

BTS SIO

Bilan atelier de professionnalisation n°2

Mediatekdocuments

Table des matières

Mission 1: Gérer les commandes.....	4
Tâche 1: gérer les commandes de livres ou de DVD.....	4
Tâche 2: gérer les commandes de revues.....	6
Mission 2: mettre en place des authentifications.....	9
Mission 3: assurer la sécurité, la qualité et intégrer des logs.....	11
Tâche 1: corriger des problèmes de sécurité.....	11
Tâche 2: intégrer des logs.....	13
Mission 4: tester et documenter.....	14
Tâche 1: gérer les tests.....	14
Tâche 2: créer les documentations techniques.....	16
Tâche 3: créer la documentation utilisateur en vidéo.....	17
Mission 5: déployer et gérer les sauvegardes des données.....	18
Tâche 1: déployer le projet.....	18

Introduction

Contexte:

«Cette application permet de gérer les documents (livres, DVD, revues) d'une médiathèque. Elle a été codée en C# sous Visual Studio 2019. C'est une application de bureau, prévue d'être installée sur plusieurs postes accédant à la même base de données.

L'application exploite une API REST pour accéder à la BDD MySQL. Des explications sont données plus loin, ainsi que le lien de récupération.»

Existant:

«Actuellement l'application est partiellement codée. Voici les fonctionnalités actuellement opérationnelles : recherches et affichage d'informations sur les documents de la médiathèque (livres, DVD, revues), réception de nouveaux numéros de revues. L'application ne comporte qu'une seule fenêtre divisée en plusieurs onglets.»

Langage et technologies utilisées:

L'application de bureau est codée en C#, l'API est codée en PHP, et le SGBD est MySQL.

Mission 1: Gérer les commandes

Tâche 1: gérer les commandes de livres ou de DVD

Tâche: Ajouter la possibilité de commander un ou plusieurs exemplaires d'un livre ou d'un DVD et de suivre l'évolution d'une commande. L'application doit permettre de voir, pour les livres ou les DVD, la liste des commandes et gérer le suivi. Lorsqu'une commande de livre ou de DVD est «livrée», il faut que les exemplaires concernés soient automatiquement générés dans la BDD, avec un numéro séquentiel par rapport au document concerné. Une commande doit pouvoir être supprimée si elle n'est pas encore livrée.

Temps estimé: 8h

Temps réel: 12 h

Kanban:

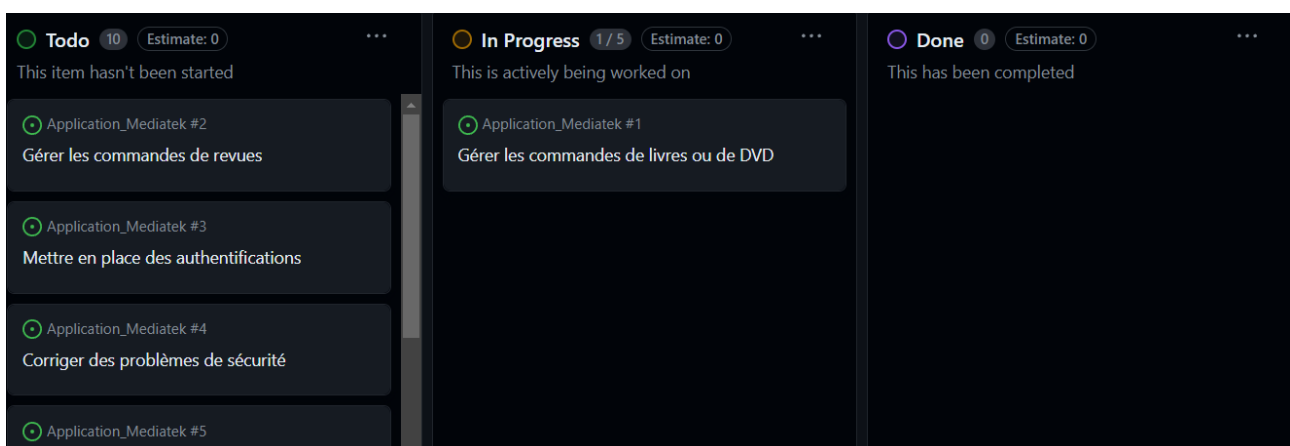
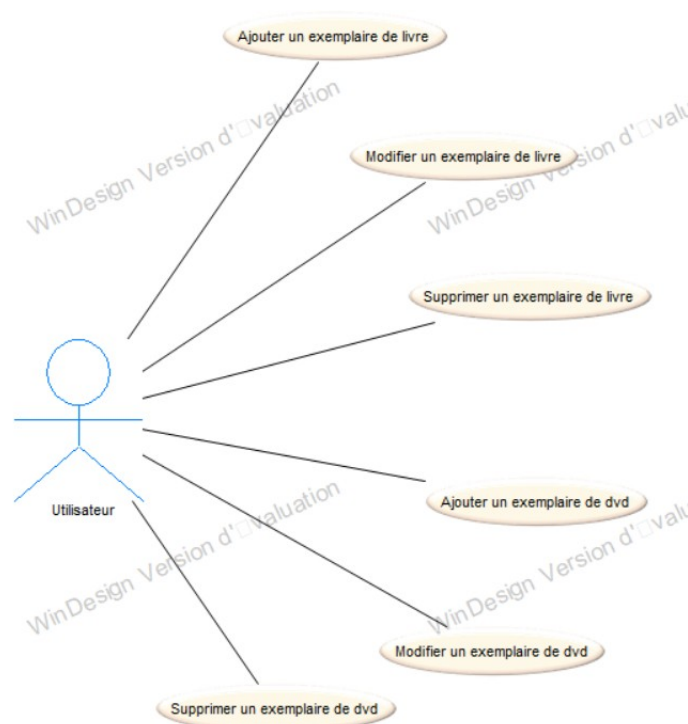


Diagramme de cas d'utilisation:



Onglet commande de livres:

Livres
DVD
Revue
Parutions des revues
Commandes de livres
Commandes de Dvd

Recherche livre

Saisir un numéro de document :

Id	Titre	Auteur	Collection	Genre	Public	Rayon
00017	Catastrophes au Brésil	Philippe Masson		Policier	Ados	Jeunesse romans
00007	Dans les coulisses du musée	Kate Atkinson		Roman	Tous publics	Littérature étrangère
00003	Et je danse aussi	Anne-Laure Bondoux		Comédie	Tous publics	Littérature française
00019	Guide Vert - Iles Canaries		Guide Vert	Voyages	Tous publics	Voyages
00020	Guide Vert - Irlande		Guide Vert	Voyages	Tous publics	Voyages
00008	Histoire du juif errant	Jean d'Omesson		Roman	Adultes	Littérature française
00025	L'archipel du danger	Ayrolles - Masbou	De cape et de crocs	Bande dessinée	Adultes	BD Adultes
00004	L'armée furieuse	Fred Vargas	Commissaire Adamsberg	Policier	Adultes	Policiers français étrangers

Et je danse aussi

Numéro de commande:
Date de commande:
Montant:
Nombre d'exemplaires:
Numéro du livre:
Etat de la commande:

	Id	DateCommar	NbExemplaire	Etat	Montant
▶	20007	22/04/2024	1	en cours	4

J'ai créé cet onglet en ajoutant une liste affichant les livres, il est possible de rechercher un livre par son numéro à l'aide du bouton rechercher. J'ai ajouté, suite au click sur un livre de la liste, l'affichage d'une seconde liste affichant les exemplaires de ce livre. Des zones de texte sont disponibles pour modifier ou ajouter des exemplaires. Le bouton «ajouter» créé un nouvel enregistrement en base de donnée, le bouton «modifier» également. Le bouton «supprimer» supprime l'exemplaire en question seulement si l'état de la commande est «en cours».

Onglet commande de dvd:

Livres
DVD
Revue
Parutions des revues
Commandes de livres
Commandes de Dvd

Recherche DVD

Saisir un numéro de document :

Id	Titre	Duree	Realisateur	Genre	Public	Rayon
20003	Jurassic Park	128	Steven Spielberg	Science Fiction	Tous publics	DVD films
20002	Le seigneur des anneaux : la communauté de l'anneau	228	Peter Jackson	Fantasy	Tous publics	DVD films
20004	Matrix	136	Les Wachowski	Science Fiction	Tous publics	DVD films
20001	Star Wars 5 L'empire contre attaque	124	George Lucas	Science Fiction	Tous publics	DVD films

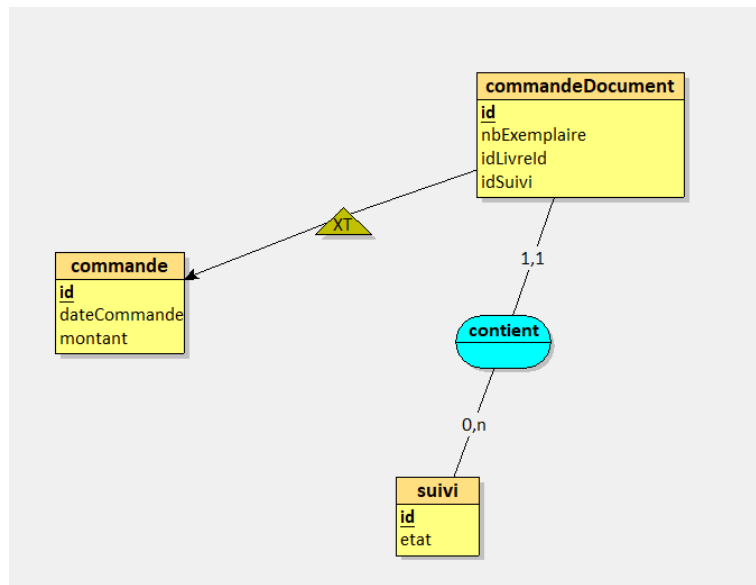
Jurassic Park

Numéro de commande:
Date de commande:
Montant:
Nombre d'exemplaires:
Numéro du dvd:
Etat de la commande:

	Id	DateCommar	NbExemplaire	Etat	Montant
▶	20002	18/04/2024	2	en cours	23
	20004	18/04/2024	2	en cours	100
	20001	19/03/2024	1	en cours	5

Cet onglet reprends les même fonctionnalités que l'onglet des livres avec cette fois ci la gestion des exemplaires de dvd.

Modèle conceptuel de données:



J'ai ajouté trois tables à ma base de donnée, la table commandeDocument héritée de la table commande. Et la table suivi liée à chaque comande de document pour connaître son état: en cours, livrée...

Pour ce qui est du code de l'api, j'ai ajouté dans le fichier Access.BDD, la méthode «public function deleteCommande(\$champs)», ainsi que «public function insertCommande(\$champs)» et «public function updateCommande(\$id, \$champs)».

Tâche 2: gérer les commandes de revues

Tâche: Ajouter la possibilité de commander une revue. Une commande de revue revient à réaliser un abonnement (ou un renouvellement d'abonnement, ce qui revient au même).

L'application doit permettre de voir la liste des commandes, même pour les abonnements expirés.

Une commande ne peut être supprimée que si aucun exemplaire (parution) lié à cette commande n'est enregistré.

Au démarrage de l'application, une petite fenêtre doit s'ouvrir automatiquement pour afficher les revues dont l'abonnement se termine dans moins de 30 jours.

Temps estimé: 4 h

Temps réel: 4h

Kanban:

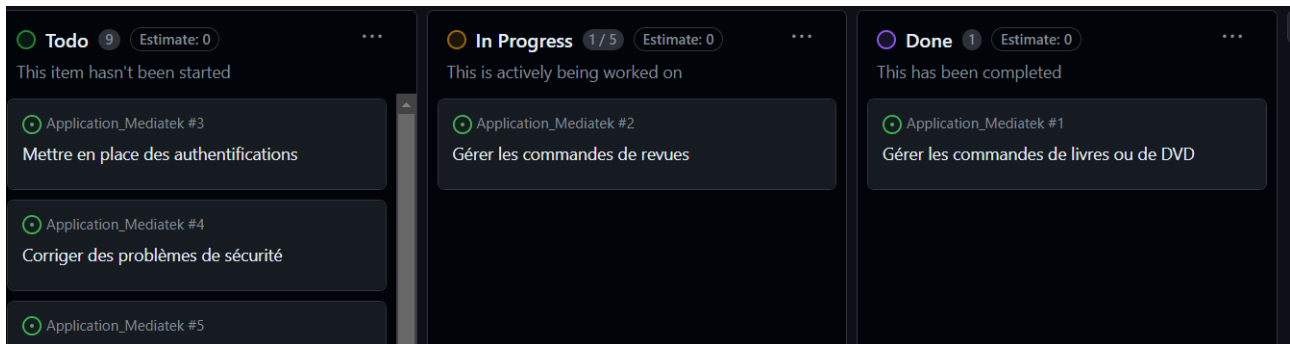
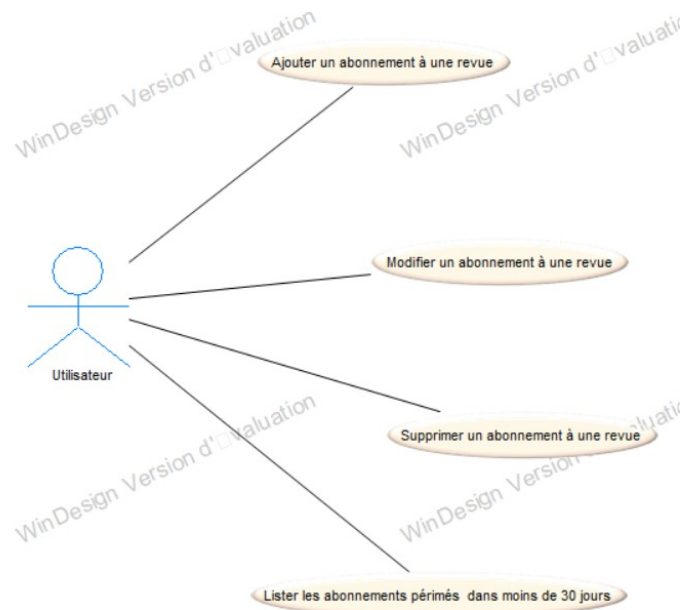


Diagramme de cas d'utilisation:



Onglet abonnement :

J'ai créé cet onglet en ajoutant une liste affichant les revues, il est possible de rechercher une revue par son numéro à l'aide du bouton rechercher. J'ai ajouté, suite au click sur une revue de la liste, l'affichage d'une seconde liste affichant les exemplaires de cette revue (abonnements). Des zones de texte sont disponibles pour modifier ou ajouter des abonnements. Le bouton «ajouter» crée un nouvel enregistrement en base de donnée, le bouton «modifier» également. Le bouton «supprimer» supprime l'abonnement en question. J'ai ajouté un bouton «Filtrer» permettant d'afficher dans la liste des revues seulement les revues dont les dates de fin d'abonnements sont inférieures à 30 jours.

LivresDVDRevueParutions des revuesCommandes de livresCommandes de DvdAbonnement

Recherches

Saisir le titre ou la partie d'un titre :Revue dont l'abonnement se termine dans moins de 30 jours :

Saisir un numéro de document :

Rechercher

Filtrer

Id	Titre	Periodicite	DelaiMiseADispo	Genre	Public	Rayon
10002	Alternatives Economiques	MS	52	Presse Economique	Adultes	Magazines
10001	Arts Magazine	MS	52	Presse Culturelle	Adultes	Magazines
10003	Challenges	HB	15	Presse Economique	Adultes	Magazines
10011	Geo	MS	52	Presse Culturelle	Tous publics	Magazines
10009	L'Equipe	QT	5	Presse sportive	Adultes	Presse quotidienne
10010	L'Equipe Magazine	HB	12	Presse sportive	Adultes	Magazines
10008	L'Obs	HB	26	Actualités	Adultes	Magazines
10006	Le Monde	QT	5	Actualités	Adultes	Presse quotidienne

Arts Magazine

Numéro de commande : 20009

Date de commande : vendredi 26 avril 2024

Montant : 50

Fin d'abonnement : mercredi 26 juin 2024

Numéro de revue : 10001

Ajouter

Modifier

Supprimer

Annuler

Valider

	Id	DateCommande	DateFinAbonnen	Montant
►	20009	26/04/2024	26/06/2024	50
	10001	19/03/2024	26/04/2024	5

Mission 2: mettre en place des authentifications

Tâche: Ajout d'une fenêtre d'authentification avant d'accéder à l'application, suivant les droits relatifs aux comptes. Restreindre l'accès à certaines fonctionnalités pour certains comptes

Temps estimé: 4 h

Temps réel: 4h

Kanban:

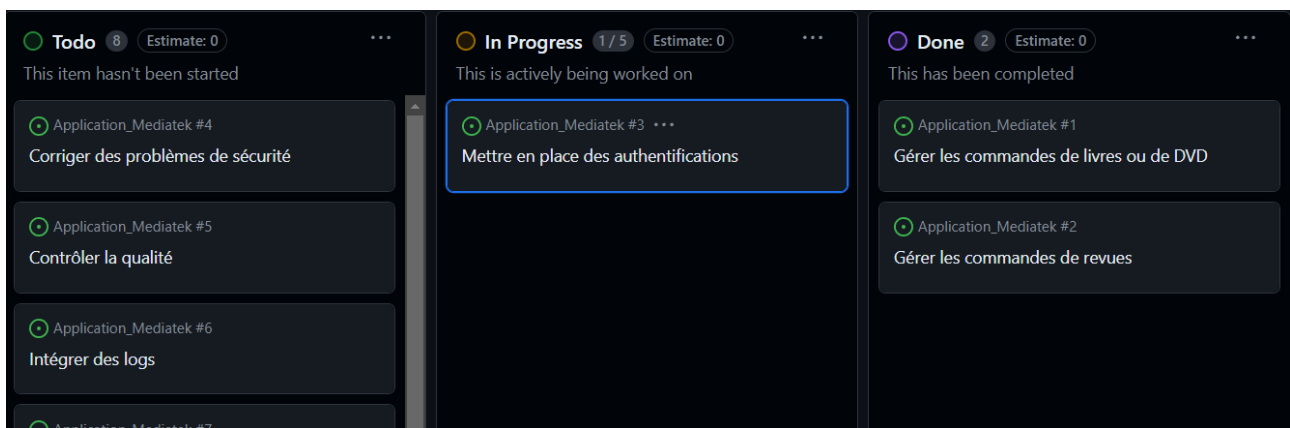
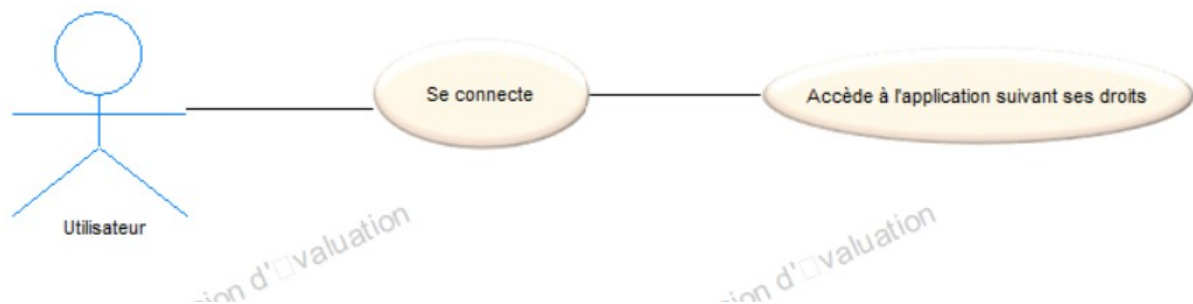
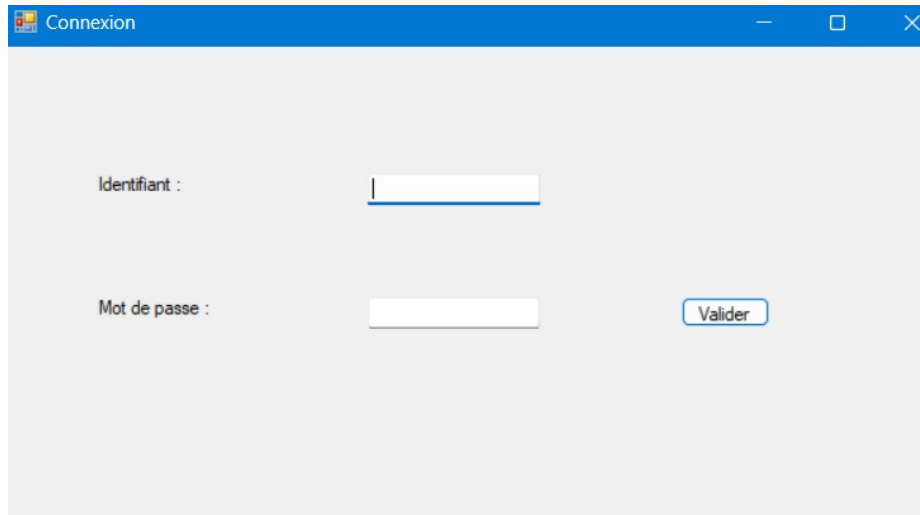


Diagramme de cas d'utilisation:



Fenêtre de démarrage:



The image shows a screenshot of a web application window titled "Connexion". The window has a blue header bar with the title and standard window controls (minimize, maximize, close). The main content area is light gray and contains two input fields. The first field is labeled "Identifiant :" and the second is labeled "Mot de passe :". To the right of the password field is a blue button labeled "Valider".

J'ai ajouté une zone de texte pour l'identifiant et une pour le mot de passe, le bouton Valider permet d'ouvrir l'application si les identifiants son correctes.

J'ai ajouté dans la base de donnée, une table Utilisateur ainsi qu'une table Service, la première contient des informations tels que le mail, le nom d'utilisateur. La deuxième contient les services rattachés à certains droits (administratif tous les droits, service de prêts seulement la modification).

Mission 3: assurer la sécurité, la qualité et intégrer des logs

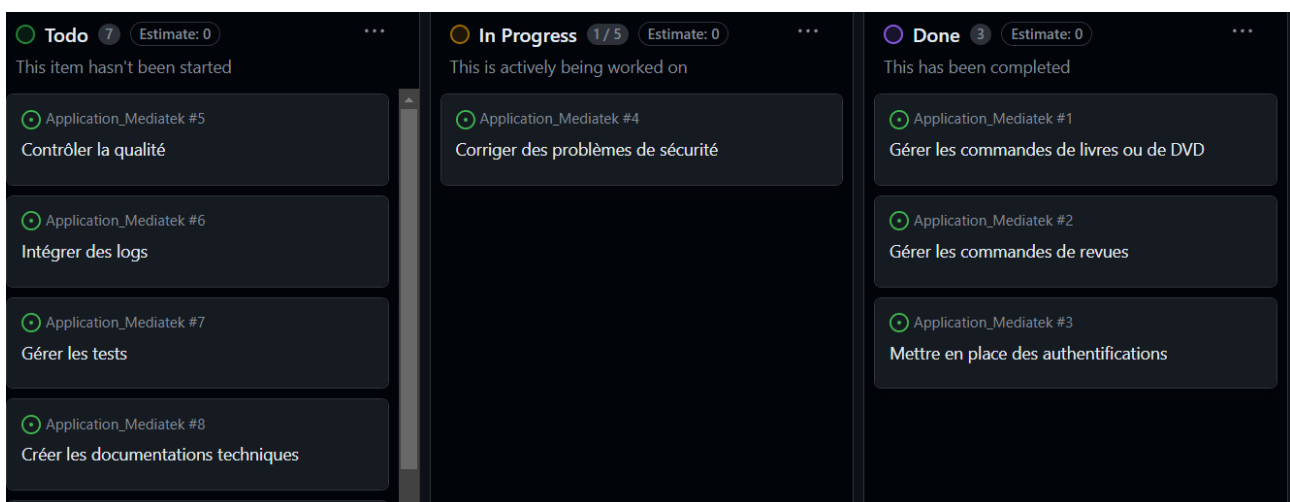
Tâche 1: corriger des problèmes de sécurité

Tâche: sécuriser l'authentification qui est actuellement en dur; permettre de ne pas lister les fichiers contenus dans le dossier de l'API par l'url.

Temps estimé: 3 h

Temps réel: 2 h

Kanban:



Authentification

J'ai remplacé la ligne d'authentification en dur: `authenticationString = "admin:adminpwd";`

par:

```
String authenticationString = GetAuthenticationString(authenticationName);
```

```
static string GetAuthenticationString(string name)
{
    string returnValue = null;
    ConnectionStringSettings settings =
    ConfigurationManager.ConnectionStrings[name];
    if (settings != null)
```

```

        returnValue = settings.ConnectionString;
    return returnValue;
}

```

URL

La page lorsque l'on accède à l'url avant modification:



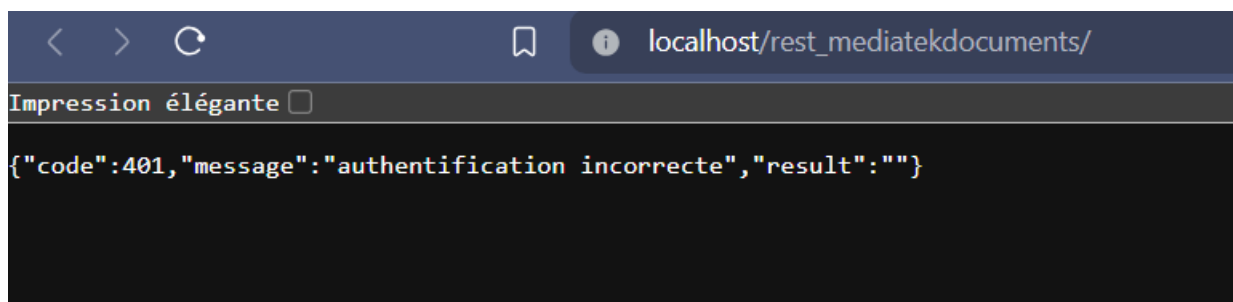
The screenshot shows a web browser window with the address bar displaying `localhost/rest_mediatekdocuments/`. The page title is "Index of /rest_mediatekdocuments". Below the title is a table listing the contents of the directory.

Name	Last modified	Size	Description
 Parent Directory		-	
 AccessBDD.php	2024-04-28 08:08	15K	
 ConnexionPDO.php	2024-03-12 06:27	2.0K	
 Controle.php	2024-04-25 09:41	3.9K	
 Mission2.sql	2024-04-22 09:18	1.3K	
 htaccess.save	2024-03-12 06:27	304	
 mediatek86.sql	2024-03-12 06:27	15K	
 mediatekdocuments.php	2024-03-14 21:47	1.6K	
 nbproject/	2024-03-13 08:08	-	

At the bottom of the page, it says: *Apache/2.4.54 (Win64) PHP/8.1.13 mod_fcgid/2.3.10-dev Server at localhost Port 80*

Ajout de la ligne «`RewriteRule ^$ mediatekdocuments.php?error=404`» dans le fichier [htaccess](#)

La page lorsque l'on accède à l'url après modification:



The screenshot shows a web browser window with the address bar displaying `localhost/rest_mediatekdocuments/`. The page content is a dark-themed error message.

Impression élégante ☐

```

{"code":401,"message":"authentification incorrecte","result":""}

```

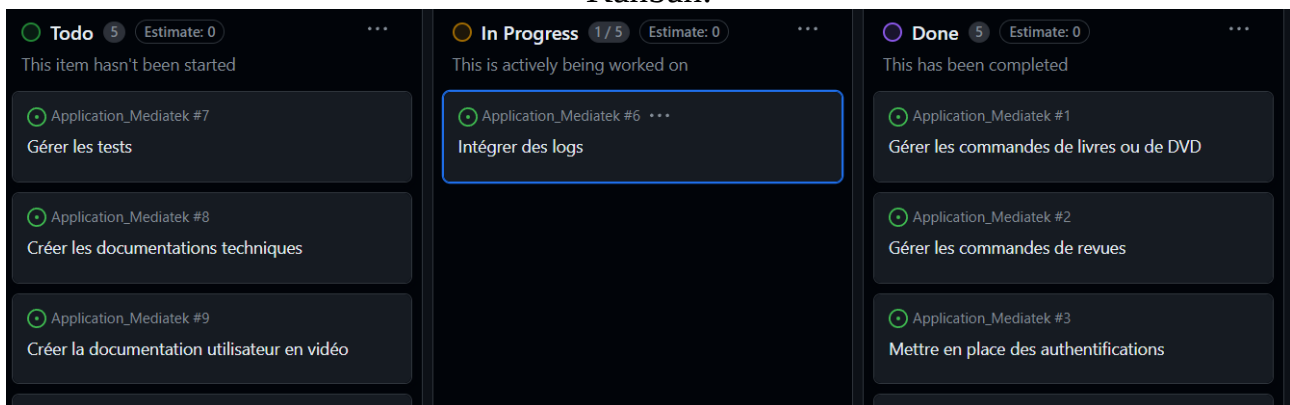
Tâche 2: intégrer des logs

Tâche: Dans la classe Access, ajouter le code de configuration des logs et des logs au niveau de chaque affichage console (à enregistrer dans un fichier de logs).

Temps estimé: 2 h

Temps réel: 1h

Kanban:

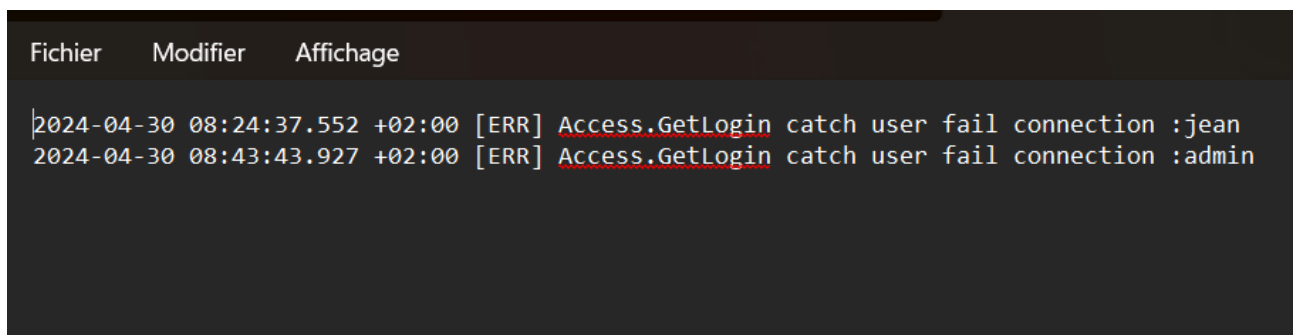


Code ajouté : `private Access()`

```
{
    try
    {
        Log.Logger = new LoggerConfiguration()
            .MinimumLevel.Verbose()
            .WriteTo.File(@"C:\logs\log.txt")
            .CreateLogger();
    }
}
```

Ajout du package Serilog

Fichier de log:



Mission 4: tester et documenter

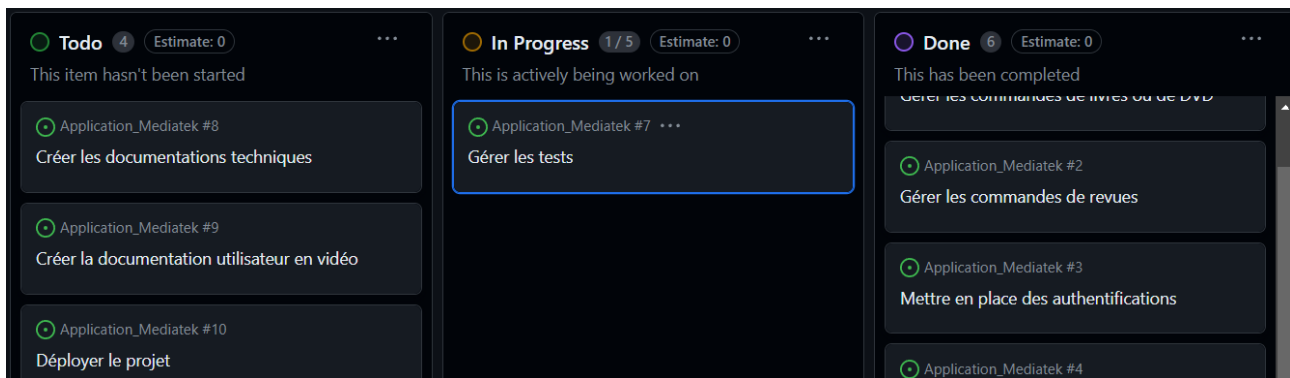
Tâche 1: gérer les tests

*Tâche: - Écrire les tests unitaires sur les classes du package Model
-Écrire les tests fonctionnels sur les recherches dans l'onglet des livres*

Temps estimé: 5h

Temps réel: 4 h

Kanban:



Exemple de code de test unitaire :

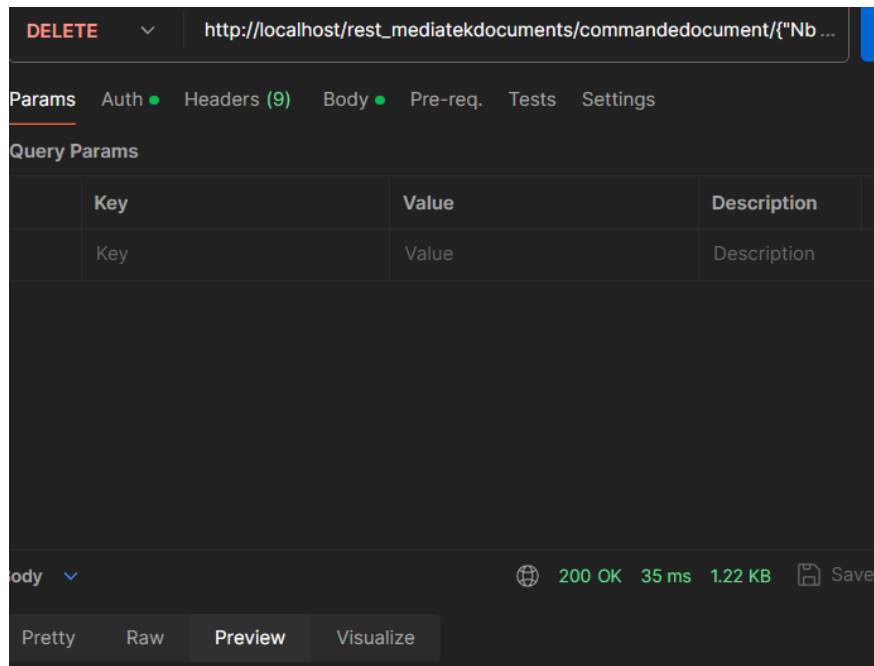
```
public class CommandeTests
{
    private const string id = "0005";
    private static readonly DateTime dateCommande = DateTime.Now;
    private const double montant = 22.4;
    private static readonly Commande commande = new Commande(id, dateCommande, montant);

    [TestMethod()]
    public void CommandeTest()
    {
        Assert.AreEqual(id, commande.Id, "devrait réussir : id valorisé");
        Assert.AreEqual(dateCommande, commande.DateCommande, "devrait réussir : date de commande valorisée");
        Assert.AreEqual(montant, commande.Montant, "devrait réussir : montant valorisé");
    }
}
```

Résultat :

Explorateur de tests			
Test	Durée	Caractéris...	Message d'erreur
TestMediatekDocuments (5)	15 ms		
TestMediatekDocuments.mediatek...	9 ms		
CommandeTests (1)	9 ms		
CommandeTest	9 ms		

Exemple de test fonctionnel avec Postman:



Champs : 1

```
C:\wamp64\www\rest_mediatekdocuments\Controle.php:68:
array (size=7)
  'NbExemplaire' => int 2
  'IdLivreDvd' => string '00001' (length=5)
  'IdSuivi' => int 1
  'Etat' => string 'en cours' (length=8)
  'Id' => string '20012' (length=5)
  'DateCommande' => string '2024-05-01' (length=10)
  'Montant' => float 30
{"code":200,"message":"OK","result":""}
```

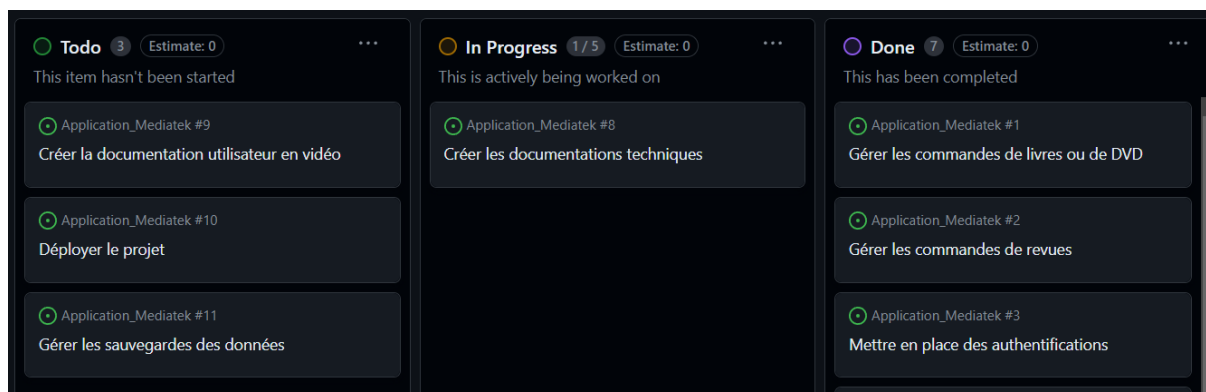
Tâche 2: créer les documentations techniques

Tâche: - Contrôler que les commentaires normalisés sont bien tous ajoutés et corrects.
- Générer les documentations techniques des 2 applications :

Temps estimé: 1h

Temps réel: 2h

Kanban:



Documentation de l'application C#

The screenshot shows the Visual Studio documentation page for the `Access` class. The browser address bar indicates the file path: `D:/Users/theom/Desktop/atelier2/mediatekdocuments/MediaTekDocuments/bin/Debug/logs/Help/html/d5b248fe-b3ca-0e5b-9aaa-187afae5d871.htm`.

Access Class

Classe d'accès aux données

Definition

Namespace: [MediaTekDocuments.dal](#)
Assembly: MediaTekDocuments (in MediaTekDocuments.exe) Version: 1.0.0.0 (1.0.0.0)

```
C#  
public class Access
```

Inheritance: [Object](#) → [Access](#)

Methods

CreerEntite	Créer une entite dans la BDD, return true si l'opération s'est correctement déroulée
CreerEntite2	
CreerExemplaire	écriture d'un exemplaire en base de données
Equals	(Inherited from Object)
Finalize	(Inherited from Object)

Documentation de l'API

The screenshot shows the API documentation website for `rest_mEDIATEKDOCUMENTS`. The browser address bar indicates the URL: `C:/wamp64/www/rest_mEDIATEKDOCUMENTS_doc/index.html`.

rest_mEDIATEKDOCUMENTS

Packages

- [Application](#)

Reports

- [Deprecated](#)
- [Errors](#)
- [Markers](#)

Indices

- [Files](#)

Documentation

Table of Contents

Packages

- [Application](#)

Classes

- [ConnexionPDO](#)
Classe de connexion et d'exécution des requêtes dans une BDD MySQL
- [Controle](#)
Contrôleur : reçoit et traite les demandes du point d'entrée

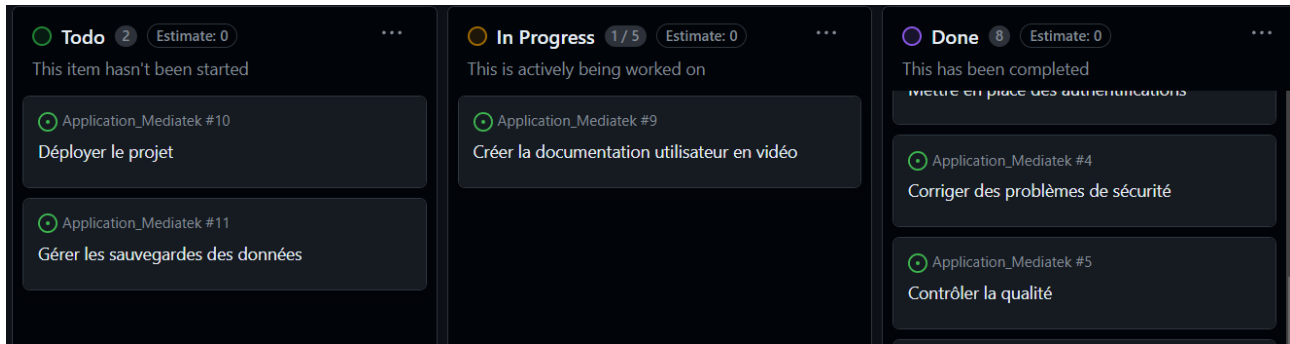
Tâche 3: créer la documentation utilisateur en vidéo

Tâche: -Créer une vidéo de 10mn maximum qui présente l'ensemble des fonctionnalités de l'application C#.

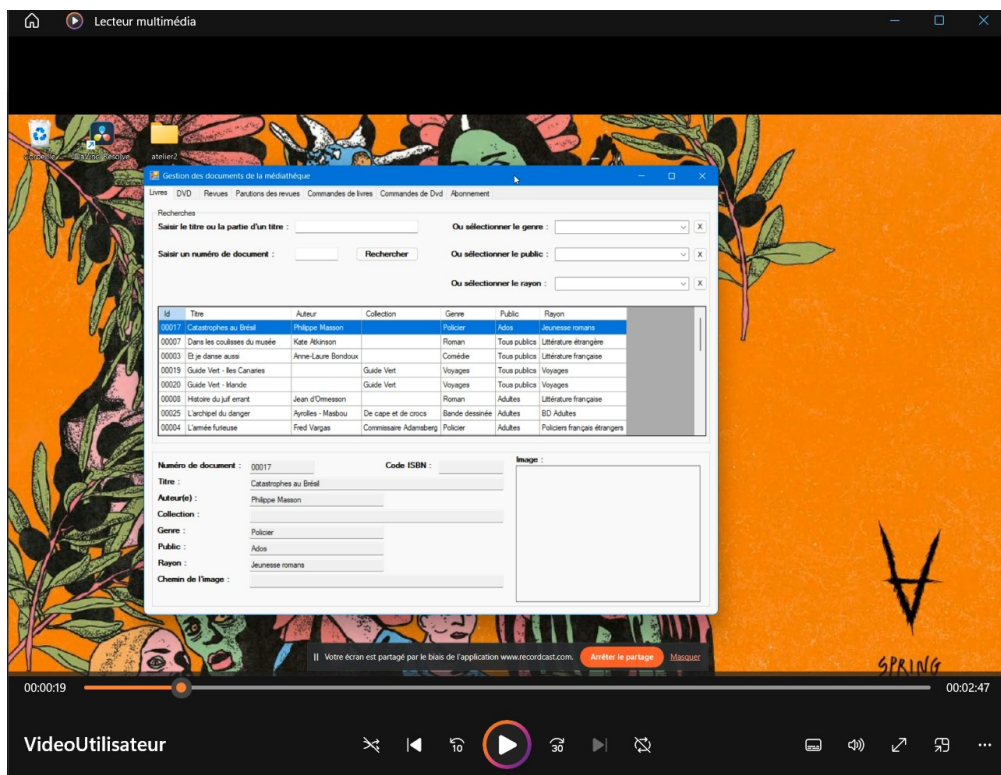
Temps estimé: 2 h

Temps réel: 1 h

Kanban:



Capture de la vidéo:



Temps de la vidéo: 3 mn

Logiciel d'enregistrement de la vidéo: RecordCast

Mission 5: déployer et gérer les sauvegardes des données

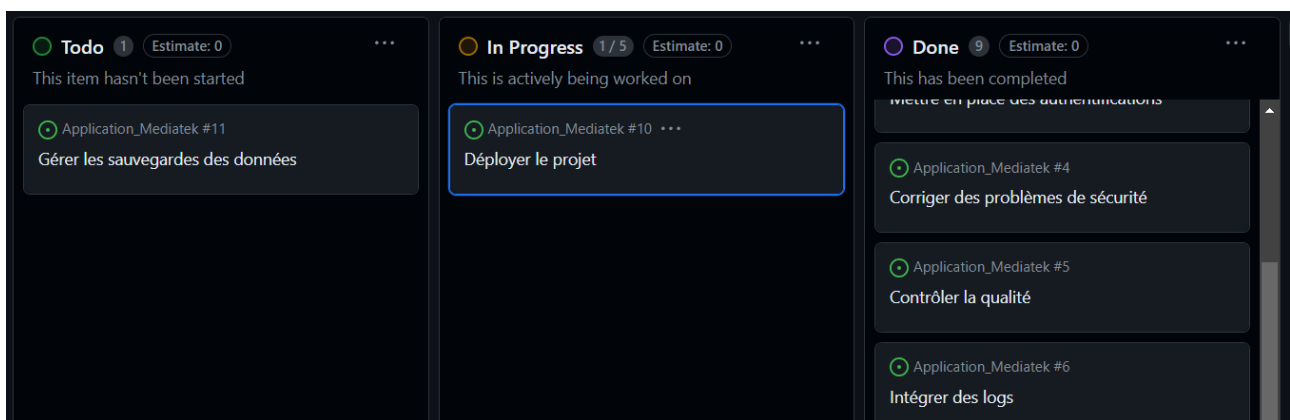
Tâche 1: déployer le projet

Tâche: -Mettre en ligne l'API
-Créer un installeur pour l'application C#

Temps estimé: 3h

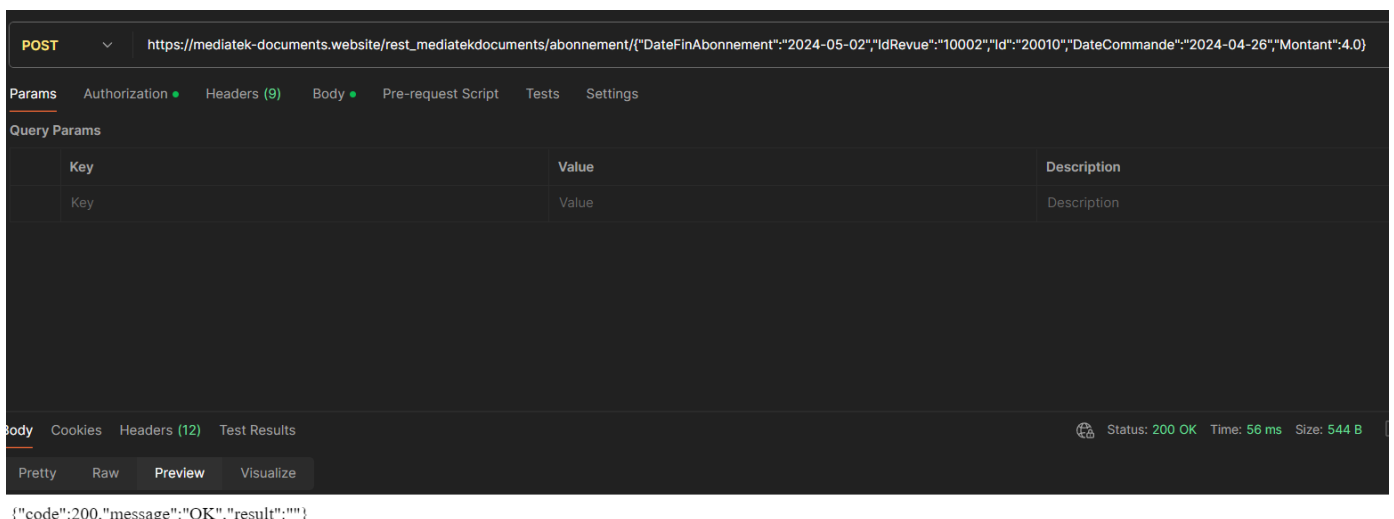
Temps réel: 4 h

Kanban:



Etapas du déploiement de l'api et de la BDD: J'ai choisi de passer par un hébergeur web (Hostinger). J'ai créé un nom de domaine, transféré les fichiers sources de mon API en FTP via FileZilla pour que les fichiers soient stockés dans le gestionnaire de fichier de l'hébergeur. Ensuite j'ai créé une base de donnée avec MySQL de l'hébergeur, j'ai copié le script de la base de donnée locale et je l'ai copié dans la nouvelle base de donnée. J'ai ensuite apporté des modifications au fichier Access.php pour y ajouter les nouvelles informations de connexion à la BDD.

J'ai testé une requête avec l'adresse du site en POST avec Postman :



Etapes du déploiement de l'application C#: J'ai installé dans visual studio l'extension «Microsoft Visual Studio Installer Project» et créé grâce à cette extension un fichier .exe permettant de lancer l'application.

Guide installation de l'application

Aller sur le dépôt de l'application:

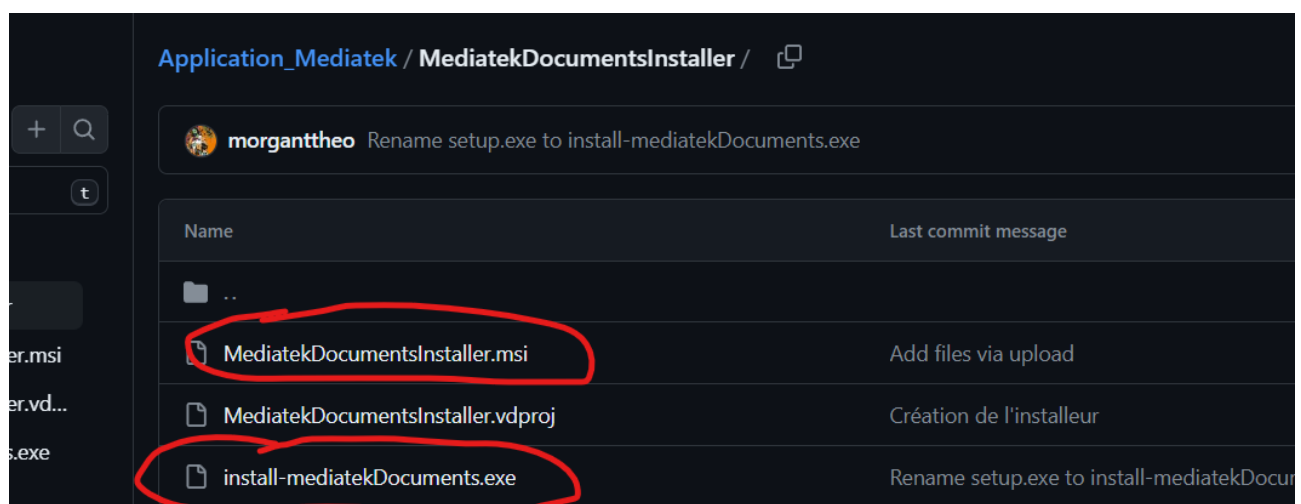
https://github.com/morganttheo/Application_Mediatek/tree/master/MediatekDocumentsInstaller

Installez:

[MediatekDocumentsInstaller.msi](#)

et

[install-mediatekDocuments.exe](#)



Une fois téléchargés, cliquez sur «install-mediatekDocuments», des autorisations vont être demandés, cliquez sur oui.

Une fois l'installation terminée, il suffit de cliquer sur l'icône créée dans le bureau et l'application se lance.

