

NFE102 Infrastructures pour le commerce électronique**Définition de projet****Gestion d'un vidéoclub**

Initiateur	Eric Boniface
------------	---------------

Num. version	1.2
Statut du document	Prod
Date	23/02/2016

Historique

Version	Date	Note de révision
1.0	05/03/2014	Première version
1.1	23/03/2015	Relecture et modifications
1.2	23/02/2016	Révision 2016

Sommaire

1.	Synopsis	3
2.	Contexte	4
2.1	Le vidéoclub	4
2.2	Le besoin	4
3.	Le projet	5
3.1	Objectifs	5
3.2	Fonctionnalités	5
3.3	Prérequis et contraintes	5
4.	Détail des fonctionnalités	6
4.1	Gestion des films	6
4.2	Gestion des clients	6
4.3	Gestion des locations	7
5.	Le projet pour l'UE NFE102	8
5.1	Contexte	8
5.2	Environnement de développement	8
5.3	Démarrage du projet	8
5.4	Organisation du développement	9
5.4.1	Développement	9

1. Synopsis

Description : développer une application Web pour la gestion d'un vidéoclub

Dans le périmètre : gestion des films et de leurs supports/média, gestion des clients, location d'un film

En dehors du périmètre : les aspects paiement/abonnement, média abimé, vente de films d'occasion, délai de location, etc.

Livrables : le code complet, un export de la base de données et une documentation présentant votre travail, le reste à faire, les problèmes éventuels rencontrés et les solutions apportées. Ce document doit faire une 20aine de pages au maximum (merci pour moi 😊).

Prérequis : le développement doit être fait sous Zend Framework, en environnement Apache/MySQL (ou MariaDB)/PHP

Contraintes : le modèle de la base de données doit être le même pour tous → ne pas modifier la structure de la base de données. Toute modification non validée en cours sera éliminatoire.

2. Contexte

2.1 Le vidéoclub

La société FilmPlus est un vidéoclub proposant à ses clients un catalogue de films en DVD ou Blu-Ray, disponibles à la location uniquement à la boutique de Grenoble. Pour pouvoir louer un film, les clients doivent s'abonner. Ils disposent ainsi d'une carte du vidéoclub, avec un numéro unique.

2.2 Le besoin

L'objectif est de remplacer le logiciel existant, interne au vidéoclub, par une application Web, utilisable aussi bien par le personnel de la boutique, que par les clients. Cette application doit donc permettre de fournir différents services tels que :

- ajout de films, et des médias correspondants (DVD, Blu-Ray) ;
- gestion des clients : ajout, modification, désactivation, visu de l'historique de location ;
- gestion des locations :
 - o saisie d'une sortie
 - o liste des films sortis
 - o retour d'une location
- affichage des films et de leur statut (disponibles ou non) que ce soit pour le personnel ou bien pour les clients

3. Le projet

3.1 Objectifs





Comme indiqué, un outil Web de gestion de l'activité.

Ce logiciel Web doit être modulaire, évolutif et respecter les règles de base de la sécurité.

Il s'agira d'un logiciel Open Source.

3.2 Fonctionnalités

Chaque fonctionnalité sera détaillée dans les rubriques suivantes, mais voici déjà la liste de ce que nous souhaitons avoir – qui reprend le paragraphe précédent :

-  la gestion des films ;
-  gestion des clients ;
-  les locations ;
-  génération de rapports permettant de suivre l'activité.

L'application sera accessible à tous (personnel, clients, internaute). Dans un premier temps, et sauf avis contraire il n'y aura pas de gestion d'authentification. Ceci fera peut être l'objet d'évolutions suivant l'avancement dans le cours.

3.3 Prérequis et contraintes

Le développement devra être réalisé via le cadriciel Zend Framework (ZF). Ceci implique de développer sous un environnement Apache/MySQL/PHP. MariaDB peut aussi être utilisée, car totalement compatible avec MySQL.

Le code réalisé devra être modulaire, au sens ZF du terme. Autrement dit chaque module aura ses propres contrôleurs, modèles et vues.

Le code doit être documenté de manière équilibrée, afin de permettre une maintenance plus aisée.

Des bibliothèques externes peuvent être utilisées – jQuery, Zurb Foundation, Bootstrap, etc. – mais cela devrait être documenté dans le document joint en fin de projet.

4. Détail des fonctionnalités

4.1 Gestion des films

Il faut être capable a minima de :

- ➔ créer/modifier/désactiver (plus louable) les films;
- ➔ voir le statut d'un film : disponible, loué, désactivé.

Données pertinentes pour un film :

- Titre
- Date de production
- Réalisateur
- Liste d'acteurs
- Date de création dans la base
- Type de média

Le modèle de données fourni vous montre la liste exhaustive des champs nécessaires pour gérer un film.

Optionnel : bien que hors du cadre du cours, il peut être intéressant de récupérer les informations d'un film sur une base de données disponible sur Internet. Par exemple Allociné, qui fournit une API permettant de récupérer ces informations, ou <https://www.themoviedb.org/>.

4.2 Gestion des clients

Comme indiqué pour pouvoir louer un film il faut être avant tout client ce qui suppose la création d'un compte.

En théorie seul un membre du personnel doit pouvoir procéder à la gestion des comptes client, cependant, dans un souci de facilité, nous allons dans un premier temps ignorer cela et rendre disponible cette gestion à tous.

Un client va être caractérisé par les données suivantes a minima :

- Nom
- Prénom
- Adresse postale
- Téléphone
- Courriel
- Un numéro d'identification unique, visible sur sa carte du club

Il faudra ici pouvoir créer, modifier et désactiver un compte. Nous ne ferons pas de suppression afin de garder l'historique.





4.3 Gestion des locations

Le cœur de l'activité.

Il faut être capable de créer une nouvelle location, ce qui veut dire associer un film ou plus exactement un média à un client, sur une date donnée.

Nous ne gérons pas ici de date limite de location.

Une location sera donc caractérisée par :

-  une date d'emprunt ;
-  un film ou plus exactement un média ;
-  un client, actif dans la base de données ;
-  une date de retour, rendant le média de nouveau disponible.

A tout moment, un film visible sur l'application devra donner son statut, en l'occurrence location en cours (indisponible), ou disponible. Dans ce dernier cas, il doit être possible de louer le média indiqué.

Plusieurs façons de voir les locations en cours :

- sur la fiche d'un film, son statut avec un lien vers la fiche de location ;
- sur la fiche client, avoir la liste des locations en cours ;
- enfin sur la page de gestion des locations, voir toutes les locations en cours.

Pour la partie location et client, un historique de locations peut être intéressant.

A tout moment, la location peut s'arrêter, par le retour du média.

Optionnel : par défaut, il n'est pas demandé de gérer le retour des locations avec les relances. En option, si le reste est fait, développer une page listant les locations par durée décroissante (autrement dit, la location la plus vieille en premier). Mettre en avant celles ayant plus d'une semaine, avec possibilité d'envoyer un courriel au client.

5. Le projet pour l'UE NFE102

5.1 Contexte

Dans le cadre de l'Unité d'Enseignement NFE102 du CNAM, il est demandé la réalisation d'un projet. Pour cette année, nous allons travailler sur ce logiciel de gestion.

Dans ce cadre, l'accent est surtout mis sur le développement, plus que sur le « design » du site. Autrement dit, il faut se concentrer sur la partie fonctionnelle/métier, et la partie codage. Le « design » est un plus.

Certaines parties sont indispensables, d'autres seront vues comme des options.

5.2 Environnement de développement

Suivant la plateforme que vous utilisez, Windows, Linux ou Mac, vous pourrez installer WAMP/XAMPP/EasyPHP ou LAMP ou MAMP. Des liens sont donnés sur le site Plei@d.

Vous avez aussi besoin d'un environnement de développement qui peut se limiter à un éditeur de texte amélioré, Notepad++, PSPPad, ou bien un environnement complet type Netbeans (<http://netbeans.org/>) ou EclipsePHP (<http://www.eclipse.org/pdt/>). Le choix de l'outil est une affaire de goût, voire d'habitude.

Vérifiez si phpMyAdmin est installé avec l'ensemble Apache/MySQL/PHP, si vous ne l'avez pas il faudra l'installer pour la gestion de la base de données – qui reste basique dans ce cours.

Le code créé devra pouvoir s'exécuter tel quel sur une autre plateforme ce qui implique différentes contraintes :

- ➔ Pas d'URL absolue « en dur » dans le code, préférez les liens relatifs utilisation du « helper » ZF qui va bien
- ➔ Les identifiants d'accès à la BD, les chemins d'accès aux fichiers, et toutes les variables et constantes dont la valeur dépend de l'infrastructure, ne doivent pas être saisis « en dur » dans les fichiers php
- ➔ Attention à la casse des caractères dans les noms des fichiers : Windows ne fait pas de distinction entre User.php et user.php, mais les systèmes Unix oui

5.3 Démarrage du projet

La première étape est d'analyser le cahier des charges ci-dessus. Si vous avez des interrogations, ne pas hésiter à me poser des questions. Nous verrons ensemble comment combler les vides.

Il faudra ensuite procéder à la modélisation de l'application. UML est le choix préféré, mais vous pouvez utiliser autre chose, genre Merise.

Vu le délai et le travail à effectuer, ne passez pas tout votre temps dans la modélisation, ce n'est pas l'objectif de ce cours.

Le modèle de la base de données vous est fourni, afin que tout le monde travaille sur le même. **Une fois encore, ne pas modifier le modèle, sauf suite à validation en cours.**

Viendra ensuite le développement lui-même.

5.4 Organisation du développement

Le projet est fait de telle sorte que le développement peut être séquentiel. En l'occurrence, il est conseillé de commencer soit par la gestion des films, soit par celle des clients. Cette dernière étant la plus facile. Dans ce cas, voir pour mettre en place la partie, création, modification et désactivation. Ce qui vous permettra de « vous faire la main » sur ces aspects.

Le travail demandé est individuel, cependant, n'hésitez pas à collaborer via entre autres le forum disponible sur Plei@d.

5.4.1 Développement

Le développement sera fait par chaque élève suivant les conventions Zend Framework (voir <http://framework.zend.com/manual/fr/coding-standard.html>). Des tests devront être écrits afin de valider le bon fonctionnement du module.

Comme indiqué le code devra être documenté, sans excès.