

## MEDIATEK DOCUMENTS

### CONTEXTE

MediaTek86 est un réseau qui gère les médiathèques de la Vienne, et qui a pour rôle de fédérer les prêts de livres, DVD et CD et de développer la médiathèque numérique pour l'ensemble des médiathèques du département. Afin de gérer le catalogue et les commandes de documents, MediaTek86 a confié un projet de développement d'une application de gestion à la société InfoTechServices 86.

### MISSIONS

Le but de cet atelier est de faire évoluer une application de bureau (C#) exploitant une API REST (PHP) pour l'accès à une base de données relationnelle MySQL et qui permet de gérer les documents et les commandes des médiathèques de la chaîne MediaTek86.

Un premier développeur s'est occupé de la construction de la base de données et du développement de certaines fonctionnalités de l'application. Il s'agit d'une application de bureau, prévue d'être installée sur plusieurs postes dans la médiathèque et accédant à la même base de données.

Le but est d'y ajouter de nouvelles fonctionnalités pour réaliser la gestion du catalogue des médiathèques et des commandes de documents.

**PRECISION** : Le dossier documentaire contient 7 missions. Dans le cadre de ma formation, il était conseiller de réaliser le projet avec un autre étudiant. Dans le cas contraire, la réalisation des missions 1 et 3 étaient optionnelles. N'ayant trouver aucun étudiant avec qui le faire, je n'ai donc pas réalisé ces missions

### Langages et technologies + outils utilisés :

Serveur : WAMP (apache, MySQL, PHP)

Langages : C#, PHP, SQL

Environnement de développement : Visual Studio 2019 (Application) et NetBeans (API)

Hébergement API : Hostinger

Versionning : GitHub

Tests : Postman (+Specflow)

Documentation : phpDocumentor + SandCastle

## **Mission 0 : Préparer l'environnement de travail**

Après avoir récupéré les documents nécessaires à la réalisation (cahier des charges + dossier documentaire contenant les missions à réaliser) et le schéma conceptuel des données j'ai installé en local les logiciels nécessaires pour le fonctionnement de l'API : WAMP, NetBeans et Postman. Ensuite j'ai récupéré le code source de l'API à faire évoluer ainsi que le script de la base de données. Le code source est dans mon « www » sur WAMP et le script est exécuté dans phpMyAdmin.

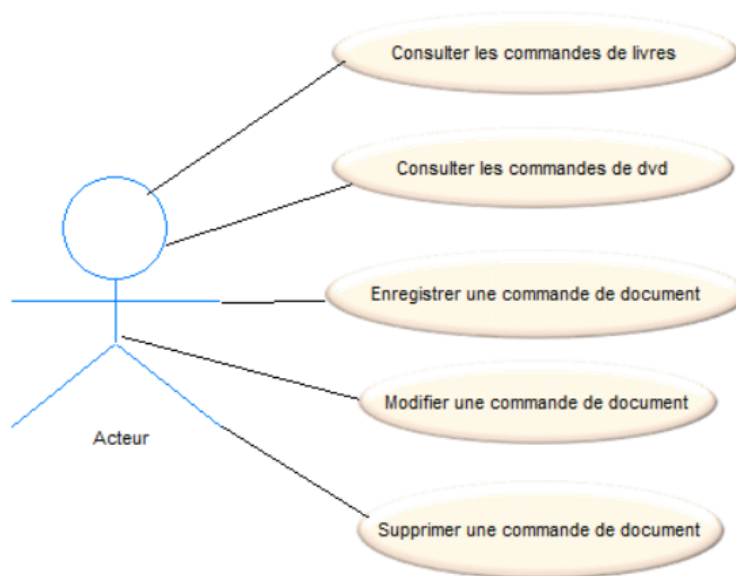
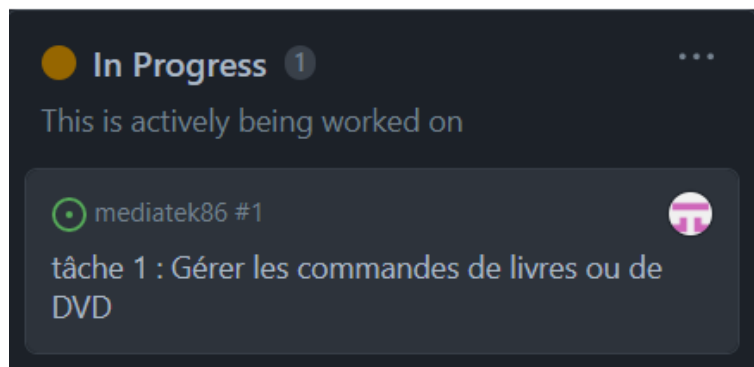
Ensuite j'ai récupéré l'application C# sur le dépôt distant et ajouté le package Nuget « Microsoft.aspnet.webapi.client » et l'application était désormais reliée à l'API et était prête à être utilisée.

## **Mission 2 : Gérer les commandes**

### **Tâche 1 : Gérer les commandes de livres et de DVD**

- Dans la base de données, créer la table 'Suivi' qui contient les différentes étapes de suivi d'une commande de document de type livre ou dvd. Relier cette table à CommandeDocument.
- Créer un onglet (ou une nouvelle fenêtre) pour gérer les commandes de livres.
- La charte graphique doit correspondre à l'existant.
- Dans toutes les listes, permettre le tri sur les colonnes.
- Dans l'onglet (ou la fenêtre), permettre la sélection d'un livre par son numéro, afficher les informations du livre ainsi que la liste des commandes, triée par date (ordre inverse de la chronologie). La liste doit comporter les informations suivantes : date de la commande, montant, nombre d'exemplaires commandés et l'étape de suivi de la commande.
- Créer un groupbox qui permet de saisir les informations d'une nouvelle commande et de l'enregistrer. Lors de l'enregistrement de la commande, l'étape de suivi doit être mise à "en cours".
- Permettre de modifier l'étape de suivi d'une commande en respectant certaines règles (une commande livrée ou réglée ne peut pas revenir à une étape précédente (en cours ou relancée), une commande ne peut pas être réglée si elle n'est pas livrée).
- Permettre de supprimer une commande uniquement si elle n'est pas encore livrée. Faire un trigger qui réalise aussi la suppression dans la classe fille.
- Créer le trigger qui se déclenche si une commande passe à l'étape "livrée" et qui crée autant de tuples dans la table "Exemplaire" que nécessaires, en valorisant la date d'achat avec la date de commande et en mettant l'état de l'exemplaire à "neuf". Le numéro d'exemplaire doit être séquentiel par rapport au livre concerné.
- Créer un onglet pour gérer les commandes de DVD en suivant la même logique que pour les commandes de livres.
- Toutes les sécurités seront mises en place pour éviter des erreurs de manipulation.
- Créer le trigger qui contrôle la contrainte de partition de l'héritage sur Commande.

**Temps estimé : 8h / temps réel : 13h**



J'ai d'abord commencé par la création du nouvel onglet pour la commande de livres en me basant sur la page des parutions de revues. De plus, il fallait rajouter une datagrid qui affichera la liste des commandes concernant le livre recherché avec le numéro de document.


Deux groupbox viennent s'ajouter : un pour l'ajout ou la suppression d'une commande sur le livre sélectionnée et l'autre pour la modification de l'état de suivi.

La partie recherche de livres fonctionne de la même manière que l'onglet « livre » avec simplement le nom des différents boutons et champs de texte qu'il fallait modifier.

Gestion des documents de la médiathèque

Livres | DVD | Revues | Parutions des revues | Commandes de livres | Commandes de DVD | Commandes de revues

Recherche de livre

Numéro de document : 00017 Recherche Code ISBN :  Image : 

Titre : Catastrophes au Brésil  
 Auteur : Philippe Masson  
 Collection :   
 Genre : Policier  
 Public : Ados  
 Rayon : Jeunesse romans  
 Chemin de l'image :

Commandes :

	Nombre d'exemplaires	Montant	Date de commande	Suivi
▶	4	20	18/04/2023	en cours
	1	5	18/04/2023	en cours
	1	5	18/04/2023	en cours

Informations de commande














Numéro de commande : 02000 Ajouter une commande  
 Nombre d'exemplaires : 4 Supprimer une commande  
 Montant : 20  
 Date de la commande : 18/04/2023 ▼

Suivi de commande

Etape de suivi : en cours  
  
Modifier le suivi

Il fallait aussi ajouter une nouvelle table « suivi » dans la base de données qui contient l'id du libelle et le libelle ( en cours, livrée, réglée, relancée) , pour faire le lien avec la table commandedocument ( car une commande de document doit contenir le libelle correspondant à sa « situation ».

J'ai ajouter dans commandedocument une clé étrangère « idsuivi » qui aura donc l'id du libelle et que nous pourrons récupérer afin d'obtenir son état.

				id	libelle
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	00001	en cours
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	00002	livrée
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	00003	réglée
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	00004	relancée

#### Mission 4 : Mettre en place des authentifications

- Dans la base de données, ajouter une table Utilisateur et une table Service, sachant que chaque utilisateur ne fait partie que d'un service. Pour réaliser les tests, remplir les tables d'exemples.
- Ajouter une première fenêtre d'authentification. Faire en sorte que l'application démarre sur cette fenêtre.
- Suivant le type de personne authentifiée, empêcher certains accès en rendant invisibles ou inactifs certains onglets ou objets graphiques.
- Dans le cas du service Culture qui n'a accès à rien, afficher un message précisant que les droits ne sont pas suffisants pour accéder à cette application, puis fermer l'application.
- Faire en sorte que l'alerte de fin d'abonnement n'apparaisse que pour les personnes concernées (qui gèrent les commandes).

**Temps estimé : 4h / temps réel : 5h**