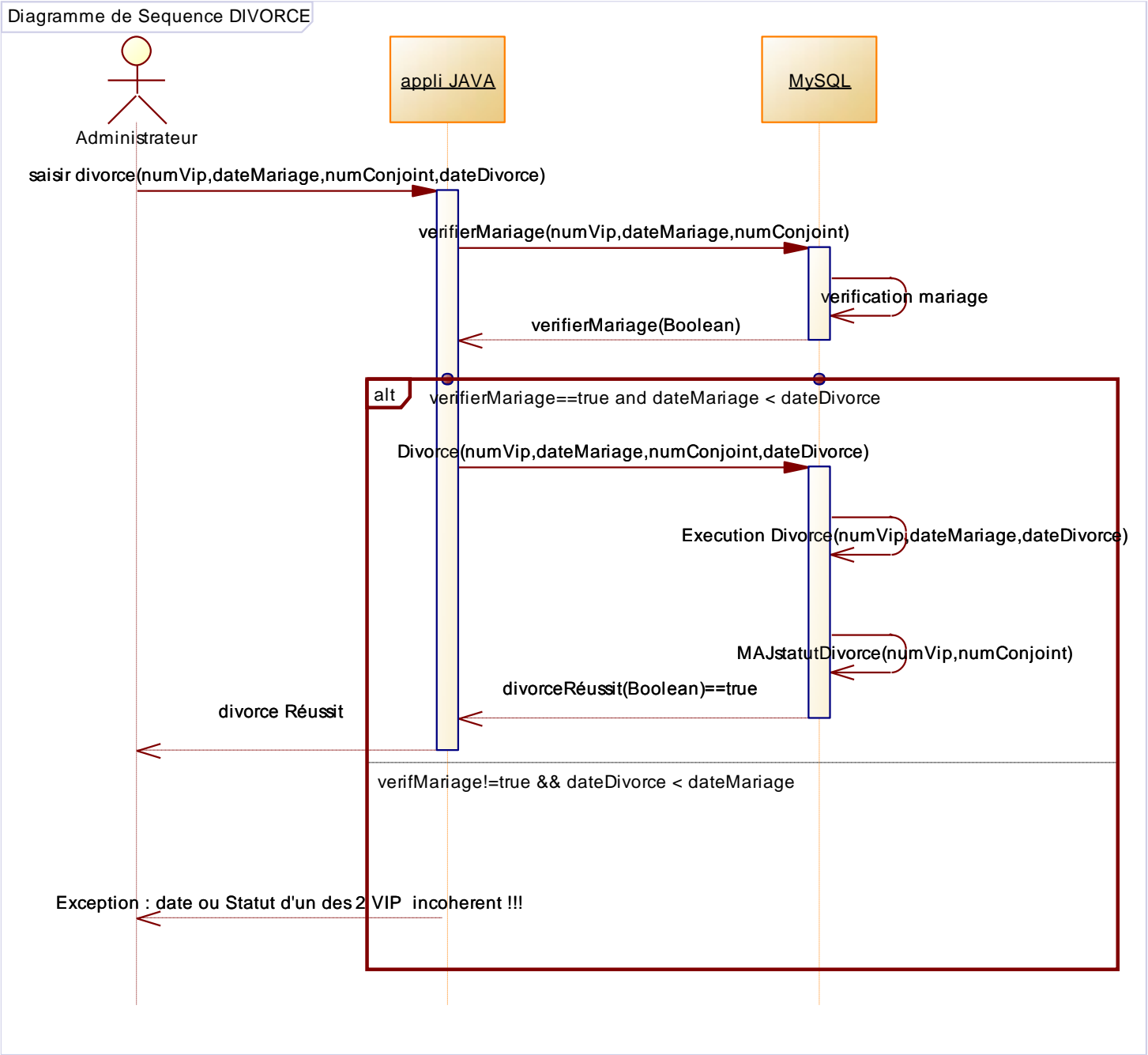


d. Diagramme de Séquence (Processus métier)

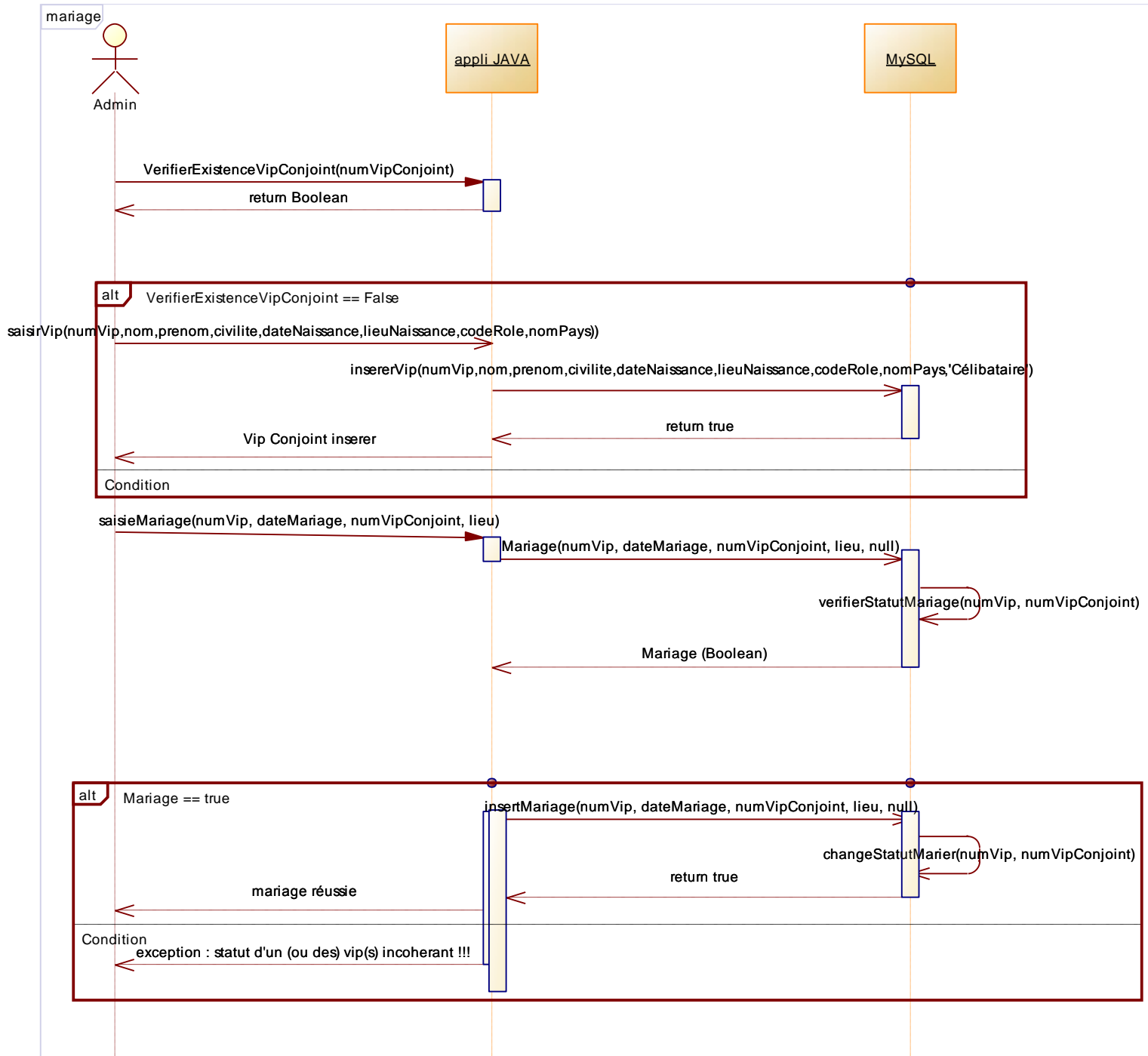
Enregistrement d'une photo



Enregistrement d’une divorce



Enregistrement d'un mariage



2. Script SQL

a. Table Casting

Création de la table

```
--  
  
-- Structure de la table `CASTING`  
  
--  
  
CREATE TABLE IF NOT EXISTS `CASTING` (  
  `numVip` int(7) NOT NULL,  
  `numVisa` int(7) NOT NULL,  
  PRIMARY KEY (`numVip`,`numVisa`),  
  KEY `fk_numvisa2` (`numVisa`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
--  
  
-- Contraintes pour la table `CASTING`  
  
--  
  
ALTER TABLE `CASTING`  
  
  ADD CONSTRAINT `fk-numvip3` FOREIGN KEY (`numVip`) REFERENCES `VIP` (`numVip`),  
  ADD CONSTRAINT `fk_numvisa2` FOREIGN KEY (`numVisa`) REFERENCES `FILM` (`numVisa`);
```

Alimentation de la table

```
INSERT INTO `CASTING` (`numVip`, `numVisa`) VALUES  
(1, 15),
```

(36, 15),
(25, 16),
(28, 1044220),
(28, 1081380),
(5, 1125520),
(26, 1132830),
(32, 1138980),
(13, 1140280),
(14, 1140280),
(35, 1196320),
(2, 1248710),
(6, 1267180),
(11, 1267180),
(13, 1319610),
(14, 1319610),
(8, 1434270),
(9, 1434270),
(28, 1434270),
(7, 1963253),
(12, 1963253),
(22, 1963253);

b. Table Evénement

Création de la table

-- Structure de la table `EVENEMENT`

```
--  
  
CREATE TABLE IF NOT EXISTS `EVENEMENT` (  
    `numVip` int(7) NOT NULL,  
    `dateMariage` date NOT NULL,  
    `numVipConjoint` int(7) NOT NULL,  
    `lieuMariage` varchar(30) NOT NULL,  
    `dateDivorce` date DEFAULT NULL,  
    PRIMARY KEY (`numVip`, `dateMariage`),  
    UNIQUE KEY `dateMariage` (`dateMariage`)  
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
--  
  
-- Contraintes pour la table `EVENEMENT`  
  
--  
  
ALTER TABLE `EVENEMENT`  
  
    ADD CONSTRAINT `fk_numvip` FOREIGN KEY (`numVip`) REFERENCES `VIP` (`numVip`);
```

Alimentation de la table

```
--  
  
-- Contenu de la table `EVENEMENT`  
  
--  
  
INSERT INTO `EVENEMENT` (`numVip`, `dateMariage`, `numVipConjoint`, `lieuMariage`, `dateDivorce`)  
VALUES  
  
    (1, '2016-06-09', 2, 'San francisco', NULL),  
  
    (4, '2013-03-05', 9, 'San Francisco', '2016-06-01'),  
  
    (4, '2016-06-16', 22, 'New York', '2016-06-18'),  
  
    (5, '2016-06-22', 6, 'londre', '2016-06-29'),  
  
    (7, '2016-06-17', 11, 'Paris', '2016-06-18'),
```

```
(34, '2016-06-19', 22, 'stanford', '2017-06-09'),  
(35, '2016-06-08', 14, 'new york', NULL);
```

Triggers MySQL

```
--
```

```
-- Déclencheurs `EVENEMENT`
```

1) *DATE_DIVORCE*

```
CREATE TRIGGER `DATE_DIVORCE` BEFORE UPDATE ON `EVENEMENT`
```

```
FOR EACH ROW BEGIN
```

```
    declare cpt1 Date;
```

```
    declare cpt2 Date;
```

```
    declare msg varchar(255);
```

```
    DECLARE DATE_DIVORCE CONDITION FOR SQLSTATE '45015';
```

```
    select MAX(OLD.dateMariage) into cpt1 from EVENEMENT where numVip=NEW.numVip;
```

```
    select OLD.dateDivorce into cpt2 from EVENEMENT where numVip=NEW.numVip;
```

```
    if(cpt2<>0) then
```

```
        CALL DATE_DIVORCE;
```

```
    END IF;
```

```
    if(NEW.dateDivorce< cpt1) then
```

```
        CALL DATE_DIVORCE;
```

```
    END IF;
```

```
END
```

2) *DIVORCE*

```
CREATE TRIGGER `DIVORCE` AFTER UPDATE ON `EVENEMENT`
```

```
FOR EACH ROW BEGIN
```

```
update VIP set codeStatut='Divorcé' WHERE numVip=NEW.numVip;  
update VIP set codeStatut='Divorcé' WHERE numVip=NEW.numVipConjoint;  
END
```

3) Mariage

```
CREATE TRIGGER `Mariage` BEFORE INSERT ON `EVENEMENT`  
FOR EACH ROW BEGIN  
  
    declare cpt1 integer;  
  
    declare cpt2 integer;  
  
    declare d1 Date;  
  
    declare d2 Date;  
  
    declare msg varchar(255);  
  
    DECLARE EXIST_Mariage CONDITION FOR SQLSTATE '45013';  
  
    DECLARE EXIST_Mariage2 CONDITION FOR SQLSTATE '45014';  
  
  
    select count(*) into cpt1 from VIP  
    where numVip = NEW.numVip and codeStatut='Marié';  
  
  
    select count(*) into cpt2 from VIP  
    where numVip=NEW.numVipConjoint and codeStatut='Marié';  
  
  
    select MAX(dateDivorce) into d1 from EVENEMENT  
    where numVip=NEW.numVipConjoint or numVipConjoint=NEW.numVipConjoint;  
  
  
    select MAX(dateDivorce) into d2 from EVENEMENT  
    where numVip=NEW.numVip or numVipConjoint=NEW.numVip;
```

```
if cpt1 <> 0 OR cpt2 <> 0 then
```

```
    CALL EXIST_Mariage;
```

```
END IF;
```

```
if d1> NEW.dateMariage OR d2>NEW.dateMariage then
```

```
    CALL EXIST_Mariage2;
```

```
END IF;
```

```
update VIP set codeStatut='Marié' where numVip=NEW.numVip;
```

```
update VIP set codeStatut='Marié' where numVip=NEW.numVipConjoint;
```

```
END
```

c. Table FILM

Création de la table

```
--  
-- Structure de la table `FILM`  
--  
CREATE TABLE IF NOT EXISTS `FILM` (  
    `numVisa` int(7) NOT NULL,  
    `titre` varchar(100) NOT NULL,  
    `LibelleGenre` varchar(50) NOT NULL,  
    `annee` int(11) NOT NULL,  
    `idPhotoF` varchar(20) NOT NULL,  
    PRIMARY KEY (`numVisa`),
```



```

KEY `LibelleGenre` (`LibelleGenre`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Contraintes pour la table `FILM`

--

ALTER TABLE `FILM`

ADD CONSTRAINT `fk-genre` FOREIGN KEY (`LibelleGenre`) REFERENCES `GENRE` (`LibelleGenre`);

```

Alimentation de la table

```

--

-- Contenu de la table `FILM`

--

INSERT INTO `FILM` (`numVisa`, `titre`, `LibelleGenre`, `annee`, `idPhotoF`) VALUES

(15, 'Maléfique', 'Fantastique', 2014, 'F15.jpg'),

(16, 'The Amazing Spider-Man', 'Fantastique', 2012, 'F16.jpg'),

(1044220, 'Alice au pays des merveilles', 'Fantastique', 2010, 'F1.jpg'),

(1081380, 'SWEENEY TODD, LE DIABOLIQUE BARBIER DE FLEET STREET', 'Fantastique', 2008, 'F2.jpg'),

(1125520, 'Nos jours heureux', 'Comédie', 2006, 'F3.jpg'),

(1132830, 'All the Boys Love Mandy Lane', 'Horreur', 2008, 'F4.jpg'),

(1138980, 'Mission : Impossible III', 'Action', 2006, 'F5.jpg'),

(1140280, 'Harry Potter et l"école des sorciers', 'Fantastique', 2001, 'F6.jpg'),

(1196320, 'V pour Vendetta', 'Thriller', 2005, 'F7.jpg'),

(1248710, 'Fight Club', 'Thriller', 1999, '8.jpg'),

(1267180, 'Hunger Games', 'Fantastique', 2012, 'F9.jpg'),

(1319610, 'HARRY POTTER ET LA COUPE DE FEU', 'Fantastique', 2005, 'F10.jpg'),

```

```
(1434270, 'Dark Shadow', 'Fantastique', 2012, 'F14.jpg'),
(1963253, 'Inception', 'Fantastique', 2010, 'F13.jpg');
```

Trigger MySQL

```
--
-- Déclencheurs `FILM`
--

CREATE TRIGGER `EXIST_FILM` BEFORE INSERT ON `FILM`
FOR EACH ROW BEGIN

    declare cpt integer;
    declare msg varchar(255);
    DECLARE EXIST_FILM CONDITION FOR SQLSTATE '45001';
    select count(*) into cpt from FILM
    where numVisa = NEW.numVisa and titre=NEW.titre and LibelleGenre=NEW.LibelleGenre
    and annee=NEW.annee;

    if cpt <> 0 then
        CALL EXIST_FILM;
    END IF;
END
-----
```

d. Table Genre

Création de la table

```
--
-- Structure de la table `GENRE`
--

CREATE TABLE IF NOT EXISTS `GENRE` (
    `LibelleGenre` varchar(50) NOT NULL,
    PRIMARY KEY (`LibelleGenre`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Alimentation de la table

```
-- Contenu de la table `GENRE`  
  
--  
  
INSERT INTO `GENRE` (`LibelleGenre`) VALUES  
  
('Action'),  
  
('Comédie'),  
  
('Fantastique'),  
  
('Horreur'),  
  
('Science Fiction'),  
  
('Thriller');
```

e. Table Pays

Création de la table

```
-- Structure de la table `PAYS`  
  
--  
  
CREATE TABLE IF NOT EXISTS `PAYS` (  
  
  `nomPays` varchar(30) NOT NULL,  
  
  PRIMARY KEY (`nomPays`)  
  
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Alimentation de la table

```
--  
  
-- Contenu de la table `PAYS`
```

```
--
```

```
INSERT INTO `PAYS` (`nomPays`) VALUES  
('D'),  
('FR'),  
('UK'),  
('US');
```

f. Table PHOTOVIP

Création de la table

```
-- Structure de la table `PHOTOVIP`  
--  
CREATE TABLE IF NOT EXISTS `PHOTOVIP` (  
  `numVip` int(7) NOT NULL,  
  `numeroSenquentId` int(7) NOT NULL,  
  `date` date NOT NULL,  
  `lieu` varchar(30) NOT NULL,  
  PRIMARY KEY (`numVip`,`numeroSenquentId`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
--  
-- Contraintes pour la table `PHOTOVIP`  
--  
ALTER TABLE `PHOTOVIP`  
  ADD CONSTRAINT `fk-numvip4` FOREIGN KEY (`numVip`) REFERENCES `VIP` (`numVip`);
```

Alimentation de la table

```
--
-- Contenu de la table `PHOTOVIP`
--

INSERT INTO `PHOTOVIP` (`numVip`, `numeroSenquentId`, `date`, `lieu`) VALUES
(37, 1, '2016-06-17', 'Lyon'),
(37, 2, '2016-06-17', 'paris');

-----
```

g. Table REALISATEUR

Création de la table

```
-- Structure de la table `REALISATEUR`
--

CREATE TABLE IF NOT EXISTS `REALISATEUR` (
  `numVip` int(7) NOT NULL,
  `numVisa` int(7) NOT NULL,
  PRIMARY KEY (`numVip`, `numVisa`),
  KEY `fk_numvisa` (`numVisa`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Contraintes pour la table `REALISATEUR`
--

ALTER TABLE `REALISATEUR`

ADD CONSTRAINT `fk_numvip2` FOREIGN KEY (`numVip`) REFERENCES `VIP` (`numVip`),
ADD CONSTRAINT `fk_numvisa` FOREIGN KEY (`numVisa`) REFERENCES `FILM` (`numVisa`);
```

Alimentation de la table

```
--  
  
-- Contenu de la table `REALISATEUR`  
  
--  
  
INSERT INTO `REALISATEUR` (`numVip`, `numVisa`) VALUES  
  
(4, 1044220),  
  
(4, 1081380),  
  
(30, 1125520),  
  
(31, 1132830),  
  
(33, 1138980),  
  
(29, 1140280),  
  
(19, 1196320),  
  
(19, 1248710),  
  
(34, 1267180),  
  
(17, 1319610),  
  
(4, 1434270);  
  
-----
```

h. Table VIP

Création de la table

```
--  
  
-- Structure de la table `VIP`  
  
--
```

```

CREATE TABLE IF NOT EXISTS `VIP` (
  `numVip` int(7) NOT NULL AUTO_INCREMENT,
  `nomVip` varchar(30) NOT NULL,
  `prenomVip` varchar(30) NOT NULL,
  `civilite` varchar(3) NOT NULL,
  `dateNaissance` date NOT NULL,
  `lieuNaissance` varchar(30) NOT NULL,
  `codeRole` varchar(30) NOT NULL,
  `nomPays` varchar(30) NOT NULL,
  `codeStatut` varchar(30) NOT NULL,
  `idPhoto` varchar(11) DEFAULT NULL,
  PRIMARY KEY (`numVip`),
  KEY `nomPays` (`nomPays`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=41 ;

```

Alimentation de la table

```
-- Contenu de la table `VIP`
```

```
--
```

```

INSERT INTO `VIP` (`numVip`, `nomVip`, `prenomVip`, `civilite`, `dateNaissance`, `lieuNaissance`, `codeRole`,
`nomPays`, `codeStatut`, `idPhoto`) VALUES

```

```
(1, 'JOLIE', 'Angelina', 'Mme', '1975-06-04', 'Los Angeles', 'Acteur', 'US', 'Marié', '1.jpg'),
```

```
(2, 'PITT', 'Brad', 'M', '1963-12-18', 'Shawnee', 'Acteur', 'US', 'Marié', 'V2.jpg'),
```

```
(4, 'BURTON', 'Tim', 'M', '1958-08-25', 'Burbank', 'Realisateur', 'US', 'Divorcé', '4.jpg'),
```

```
(5, 'SY', 'Omar', 'M', '1978-01-20', 'Trappes', 'Acteur', 'FR', 'Divorcé', '5.jpg'),
```

```
(6, 'HUTCHERSON', 'Josh', 'M', '1992-10-12', 'Union', 'Acteur', 'US', 'Divorcé', 'V6.jpg'),
```

```
(7, 'DICAPRIO', 'Leonardo', 'M', '1974-11-11', 'Hollywood', 'Acteur', 'US', 'Divorcé', 'V7.jpg'),
```

```
(8, 'MORETZ', 'Chloë', 'Mme', '1997-02-10', 'Atlanta', 'Acteur', 'US', 'Célibataire', 'V8.jpg'),
```

(9, 'CARTER', 'Helena', 'Mme', '1966-05-26', 'Londres', 'Acteur', 'UK', 'Divorcé', '9.jpg'),
 (11, 'LAWRENCE', 'Jennifer', 'Mme', '1990-08-15', 'Louisville', 'Acteur', 'UK', 'Divorcé', 'V11.jpg'),
 (12, 'COTILLARD', 'Marion', 'Mme', '1975-09-30', 'Paris', 'Acteur', 'FR', 'Célibataire', 'V12.jpg'),
 (13, 'WATSON', 'Emma', 'Mme', '1990-04-15', 'Paris', 'Acteur', 'UK', 'Célibataire', 'V13.jpg'),
 (14, 'RADCLIFFE', 'Daniel', 'M', '1989-07-23', 'Londres', 'Acteur', 'UK', 'Marié', 'V14.jpg'),
 (17, 'NEWELL', 'Mike', 'M', '1942-03-28', 'St Albans', 'Realisateur', 'UK', 'Célibataire', 'V17.jpg'),
 (19, 'SPIELBERG', 'Steven', 'M', '1946-11-18', 'Ohio', 'Realisateur', 'US', 'Célibataire', 'V19.jpg'),
 (22, 'Page', 'Ellen', 'Mme', '1987-02-21', 'Nouvelle Ecosse', 'Acteur', 'US', 'Divorcé', '22.jpg'),
 (25, 'STONE', 'Emma', 'Mme', '1988-11-06', 'Arizona', 'Acteur', 'US', 'Célibataire', 'V25.jpg'),
 (26, 'HEARD', 'AMBER', 'Mme', '1986-04-22', 'Texas', 'Acteur', 'US', 'Célibataire', 'V26.jpg'),
 (28, 'Deep', 'Johnny', 'M', '1963-06-09', 'Owensbor', 'Acteur', 'US', 'Célibataire', 'V28.jpg'),
 (29, 'Colombu', 'Chris', 'M', '1958-06-10', 'PennSylvanie', 'Realisateur', 'US', 'Célibataire', 'V29.jpg'),
 (30, 'Toledano', 'Eric', 'M', '1971-08-03', 'Paris', 'Realisateur', 'FR', 'Celibataire', 'V30.jpg'),
 (31, 'Levine', 'Jonathan', 'M', '1976-06-18', 'New York', 'Realisateur', 'US', 'Célibataire', 'V31.jpg'),
 (32, 'Cruise', 'Tom', 'M', '1962-07-03', 'Syracuse', 'Acteur', 'US', 'Célibataire', 'V32.jpg'),
 (33, 'Abrams', 'J.J.', 'M', '1966-06-27', 'New York', 'Realisateur', 'US', 'Célibataire', 'V33.jpg'),
 (34, 'Ross', 'Gary', 'M', '1956-11-03', 'Los Angeles', 'Realisateur', 'US', 'Divorcé', 'V34.jpg'),
 (35, 'Portman', 'Nathalie', 'Mme', '1981-10-09', 'Jérusalem', 'Acteur', 'US', 'Marié', 'V35.jpg'),
 (36, 'Fanning', 'Elle', 'Mme', '1998-04-10', 'Géorgie', 'Acteur', 'US', 'Celibataire', 'V36.jpg'),
 (37, 'CHAPLIN', 'charlie', 'M', '1930-06-07', 'New York', 'Acteur/Realisateur', 'US', 'Célibataire', NULL);

Trigger MySQL

```
-- Déclencheurs `VIP`

--

DROP TRIGGER IF EXISTS `EXIST_VIP`;

DELIMITER //

CREATE TRIGGER `EXIST_VIP` BEFORE INSERT ON `VIP`

FOR EACH ROW BEGIN
```



```
        declare cpt integer;

        declare msg varchar(255);

        DECLARE EXIST_VIP CONDITION FOR SQLSTATE '45000';

        select count(*) into cpt from VIP

        where nomVip = NEW.nomVip and prenomVip=NEW.prenomVip and dateNaissance=NEW.dateNaissance
        and lieuNaissance=NEW.lieuNaissance;

        if cpt <> 0 then

            CALL EXIST_VIP;

        END IF;

    END

--

-- Contraintes pour la table `VIP`

--

ALTER TABLE `VIP`

    ADD CONSTRAINT `fk_nomPays` FOREIGN KEY (`nomPays`) REFERENCES `PAYS` (`nomPays`);
```

PARTIE DEVELOPPEMENT

Couche présentation

Connexion

Tout d'abord nous avons la fenêtre connexion qui va permettre à l'administrateur se connecter et de sécuriser l'accès à son appli. Si il y a une tentative de connexion par une autre personne ne connaissant pas le code ou par l'administrateur lui-même une boîte de dialogue l'avertie.

Onglet VIP

Une fois l'accès accordé on arrive sur la fenêtre principale qui fonctionne avec un système d'onglet permettant à l'administrateur de passer facilement d'un onglet à l'autre pour effectuer différentes tâches sur un Vip.

| numVip | nomVip | prenom... | civilite | dateNai... | lieuNais... | codeRole | nomPays | codeSt... |
|--------|------------|-----------|----------|------------|-------------|-------------|---------|-------------|
| 1 | JOLIE | Angelina | Mme | 1975-06... | Los Ang... | Acteur | US | Marié |
| 2 | PITT | Brad | M | 1963-12... | Shawnee | Acteur | US | Marié |
| 4 | BURTON | Tim | M | 1958-08... | Burbank | Realisat... | US | Divorcé |
| 5 | SY | Omar | M | 1978-01... | Trappes | Acteur | FR | Divorcé |
| 6 | HUTCH... | Josh | M | 1992-10... | Union | Acteur | US | Divorcé |
| 7 | DICAPRIO | Leonardo | M | 1974-11... | Hollywood | Acteur | US | Divorcé |
| 8 | MORETZ | Chloé | Mme | 1997-02... | Atlanta | Acteur | US | Célibataire |
| 9 | CARTER | Helena | Mme | 1966-05... | Londres | Acteur | UK | Divorcé |
| 11 | LAWREN... | Jennifer | Mme | 1990-08... | Louisville | Acteur | UK | Divorcé |
| 12 | COTILLA... | Marion | Mme | 1975-09... | Paris | Acteur | FR | Célibataire |
| 13 | WATSON | Emma | Mme | 1990-04... | Paris | Acteur | UK | Célibataire |
| 14 | RADCLIFFE | Daniel | M | 1989-07... | Londres | Acteur | UK | Marié |
| 17 | NEWELL | Mike | M | 1942-03... | St Albans | Realisat... | UK | Célibataire |
| 19 | SPIELBERG | Steven | M | 1946-11... | Ohio | Realisat... | US | Célibataire |
| 22 | Page | Ellen | Mme | 1987-02... | Nouvelle... | Acteur | US | Divorcé |
| 25 | STONE | Emma | Mme | 1988-11... | Arizona | Acteur | US | Célibataire |

Buttons: Insérer, Modifier un VIP

Ajouter un vip il suffit de cliquer sur le bouton insérer dans l'onglet vip, ensuite de remplir les champs et enfin valider pour que le vip soit insérer dans la vue ainsi que la table.

The screenshot shows a web application interface for managing VIPs. A modal dialog titled 'Modifier un VIP' is open, displaying the following fields:

- Nom VIP: VIN
- Prenom VIP: diesel
- Civilité: M (dropdown)
- Date de Naissance: 0-06-1978 (calendar icon)
- Lieu de Naissance: CALIFORNIE
- Code Role: Acteur/Realisateur (dropdown)
- Nom Pays: US
- Buttons: Valider

In the background, a table lists VIPs. The selected row (highlighted in blue) is:

| ID | Nom | Prenom | Civilité | Date de Naissance | Lieu de Naissance | Code Role | Nom Pays | Statut |
|----|---------|---------|----------|-------------------|-------------------|-------------|----------|-------------|
| 37 | CHAPLIN | charlie | M | 1950-00-00 | NEW YORK | Acteur/R... | US | Célibataire |
| 38 | VIN | diesel | M | 1978-06-00 | californie | Acteur/R... | US | Célibataire |

Buttons 'Insérer' and 'Modifier un VIP' are visible below the table.

Modifier un Vip : il suffit de sélectionner un vip dans la liste(en bleu), ensuite d'appuyer sur le bouton modifier un VIP, en suit une fenêtre ayant récupéré toutes les caractéristiques de celui-ci (Californie était en minuscule au départ je l'ai mis en majuscule pour tester la modification).Il vous suffira ensuite de modifier les champs voulu et de valider votre saisie (bouton valider) pour que la modification est bien était effectuer, voir ci-dessous.

Onglet EVENEMENT

Passons dans l'onglet évènement, celui-ci vous permet de gérer les mariages et divorce de vip.

The screenshot shows the 'VOICELA' application interface. The 'EVENEMENT' tab is selected, displaying a table of events:

| numVip | dateMariage | numVipConjoint | lieuMariage | dateDivorce |
|--------|-------------|----------------|---------------|-------------|
| 1 | 2016-06-09 | 2 | San Francisco | |
| 4 | 2013-03-05 | 9 | San Francisco | 2016-06-01 |
| 4 | 2016-06-16 | 22 | New York | 2016-06-18 |
| 5 | 2016-06-22 | 6 | londre | 2016-06-29 |
| 7 | 2016-06-17 | 11 | Paris | 2016-06-18 |
| 34 | 2016-06-19 | 22 | stanford | 2017-06-09 |
| 35 | 2016-06-08 | 14 | new york | |

A modal dialog titled 'Inserer un Mariage' is open, displaying the following fields:

- Numéro VIP: 4
- Date Mariage: 21-06-2018 (calendar icon)
- Num VIP Conjoint: 34
- Lieu Mariage: paris
- Buttons: Valider, Défaire une Union

Mariage : Pour les mariages il faudra au préalable consulter la liste des vips dans l'onglet Vip sélectionner 2 vip avec leur numVip en regardant leur code statut pour voir si celui-là est différent de marié. Une fois cela fait retourner sur l'onglet évènement cliquer sur le bouton insérer un mariage remplir les champs et valider. Le résultat est ci-dessous.

| | | | | |
|---|------------|----|----------|------------|
| 4 | 2010-00-10 | 22 | NEW YORK | 2010-00-10 |
| 4 | 2018-06-21 | 34 | paris | |
| 5 | 2016-06-22 | 6 | londre | 2016-06-29 |

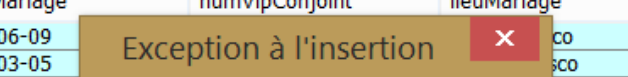
The screenshot shows the application interface with three main components:

- VIP Table:** A table with 4 columns: ID, Date, City, and Name. It contains 4 rows of data.

| 4 | 2016-06-18 | New York | 2016-06-18 |
|----|------------|----------|------------|
| 4 | 2018-06-21 | paris | |
| 5 | 2016-06-22 | londre | 2016-06-29 |
| 7 | 2016-06-17 | Paris | 2016-06-18 |
| 34 | 2016-06-19 | stanford | 2017-06-09 |
| 35 | 2016-06-08 | new york | |
- Insert Marriage Dialog:** A modal dialog box titled "Insérer un Mariage" with the following fields:
 - Número VIP: 4
 - Date Mariage: 10-06-2016
 - Num VIP Conjoint: 34
 - Lieu Mariage: nantes
 Buttons: "Insérer un Mariage" (blue), "Défaire une Union" (grey), and "Valider" (grey).
- Date Table:** A table with 2 columns: Date and Name. It contains 3 rows of data.

| dateMariage | 2016-06-18 |
|-------------|------------|
| | 2016-06-09 |
| | 2013-03-05 |

Mariage : si l'on veut insérer un mariage avec une personne déjà marié ou avec une date de mariage inferieur a la plus grande des dernières dates de divorce des 2 vips l'application JAVA affichera une exception refusant notre action et nous en informant dans une boite de dialogue.



The screenshot shows a database table with columns: dateMariage, numVipConjoint, lieuMariage, and d. The table contains several rows of data. An exception dialog box titled "Exception à l'insertion" is displayed over the table. The dialog box has a yellow warning icon and an "OK" button. The row with dateMariage "2018-06-21" is highlighted in blue.

| dateMariage | numVipConjoint | lieuMariage | d |
|-------------|----------------|-------------|----|
| 2016-06-09 | | co | |
| 2013-03-05 | | sco | 20 |
| 2016-06-16 | | | 20 |
| 2018-06-21 | | | |
| 2016-06-22 | | | 20 |
| 2016-06-17 | | | 20 |
| 2016-06-19 | | | 20 |
| 2016-06-08 | | | |

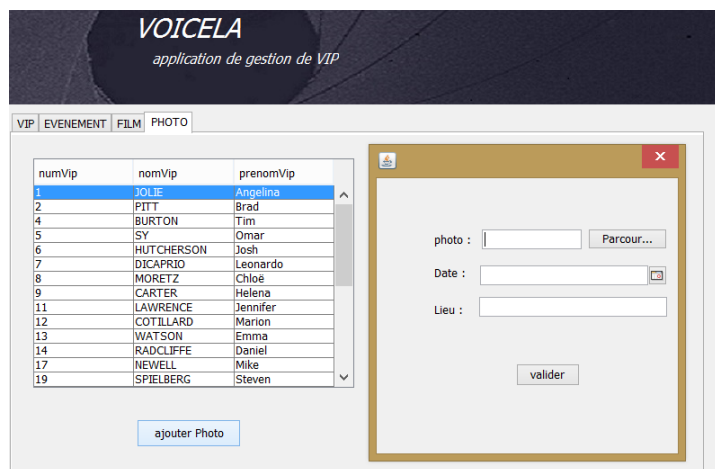
| | 2010-00-10 | 22 | new york | 2010-00-10 |
|----|------------|----|----------|------------|
| 2 | 2018-06-21 | 34 | paris | |
| 5 | 2016-06-22 | 6 | londre | 2016-06-29 |
| 7 | 2016-06-17 | 11 | Paris | 2016-06-18 |
| 14 | 2016-06-19 | 22 | stanford | 2017-06-09 |
| 15 | 2016-06-08 | 14 | new york | |

Divorce : : si l'on veut divorcer 2 personne déjà marié il suffit de cliquer sur la ligne concernée (en bleu) de rentrer la date et de valider. Si l'on veut divorcer 2 personne déjà marié avec une date de divorce inferieur a leur date de Mariage l'application JAVA affichera une exception refusant notre action et nous en informant dans une boite de dialogue. De meme si l'on veut divorcer 2 vip qui le sont déjà.

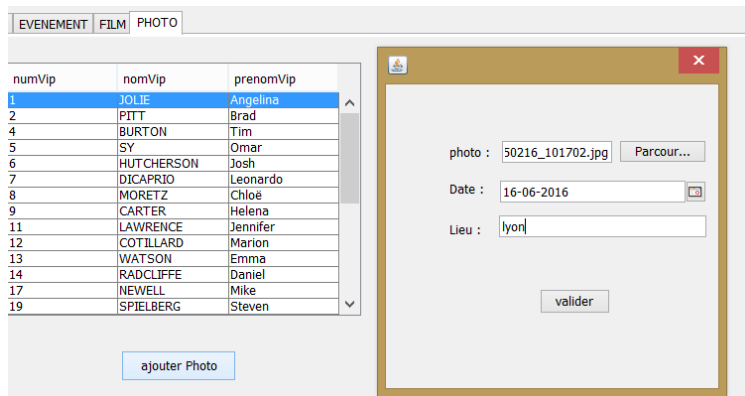
Si l'application ne relève pas d'exception alors le divorce aura bien lieu voir ci-dessous :

| | 2018-08-10 | 22 | NEW YORK | 2018-08-10 |
|---|------------|----|----------|------------|
| 4 | 2018-06-21 | 34 | paris | 2018-08-07 |

Onglet PHOTO



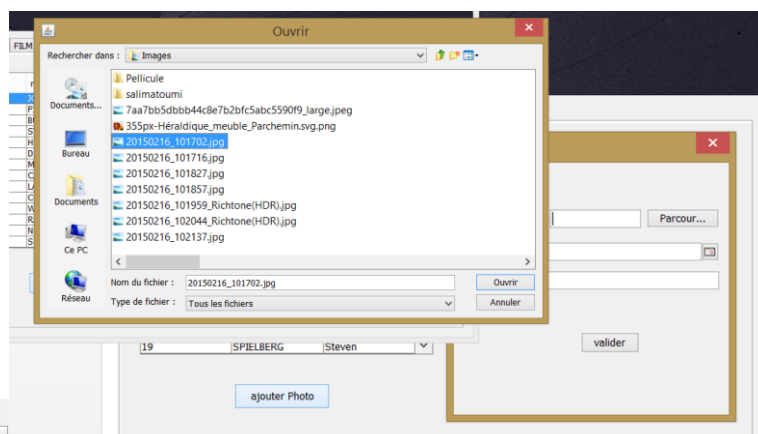
Une fois cela effectuer on peut voir que la photo est récupéré dans le champ, il vous suffit ensuite de remplir tous les champs restants et de valider (ci-dessous).



Comme on peut le voir à droite l'ajout de la photo à été un succès :

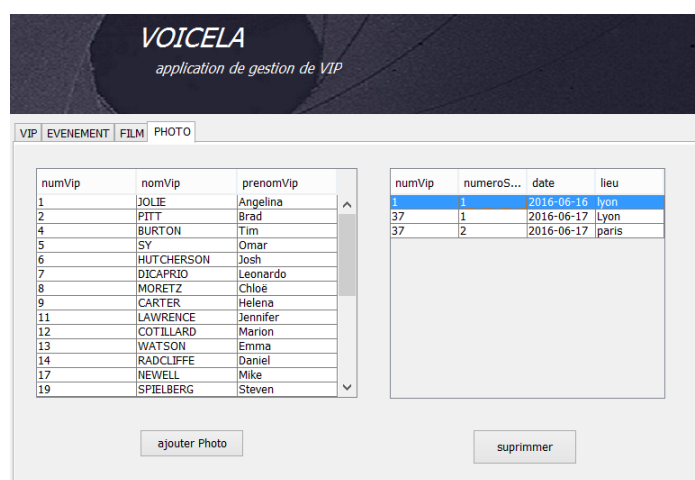
- Ajout de la photo dans la base de données (voir à droite en bleu).
- Ajout de la photo dans le serveur au chemin destination donnée (voir ci-dessous).

Ajouter une Photo : Il suffit de sélectionner le vip pour lequel vous souhaitez ajouter une photo dans la liste des vip (en bleu à gauche) et d'appuyer sur « ajouter Photo ». Une fois cela fait une boîte de dialogue s'ouvre, ensuite appuyer sur parcourir, rechercher et sélectionner la photo que vous voulez ajouter (ci-dessous).



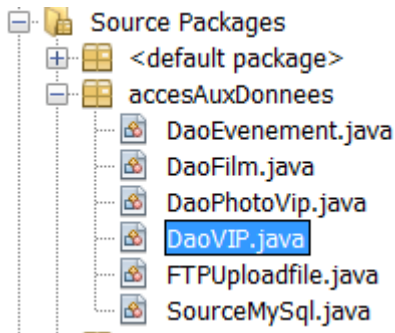
Ensuite l'application va vérifier la connexion au serveur et va upload votre photo dans celui-ci au chemin destination que vous avez initialisé (succès de l'upload voir ci-dessous).

```
numvip : 1
laPhotoVip :metier.PhotoVip@109f0dfe
User logged in successfully !
Début de l'upload
Le fichier /public_html/ a été transféré avec succès
```



| | | | |
|---------|------------------|------------|----------|
| 1-1.jpg | 16/02/2015 10:17 | Image JPEG | 2 558 Ko |
|---------|------------------|------------|----------|

Couche accès aux données de la base



La couche accès aux données de la base contient plusieurs Dao's dans lesquelles

Il y a toutes les requêtes appelant la base de données ou modifiant celle-ci, concernant les tables utilisées dans l'appli JAVA.

Exemple : le Dao Evènement fait des actions sur la table EVENEMENT situé dans la base de données.

Même choses pour les autres Dao's.

Méthode d'intégration incrémentale et tests réalisés

```
try {
    VIP vip = new VIP();
    FenetreSaisie laSaisie = new FenetreSaisie(this, vip, leModeleJComboBoxPays);
    System.out.print(laSaisie.doModal());
    if (laSaisie.doModal() == true) {
        int numVip = leModele.maxNumVip();
        vip.setNumVip(numVip);
        vip.setCodeStatut("Célibataire");
        System.out.print("\n\n"+vip);
        leModele.insererEmploye(vip);
    }
} catch (Exception e) {
```

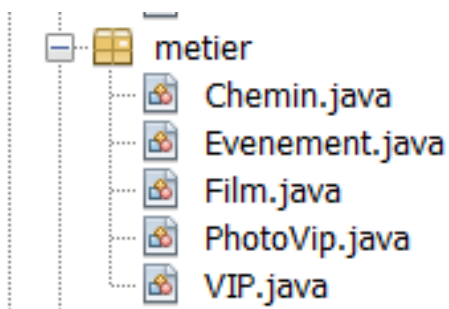
Exemple avec l'ajout d'un Vip on peut voir le bon fonctionnement de la fonctionnalité qui passe par les endroits attendus.

Ne connaissant pas très bien les méthodes d'intégration et n'ayant pas les compétences dans celles-ci nous avons décidé de faire des tests après chaque fonctionnalité pour tester le bon fonctionnement de celle-ci en établissant plusieurs scénarios et en faisant des systèmes out dans les endroits jugés importants selon les scénarios joués.

st Results Output - TPJTableMysql (run) ×

```
run:
true
VIP{numVip=40, nomVip=b, prenomVip=b, civilite=M, dateNaissance=2016-06-17, lieuNaissance=b, codeRole=Acteur, nomPays=UK, codeStatut=Célibataire};
```

Couche métier



La couche métier contient toutes les classes métiers nécessaires la bonne gestion des Vips, des fonctionnalités présentes dans l'application JAVA.

Dans chacune de ces classes métier on retrouve toutes les méthodes nécessaires à la création d'objet de cette classe (constructeurs), ainsi que celles qui nous permettent de modifier et de récupérer des éléments concernant cet objet (getters et setters).

Exemple avec la classe métier PhotoVip :

```
public class PhotoVip {
    int numVip;
    int numeroSequentid;
    Date date;
    String lieu;

    public PhotoVip(int numVip, int numeroSequentid, Date date, String lieu) {
        this.numVip = numVip;
        this.numeroSequentid = numeroSequentid;
        this.date = date;
        this.lieu = lieu;
    }

    public PhotoVip() {
    }

    public int getNumVip() {
        return numVip;
    }

    public int getNumeroSequentid() {
        return numeroSequentid;
    }

    public Date getDate() {
        return date;
    }

    public String getLieu() {
        return lieu;
    }

    public void setNumVip(int numVip) {
        this.numVip = numVip;
    }

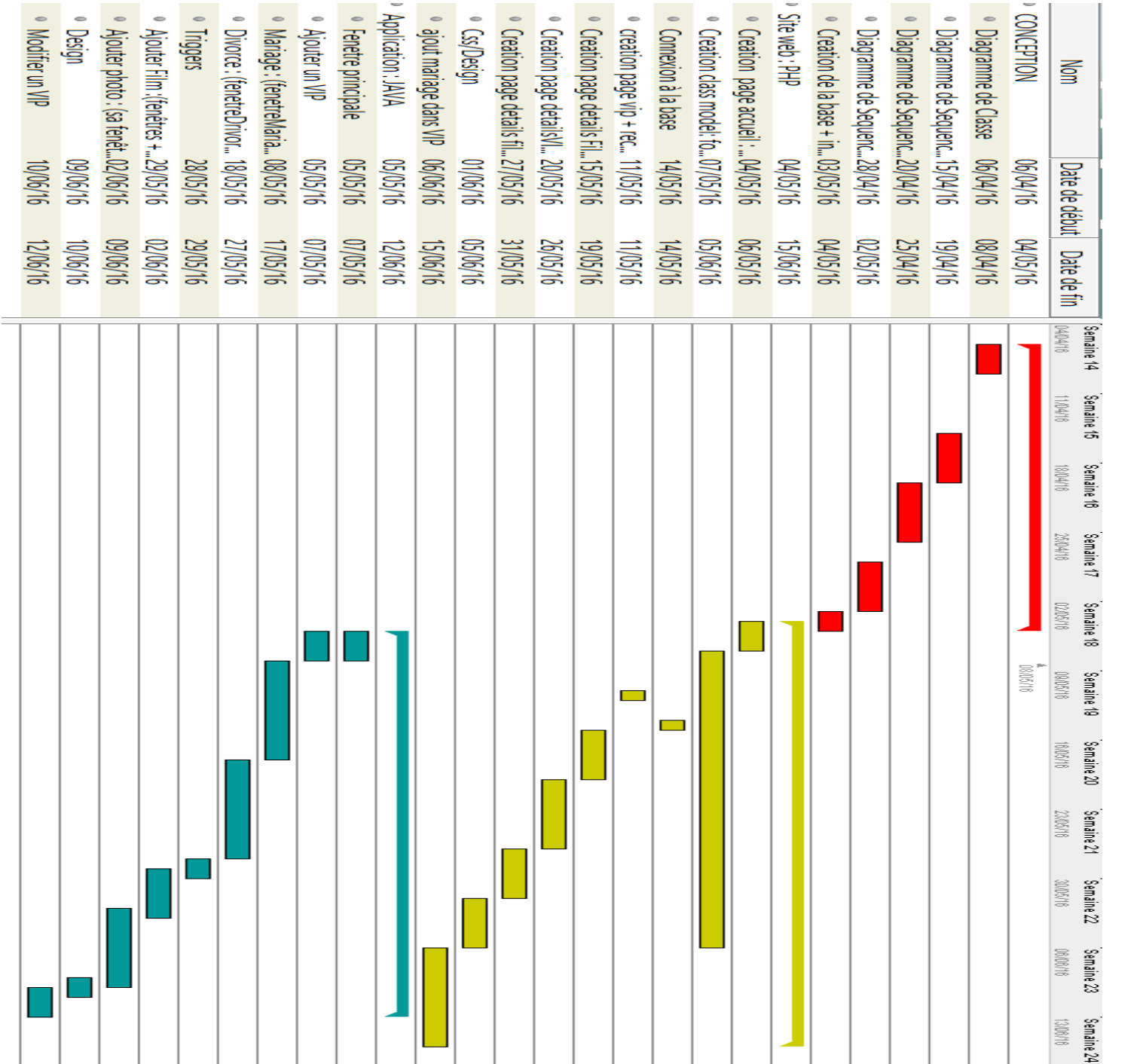
    public void setNumeroSequentid(int numeroSequentid) {
        this.numeroSequentid = numeroSequentid;
    }

    public void setDate(Date date) {
        this.date = date;
    }

    public void setLieu(String lieu) {
        this.lieu = lieu;
    }
}
```

BILAN CONDUITE DE PROJET

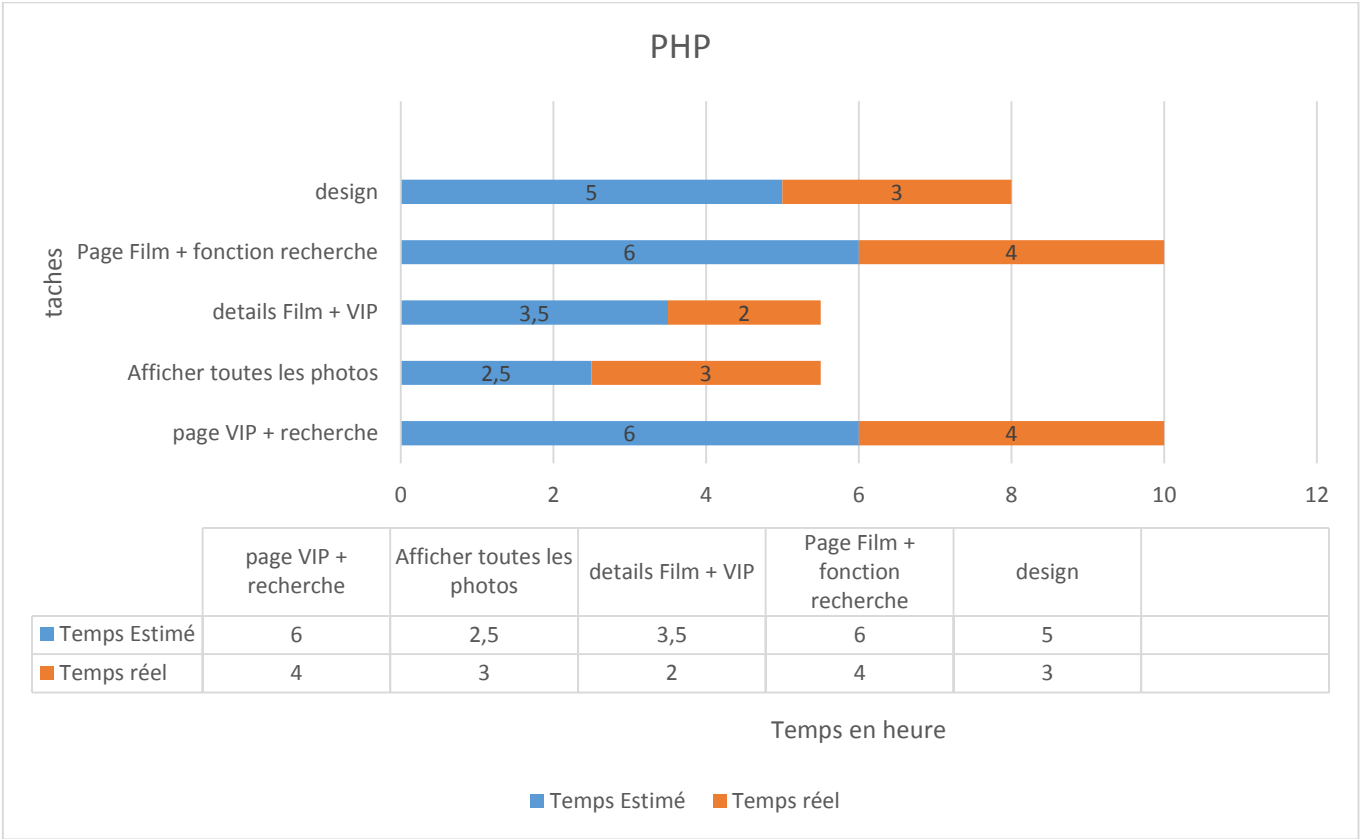
Le partage des tâches a été fait assez facilement en fonction des qualités de chacun : TOUMI Nacereddine s’est occupé du développement en Java de l’application VOICELA ainsi que de la partie gestion de projet et de la révision des diagrammes UML et MAISONHAUTE Noémie de la partie site Web(PHP) et des bases de données nécessaires aux deux applications au fur et à mesure de l’avancement de celles-ci ainsi que du Design.



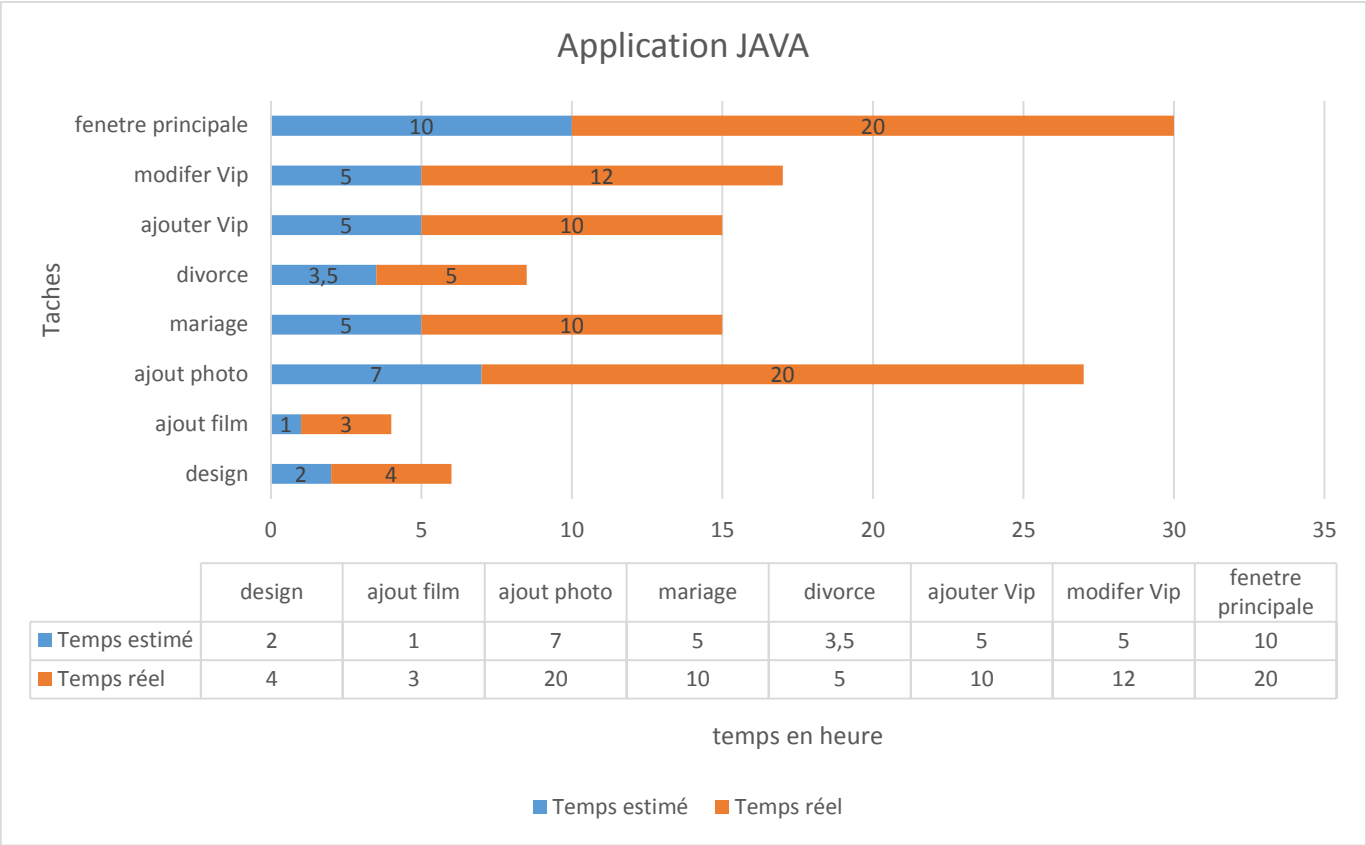
Découpage du projet, temps estimés/réel et explications

Nous avons décidé de vraiment séparer les tâches car nous n’avions pas le même niveau ni la même manière de coder dans ces différents langages, on se serait donc plus ralenti à vouloir travailler en même temps sur une même application. Nous pensons que cette répartition des tâches a été bénéfique au niveau temps et a permis d’éviter de « se marcher dessus » pendant la programmation.

De plus nous avons utilisé une méthode agile pour développer ces deux applications, nous avons fixé des objectifs à réaliser étapes par étapes plutôt que de vouloir directement terminer l’application de manière désordonnée au risque d’oublier des éléments.



Pour la partie PHP les dépassements des délais estimés sont en majeure partie dû aux différentes modifications de la base qui ont été effectuées pour une optimisation et un meilleur fonctionnement de l’appli Java, mais aussi dû à la non connaissance de la réalité du volume horaire, de toutes les étapes et tâches que nécessite un projet car celui-ci fut notre 1^{er} réel projet élaborer du début jusqu’à la fin.



Pour la partie JAVA les dépassements des délais estimés sont en majeure partie dû à la non connaissance de la réalité du volume horaire, de toutes les étapes et taches que nécessite un projet car celui-ci fut notre 1^{er} réel projet élaborer du début jusqu'à la fin. Mais aussi dû aux manques de connaissances et de compétences car nous n'étions que des apprentis programmeur pas encore assez apte et n'ayant pas assez d'expériences de projet derrière nous pour maitriser cela.

Gestion de version avec github

Durant ce projet de CPOA nous avons décidé d'utiliser github comme outils de gestion de version car git en local c'est bien mais si l'on veut partager les données quand on travail en équipes github c'est mieux. Il a permis à tous les membres du groupe d'avoir le code déjà effectué à tout moment et de contribuer au projet chacun de notre côté et à tout temps.

CONCLUSION GENERALE

Bilan technique

Partie JAVA :

On a appris beaucoup au niveau des modèles, et du Dao, comment utiliser un modèle de JTable, de JListe ainsi que de JComboBox, ...

Utilisation de fonctions dans le DAO donc on a appris aussi à gérer le SQL.

Les liens entre différentes fenêtres et la transition de données.

Utilisation et Upload de photos avec FTP.

Partie PHP :

On a appris à mieux utiliser le modèle MVC (modèle/vue/contrôleur)

On a appris à améliorer nos compétences en PHP

On a appris aussi beaucoup de nouvelle chose au niveau html (exemple : utilisation du display block, savoir jouer avec les mantes)

Bonne réalisations :

Utilisation du modèle MVC

Utilisation des modèles pour améliorer l'utilisation de l'application par le client

Utilisation du FTP pour upload les photos

Bilan, on a :

Evoluer (expérience, connaissances, compétences).

Appris à gérer un projet.

Appris à se faire confiance (travail d'équipe).

Bilan personnel

TOUMI :

J'ai trouvé ce projet vraiment intéressant. Tout d'abord, parce que c'est la première fois à l'IUT que nous avons vraiment appliqué toutes nos nouvelles connaissances dans un projet (gestion de projet, UML, différents langages, gestionnaire de version, ...).

De plus ce projet m'a permis de m'améliorer énormément sur le langage JAVA. Comme nous l'avons dit plus haut, nous avons utilisé le modèle MVC, et j'ai pu me rendre compte à quel point celui-ci était important dans les projets de grande, mais aussi la partie conception qui elle est aussi une partie très importante dans un projet, et que bien la faire nous fera gagné énormément de temps au niveau du développement. J'ai pu découvrir la quantité de travail que demande un vrai projet. Même en ayant travaillé pendant les week-ends, le projet a été difficile à finir à temps.

Je pense que ce projet m'a beaucoup apporté et que je pourrais réutiliser ces nouvelles connaissances pendant mon stage et plus tard.

MAISONHAUTE :

Je tire de ce projet un bilan positif, en effet il nous montre comment se déroule un projet : de l'étape de la lecture du cahier des charges jusqu'au déploiement des applications. D'un point de vue personnel ce module m'a permis de me rendre compte de l'importance de la partie analyse d'un projet, dont je sous-estimais (à tort) sa nécessité. En effet nous aurions perdu énormément de temps à cause d'une incompréhension du cahier des charges dans le projet si il était mal défini, mais une conception réalisée dans les règles de l'art facilite beaucoup le développement. Pour la partie web, j'ai maintenant des connaissances solides avec celui-ci qui me serviront dans mon futur.

ANNEXE

- Fichier du Code Commenté