

TP d'application Oeuvres

1 Sujet

Une médiathèque propose à des artistes d'exposer des œuvres dans ses locaux et propose à ses adhérents la possibilité de réserver ces œuvres.

Pour avoir une idée plus précise de l'application à développer, se connecter à l'URL suivante : <http://debian-epul.univ-lyon1.fr/oeuvres/public>

2 Fonctionnalités

Vous devez développer une application Web offrant les fonctionnalités suivantes :

Authentification :

Il faut se connecter en tant que propriétaire ce qui donne accès à toutes les opérations définies ci-après.

Ajouter une œuvre :

Saisie d'une œuvre et affectation d'un propriétaire

Consulter le catalogue :

Liste les œuvres et leur propriétaire, donne un accès à la réservation, la modification et la suppression de l'œuvre.

Réservation d'une œuvre :

Affichage de l'œuvre, saisie de la date et sélection de l'adhérent dans une liste déroulante.

Confirmation d'une réservation :

Liste les réservations et donne la possibilité de confirmer ou supprimer cette réservation.

Modification d'une œuvre :

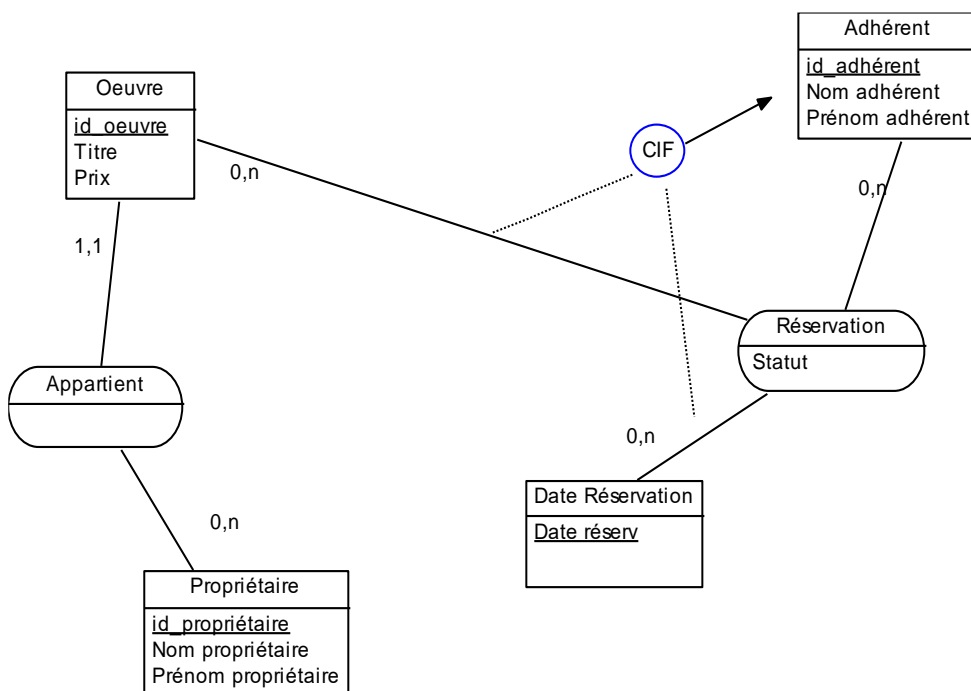
Modification du titre, du prix et du propriétaire.

Ajout d'une œuvre

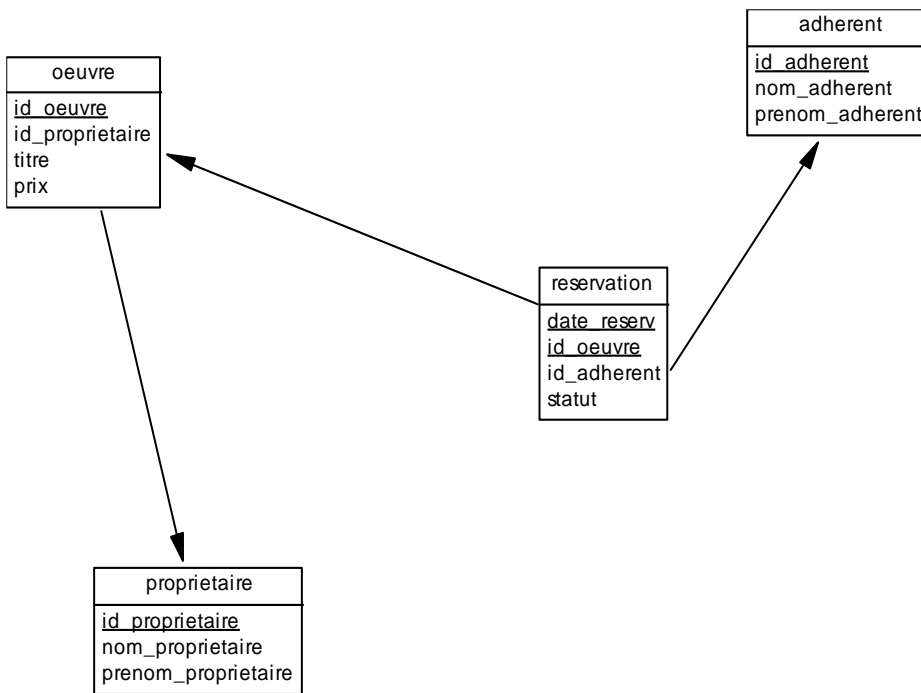
Saisie du titre, du prix et choix du propriétaire.

Déconnection

3 MCD

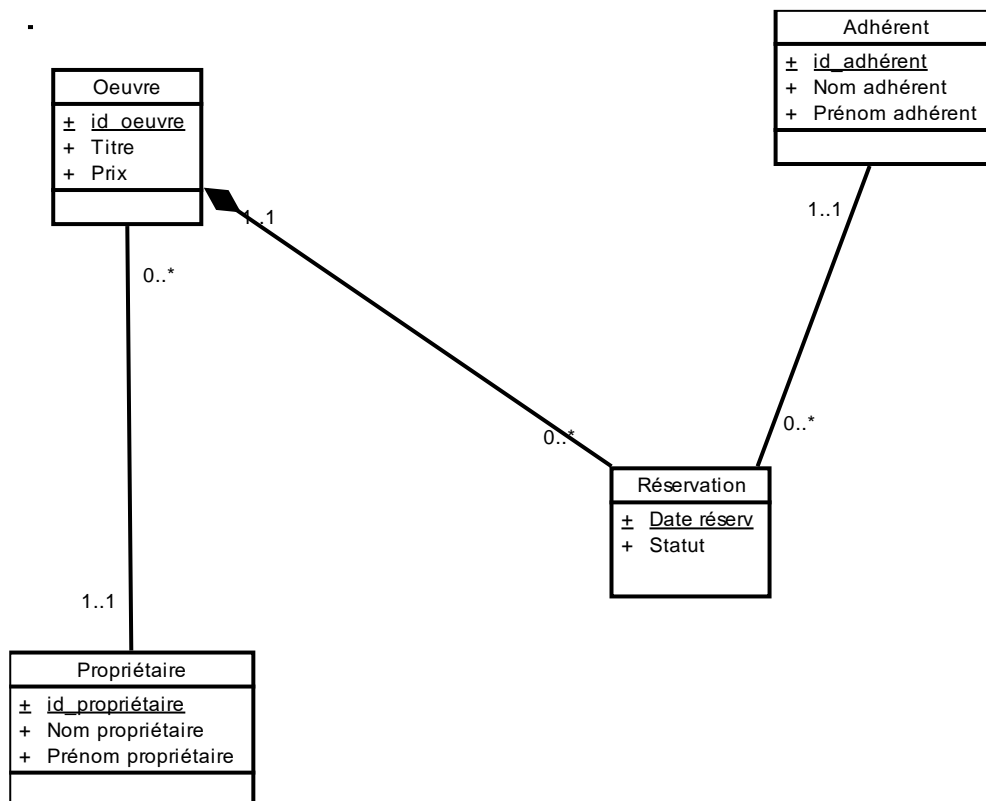


4 MLD



Note : statut indique si une réservation est en Attente (valeur affectée lors de la réservation d'une œuvre) ou Confirmée, valeur affectée lors de la confirmation de la réservation.

5 Modèle de classes



6 Environnement

Framework : Laravel

Serveur HTTP : Apache

Base de données : Mysql

7 Aide

7.1 Application

Vous disposez d'un squelette de l'application disponible sur [www.github.com](https://github.com/PolytechLyonInfo/TP_ISI2_Oeuvres.git) :

https://github.com/PolytechLyonInfo/TP_ISI2_Oeuvres.git

Attention : le squelette fournit les vues HTML qu'il faudra compléter (/* A compléter */) pour qu'elles soient réellement utilisables et c'est tout ! Tout le reste est à créer ex nihilo !

7.2 Base de données

Les scripts de création se trouvent dans le dossier database.

7.3 Restitution

Créer un repository nommé TP_Oeuvres_NomEquipe dans lequel vous déposerez votre application, la remise se faisant par envoi d'une invitation.

7.4 Validation

La reproduction fonctionnelle de l'application telle qu'elle est proposée en exemple vaut 12 points.

A ces points s'ajouteront les bonus suivants :

Quoi	Fonctionnalité	Bonus
Messages d'erreur	Plutôt que des messages "brut de décoffrage", afficher des messages compréhensibles par un utilisateur lambda. Il faut en plus que la structure employée soit généralisable, donc éviter les constantes dans le code. Une piste : baser le libellé du message affiché sur le nom de la contrainte mysql retournée.	4
Date réservation	Utiliser un composant de type Calendrier. Attention, il faut que cela fonctionne sur tout type de navigateur !	1
Date réservation	Afficher et saisir la date au format français, mais continuer à l'enregistrer au format anglais	2
Réservation	Quand on crée une réservation et qu'il y a une erreur, on perd les données. Faire en sorte que les données saisies restent affichées.	2
Saisie nouvelle œuvre	Quand on crée une nouvelle œuvre et qu'il y a une erreur, on perd les données. Faire en sorte que les données saisies restent affichées.	2
Confirmation de réservation	Une réservation ne devrait pouvoir être confirmée que si elle est en Attente.	2
Confirmation de réservation	Comme pour une suppression, demander une confirmation avant d'effectuer la bascule.	1
Liste des réservations	Quand il n'y a pas de réservations, seule l'entête du tableau est affichée, il vaudrait mieux afficher le texte : Aucune réservation enregistrée, ou quelque chose de similaire.	2

Il faudra indiquer dans le fichier README du repository quelles fonctionnalités complémentaires vous avez choisi de développer.