

---

## ATELIER DE PROFESSIONNALISATION 2

### APPLICATION DE BUREAU C#

#### SOMMAIRE

---

#### 1. INTRODUCTION

- Contexte général du projet "Gestion des Documents de la Médiathèque"
- Objectifs et utilité pour les utilisateurs

---

#### 2. DOCUMENTATION TECHNIQUE

- MISSION 1 : Préparation de l'Environnement et Conception de la Base de Données
- MISSION 2 : Conception des Interfaces et Structuration MVC, Gestion de Version avec Git
- MISSION 3 : Implémentation du Modèle et Connexion à la Base de Données, Documentation Technique
- MISSION 4 : Développement des Fonctionnalités Principales et Tests Unitaires
- MISSION 5 : Déploiement et Résumé du Projet

---

#### 1. INTRODUCTION

---

##### CONTEXTE GENERAL DU PROJET "GESTION DES DOCUMENTS DE LA MÉDIATHÈQUE"

Le projet "Gestion des Documents de la Médiathèque" vise à faciliter la gestion des documents (livres, DVD, revues) au sein des médiathèques via une application de bureau en C#. Cette application adopte une architecture MVC pour séparer les données, la logique métier et l'interface utilisateur, offrant ainsi une gestion centralisée des collections et des emprunts.

---

##### OBJECTIFS ET UTILITÉ POUR LES UTILISATEURS

L'objectif principal est de fournir un outil intuitif pour les bibliothécaires, permettant la gestion efficace des documents et des opérations d'emprunt/retour. Les fonctionnalités incluent la création, la modification et la suppression de documents, ainsi que le suivi des emprunts pour optimiser l'organisation interne des médiathèques.

---

#### 2. DOCUMENTATION TECHNIQUE

---

##### 2.1. MISSION 1 : PRÉPARATION DE L'ENVIRONNEMENT ET CONCEPTION DE LA BASE DE DONNÉES

**OBJECTIFS :** Configurer l'environnement de développement et concevoir la structure de base de données pour la gestion des documents.

## ÉTAPES :

### 1. Configuration de l'Environnement de Développement :

- Installation de MySQL pour le système de gestion de base de données relationnel (SGBDR).
- Configuration initiale de MySQL sur Linux en utilisant les commandes suivantes :

```
bash
Copier le code
sudo apt update
sudo apt install mysql-server
```

### 2. Conception du Schéma de Base de Données :

- Création du schéma initial pour la gestion des documents en utilisant SQL.
- Exemple de script pour la table Documents :

```
sql
Copier le code
CREATE TABLE Documents (
    DocumentID INT PRIMARY KEY AUTO_INCREMENT,
    Titre VARCHAR(255) NOT NULL,
    Auteur VARCHAR(100),
    Genre VARCHAR(50),
    Disponible BOOLEAN DEFAULT TRUE
);
```

- Définition de la structure pour la table `Emprunts` avec une clé étrangère vers `Documents` pour gérer les emprunts.

---

## 2.2. MISSION 2 : CONCEPTION DES INTERFACES ET STRUCTURATION MVC, GESTION DE VERSION AVEC GIT

**OBJECTIFS :** Créer des interfaces utilisateur conviviales, structurer l'application en MVC et mettre en place le contrôle de version avec Git.

## ÉTAPES :

### 1. Conception des Interfaces Utilisateur (UI) :

- Création de maquettes d'interfaces pour la gestion des documents.
- Implémentation en C# pour permettre l'affichage des listes de documents et des formulaires d'ajout.

```
csharp
Copier le code
public class GestionDocumentsView
{
    public void AfficherListeDocuments(List<DocumentModel>
documents)
    {
        // Code pour afficher la liste des documents dans
l'interface utilisateur
    }

    public void AfficherFormulaireAjoutDocument()
    {
```

```

        // Code pour afficher le formulaire d'ajout d'un nouveau
document
    }

    // Méthodes pour gérer les interactions utilisateur
}

```

## 2. Structuration de l'Application en MVC :

- o Organisation du code selon le modèle MVC pour séparer les responsabilités.
- o Exemple de modèle de données et de contrôleur en C# :

```

csharp
Copier le code
public class DocumentModel
{
    public int DocumentID { get; set; }
    public string Titre { get; set; }
    public string Auteur { get; set; }
    public string Genre { get; set; }
    public bool Disponible { get; set; }
}

public class GestionDocumentsController
{
    private DocumentModel documentModel;

    public GestionDocumentsController()
    {
        this.documentModel = new DocumentModel();
    }

    public void AjouterDocument(DocumentModel nouveauDocument)
    {
        // Code pour ajouter le document à la base de données
via le modèle
    }

    // Méthodes pour gérer les opérations CRUD sur les documents
}

```

## 3. Gestion de Version avec Git :

- o Initialisation d'un dépôt Git pour contrôler les versions du code source.
- o Utilisation des commandes Git pour la gestion du dépôt :

```

bash
Copier le code
git init
git add .
git commit -m "Initial commit"
git remote add origin <URL_du_dépôt>
git push -u origin master

```

---

## 2.3. MISSION 3 : IMPLÉMENTATION DU MODÈLE ET CONNEXION À LA BASE DE DONNÉES, DOCUMENTATION TECHNIQUE

**OBJECTIFS :** Implémenter les modèles de données, développer les outils de connexion à la base de données et générer la documentation technique.

### ÉTAPES :

#### 1. Développement des Modèles de Données :

- Création des classes pour gérer les opérations CRUD sur les documents.
- Exemple de classe en C# pour le modèle de données :

```
csharp
Copier le code
public class DocumentRepository
{
    public List<DocumentModel> GetAllDocuments()
    {
        // Code pour récupérer tous les documents depuis la base
de données
        return new List<DocumentModel>();
    }

    public void AjouterDocument(DocumentModel document)
    {
        // Code pour ajouter un document à la base de données
    }

    // Méthodes pour gérer les opérations CRUD sur les documents
}

```

#### 2. Implémentation des Outils de Connexion à la Base de Données :

- Développement d'une classe pour gérer la connexion à la base de données MySQL.
- Exemple de classe en C# pour la gestion de la connexion :

```
csharp
Copier le code
public class DatabaseConnection
{
    private string connectionString;

    public DatabaseConnection(string connectionString)
    {
        this.connectionString = connectionString;
    }

    public void OpenConnection()
    {
        // Code pour ouvrir la connexion à la base de données
    }

    public void CloseConnection()
    {
        // Code pour fermer la connexion à la base de données
    }
}

```

```
        // Méthodes pour exécuter des requêtes SQL et interagir avec
la base de données
    }
```

### 3. Génération de la Documentation Technique :

- Utilisation d'outils dans Eclipse pour générer une documentation technique à partir du code source.
- Création de documentation claire et détaillée pour référence future et maintenabilité du projet.

---

## 2.4. MISSION 4 : DÉVELOPPEMENT DES FONCTIONNALITÉS PRINCIPALES ET TESTS UNITAIRES

**OBJECTIFS :** Implémenter les fonctionnalités principales de l'application basées sur les cas d'utilisation et effectuer des tests unitaires pour valider le bon fonctionnement.

### ÉTAPES :

#### 1. Développement des Fonctionnalités Métier :

- J'ai programmé les fonctionnalités essentielles de l'application telles que la gestion des documents, des utilisateurs, et des emprunts.
- Les services métier ont été développés en respectant le principe de responsabilité unique (Single Responsibility Principle) pour garantir une meilleure maintenabilité du code.

```
csharp
Copier le code
// Exemple de service pour gérer les documents
public class GestionDocumentsService
{
    private DocumentRepository documentRepository;

    public GestionDocumentsService()
    {
        this.documentRepository = new DocumentRepository();
    }

    public List<DocumentModel> ObtenirTousLesDocuments()
    {
        return documentRepository.GetAllDocuments();
    }

    public void AjouterDocument(DocumentModel nouveauDocument)
    {
        documentRepository.AjouterDocument(nouveauDocument);
    }

    // Autres méthodes pour la gestion des documents
    (modification, suppression, etc.)
}
```

## 2. Implémentation des Tests Unitaires :

- Les tests unitaires ont été écrits pour chaque fonctionnalité critique afin de vérifier leur bon fonctionnement.
- J'ai utilisé des frameworks de test comme NUnit ou MSTest pour structurer et exécuter les tests de manière systématique.

```
csharp
Copier le code
[TestFixture]
public class GestionDocumentsServiceTests
{
    private GestionDocumentsService gestionDocumentsService;

    [SetUp]
    public void Setup()
    {
        gestionDocumentsService = new GestionDocumentsService();
    }

    [Test]
    public void
ObtenirTousLesDocuments_RetourneListeDocumentsNonVide()
    {
        // Arrange (Préparation)
        DocumentModel documentTest = new DocumentModel { Titre =
"Test Document", Auteur = "Test Auteur", Genre = "Test Genre",
Disponible = true };
        gestionDocumentsService.AjouterDocument(documentTest);

        // Act (Action)
        List<DocumentModel> documents =
gestionDocumentsService.ObtenirTousLesDocuments();

        // Assert (Vérification)
        Assert.IsNotNull(documents);
        Assert.IsTrue(documents.Count > 0);
    }

    // Autres tests unitaires pour couvrir différentes
fonctionnalités du service
}
```

## 3. Exécution et Validation des Tests :

- Tous les tests unitaires ont été exécutés pour vérifier que chaque fonctionnalité répond aux spécifications et ne provoque pas d'erreurs inattendues.
- J'ai utilisé des données de test représentatives pour couvrir différents scénarios d'utilisation.

## 4. Intégration Continue (CI) :

- Pour automatiser les tests et s'assurer qu'ils sont exécutés régulièrement, j'ai intégré les tests unitaires dans un processus d'intégration continue (CI).
- Utilisation de plateformes comme Jenkins, Azure DevOps, ou GitHub Actions pour l'exécution automatique des tests à chaque validation de code.

---

## 2.5. MISSION 5 : DEPLOIEMENT ET RESUME DU PROJET

**OBJECTIFS :** Déployer l'application sur le poste administratif et rédiger un compte rendu exhaustif du projet.

**ÉTAPES :**

**1. Préparation de l'Application pour le Déploiement :**

- Génération d'un package d'installation (setup.exe) à partir de Visual Studio pour simplifier le déploiement.
- Inclusion de tous les fichiers nécessaires à l'exécution sur un environnement Windows standard.

**2. Installation et Configuration sur le Poste Administratif :**

- Installation de l'application en suivant les bonnes pratiques de déploiement logiciel.
- Vérification de la compatibilité avec l'environnement cible (versions de Windows, prérequis logiciels).

**3. Validation du Bon Fonctionnement :**

- Tests sur le poste administratif pour assurer le bon lancement et le respect des exigences.
- Vérification des fonctionnalités principales comme la gestion des documents et des utilisateurs.

**4. Rédaction du Compte Rendu du Projet :**

- Documentation complète de toutes les phases du projet, de la conception au déploiement.
- Inclusion de captures d'écran des interfaces, extraits de code pertinents et résultats des tests.

**5. Conclusion et Perspectives :**

- Analyse des succès et des défis rencontrés tout au long du projet.
- Identification des améliorations potentielles pour les futures itérations de l'application.

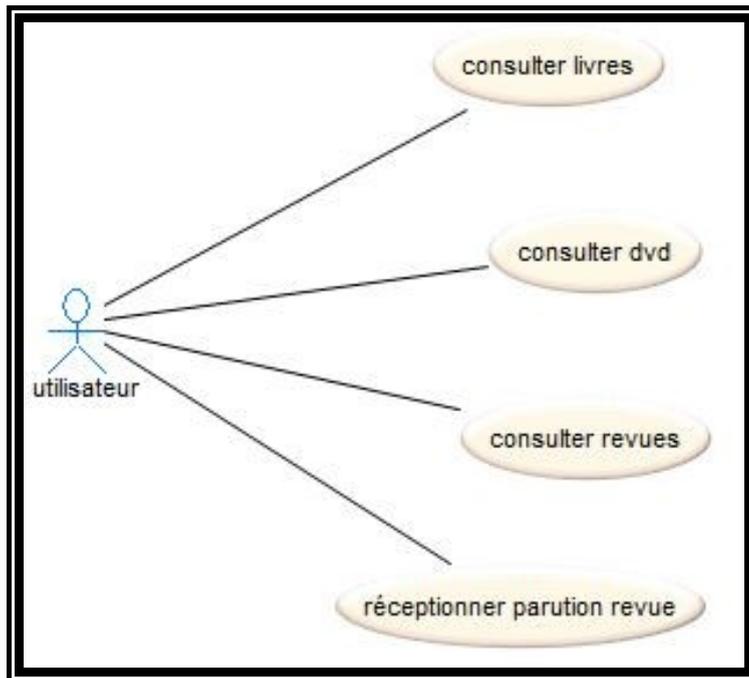
---

## RÉSUMÉ DU PROJET

L'application "MediatekDocuments" est une solution pour la gestion des documents dans un environnement de médiathèque. Basée sur une architecture MVC et développée avec des méthodes Agile, elle répond efficacement aux besoins des utilisateurs finaux. Le déploiement réussi sur le poste administratif témoigne de sa fiabilité opérationnelle.

Les tests réalisés tout au long du développement garantissent la qualité et la stabilité de l'application. Le compte rendu détaillé offre une documentation précise du projet, facilitant ainsi sa maintenance future et son évolution en réponse aux besoins évolutifs des utilisateurs.

Cette mission a permis de mettre en pratique des compétences avancées en développement logiciel et de contribuer significativement à l'amélioration des processus de gestion au sein des médiathèques.



Gestion Médiathèque

Livres 
  DVD 
  Revues 
  Parutions des revues

Recherches

Saisir le titre ou la partie d'un titre :  
 Ou sélectionner le genre :  X

Saisir un numéro de document :   
 Ou sélectionner le public :  Tous publics X

Ou sélectionner le rayon :  X

Id	Titre	Auteur	Collection	Genre	Public	Rayon
00019	Guide Vert - Iles Canaries		Guide Vert	Voyages	Tous publics	Voyages
00020	Guide Vert - Irlande		Guide Vert	Voyages	Tous publics	Voyages
00011	L'île des oubliés	Victoria Hislop		Roman	Tous publics	Littérature française
00006	La marque jaune	Edgar P. Jacobs	Blake et Mortimer	Bande dessinée	Tous publics	BD Adultes
00026	La planète des singes	Pierre Boule	Julliard	Science Fiction	Tous publics	Littérature française
00012	La souris bleue	Kate Atkinson		Roman	Tous publics	Littérature française
00018	Le Routard - Maroc		Guide du Routard	Voyages	Tous publics	Voyages
00014	Mauvaise étoile	RJ Elory		Policier	Tous publics	Policiers français étrangers

Informations détaillées

Numéro de document :  00026 
 Code ISBN :  2266283022

Titre :  La planète des singes

Auteur(e) :  Pierre Boule

Collection :  Julliard

Genre :  Science Fiction

Public :  Tous publics

Rayon :  Littérature française

Chemin de l'image :  C:\Images\Livres\La\_planète\_des\_singes.jpg



Gestion des documents de la médiathèque

Livres  DVD  Revues  Parutions des revues

Recherche revue

Numéro revue :

Titre :

Périodicité :

Délai mise à dispo :

Genre :

Public :

Rayon :

Chemin de l'image :

Parutions :
 

Numero	DateAchat	Photo
505	16/10/2022	
514	01/12/2021	
513	01/11/2021	
512	06/10/2021	
511	01/09/2021	

Image revue :

Image exemplaire :

---

Nouvelle parution réceptionnée pour cette revue

Numéro réceptionné :

Date de parution :

Emplacement image :

Image exemplaire :

