
ATELIER DE PROFESSIONNALISATION 2

APPLICATION DE BUREAU C#

SOMMAIRE

1. INTRODUCTION

- Contexte général du projet "Gestion du Personnel des Médiathèques"
- Objectifs et utilité pour les utilisateurs

2. DOCUMENTATION TECHNIQUE

- MISSION 1 : Préparer l'Environnement de Travail et Créer la Base de Données
- MISSION 2 : Dessiner les Interfaces, Structurer l'Application en MVC, Créer un Dépôt, Coder le Visuel
- MISSION 3 : Coder le Modèle et les Outils de Connexion, Générer la Documentation Technique
- MISSION 4 : Coder les Fonctionnalités de l'Application à Partir des Cas d'Utilisation
- MISSION 5 : Gérer le Déploiement, Rédiger le Compte Rendu

1. INTRODUCTION

CONTEXTE GENERAL DU PROJET "GESTION DU PERSONNEL DES MEDIATHEQUES"

Le projet "Gestion du Personnel des Médiathèques" est une application de bureau développée en C# visant à faciliter la gestion des employés dans diverses médiathèques. L'application doit permettre de gérer les informations du personnel, les affectations aux services, et les absences.

OBJECTIFS ET UTILITE POUR LES UTILISATEURS

L'objectif principal est de fournir un outil intuitif et efficace pour les administrateurs des médiathèques, permettant une gestion simplifiée et centralisée du personnel. L'application vise à améliorer la productivité et l'organisation interne.

2. DOCUMENTATION TECHNIQUE

2.1. MISSION 1 : PREPARER L'ENVIRONNEMENT DE TRAVAIL ET CREER LA BASE DE DONNEES

OBJECTIFS :

- Installer et configurer MySQL ou MariaDB pour le SGBDR.
- Créer la structure de base de données pour la gestion du personnel des médiathèques.

TACHES :

1. **Télécharger et Installer MySQL ou MariaDB :**
 - **Étape 1 :** Télécharger le fichier d'installation de MySQL ou MariaDB depuis le site officiel.
 - **Étape 2 :** Suivre les instructions d'installation pour configurer le serveur de base de données.
 - **Étape 3 :** Créer un utilisateur avec les privilèges nécessaires pour gérer la base de données.
2. **Concevoir le Schéma de Base de Données :**
 - **Étape 1 :** Identifier les entités principales : Personnel, Services, Absences.
 - **Étape 2 :** Définir les attributs nécessaires pour chaque entité (par exemple, pour Personnel : ID, Nom, Prénom, Service, etc.).
 - **Étape 3 :** Utiliser un outil de modélisation (comme MySQL Workbench) pour créer le schéma de base de données.
 - **Étape 4 :** Générer le script SQL pour créer les tables.
3. **Établir les Relations entre les Entités :**
 - **Étape 1 :** Définir les clés primaires et étrangères pour chaque table.
 - **Étape 2 :** Mettre en place les relations entre les tables (par exemple, une relation entre Personnel et Services).
 - **Étape 3 :** Exécuter le script SQL pour créer la base de données dans MySQL ou MariaDB.

2.2. MISSION 2 : DESSINER LES INTERFACES, STRUCTURER L'APPLICATION EN MVC, CREER UN DEPOT, CODER LE VISUEL

OBJECTIFS :

- Concevoir les interfaces utilisateur (UI) pour la gestion du personnel.
- Structurer l'application en architecture MVC (Modèle-Vue-Contrôleur).
- Créer un dépôt Git pour le contrôle de version du code.

TACHES :

1. **Créer les Maquettes d'Interfaces Utilisateur (UI) :**
 - **Étape 1 :** Utiliser un outil de conception d'UI (comme Figma ou Adobe XD) pour dessiner les maquettes.
 - **Étape 2 :** Définir les interfaces pour chaque fonctionnalité principale (par exemple, ajout de personnel, gestion des absences).
 - **Étape 3 :** Valider les maquettes avec les utilisateurs potentiels pour s'assurer de leur convivialité.

2. Définir les Modèles de Données et les Contrôleurs :

- **Étape 1** : Créer des classes de modèles en C# pour représenter les entités (Personnel, Services, Absences).
- **Étape 2** : Développer les contrôleurs pour gérer les interactions entre l'utilisateur et le modèle.
- **Étape 3** : Implémenter les vues en utilisant Windows Forms ou WPF pour l'affichage des données.

3. Initialiser un Dépôt Git :

- **Étape 1** : Ouvrir un terminal et initialiser un dépôt Git local avec la commande `git init`.
- **Étape 2** : Ajouter tous les fichiers de projet et faire le premier commit avec `git add .` et `git commit -m "Initial commit"`.
- **Étape 3** : Lier le dépôt local à un dépôt distant sur GitHub ou GitLab en utilisant `git remote add origin <URL_du_dépôt>` et pousser les modifications avec `git push -u origin master`.

2.3. MISSION 3 : CODER LE MODELE ET LES OUTILS DE CONNEXION, GENERER LA DOCUMENTATION TECHNIQUE

OBJECTIFS :

- Implémenter les modèles de données pour la gestion du personnel.
- Écrire les outils de connexion à la base de données.
- Générer une documentation technique complète du projet.

TACHES :

1. Développer les Classes et Méthodes pour les Modèles de Données :

- **Étape 1** : Créer des classes C# pour chaque entité (Personnel, Services, Absences) avec les attributs nécessaires.
- **Étape 2** : Ajouter des méthodes pour les opérations CRUD (Create, Read, Update, Delete).
- **Étape 3** : Tester chaque modèle pour vérifier qu'il fonctionne correctement avec les données.

2. Établir les Connexions à la Base de Données :

- **Étape 1** : Utiliser les bibliothèques ADO.NET ou Entity Framework pour gérer les connexions.
- **Étape 2** : Écrire des classes de connexion pour interagir avec la base de données.
- **Étape 3** : Tester les connexions pour s'assurer qu'elles fonctionnent correctement et de manière sécurisée.

3. Générer la Documentation Technique :

- **Étape 1** : Créer des diagrammes UML pour représenter la structure et les relations entre les classes.
- **Étape 2** : Utiliser un outil de documentation comme Doxygen ou Sandcastle pour générer la documentation du code.
- **Étape 3** : Compiler une documentation complète incluant les diagrammes, les descriptions de classes, et les guides d'utilisation.

2.4. MISSION 4 : CODER LES FONCTIONNALITES DE L'APPLICATION A PARTIR DES CAS D'UTILISATION

OBJECTIFS :

- Implémenter les fonctionnalités principales de gestion du personnel (affectation à un service, gestion des absences).
 - Effectuer des tests unitaires pour valider les fonctionnalités.
-

TACHES :

1. Programmer les Actions et Logiques Métier :

- **Étape 1** : Écrire les méthodes pour gérer l'affectation du personnel aux services.
- **Étape 2** : Implémenter la gestion des absences en permettant aux administrateurs de suivre les congés et les maladies.
- **Étape 3** : Ajouter des notifications pour les événements importants (par exemple, un employé en congé).

2. Mettre en Place la Gestion des Absences :

- **Étape 1** : Créer des formulaires pour l'entrée des absences.
- **Étape 2** : Écrire la logique pour calculer les jours de congé restants.
- **Étape 3** : Implémenter des rapports pour suivre les absences au fil du temps.

3. Développer des Tests Unitaires :

- **Étape 1** : Utiliser un framework de test comme NUnit ou MSTest pour créer des tests unitaires.
 - **Étape 2** : Écrire des tests pour chaque méthode critique afin de vérifier leur bon fonctionnement.
 - **Étape 3** : Exécuter les tests régulièrement pour s'assurer qu'aucune nouvelle modification ne casse le code existant.
-

2.5. MISSION 5 : GERER LE DEPLOIEMENT, REDIGER LE COMPTE RENDU

OBJECTIFS :

- Déployer l'application sur le poste du service administratif.
 - Compiler un compte rendu détaillé du projet.
-

TACHES :

1. Installer et Configurer l'Application :

- **Étape 1** : Créer un fichier exécutable (EXE) pour l'application en utilisant Visual Studio.
- **Étape 2** : Installer l'application sur le poste administratif en suivant les meilleures pratiques de déploiement.
- **Étape 3** : Vérifier que l'application fonctionne correctement dans l'environnement cible.

2. Écrire un Compte Rendu du Projet :

- **Étape 1** : Documenter chaque étape du développement, des premières réunions jusqu'au déploiement final.

- **Étape 2** : Inclure des captures d'écran des interfaces, des extraits de code, et des résultats de tests.
- **Étape 3** : Préparer une section de conclusion avec les leçons apprises et les améliorations possibles pour les futures versions de l'application.

3. SITUATION PROFESSIONNELLE

3.1. PRESENTATION DU CONTEXTE

DESCRIPTION DE L'ENTREPRISE

L'entreprise pour laquelle j'ai travaillé est spécialisée dans le développement de logiciels de gestion pour diverses institutions. Grâce à une croissance rapide, l'entreprise a récemment recruté une équipe de développeurs juniors, dont je fais partie, pour participer à divers projets en développement d'applications de bureau.

OBJECTIFS DE LA MISSION

Cette mission a représenté une occasion idéale pour évaluer mes compétences en développement C#. J'ai été encadré par un responsable qui m'a fourni les documents et productions nécessaires et m'a également guidé dans mes tâches.

3.2. PRESENTATION DE L'APPLICATION

DESCRIPTION DE L'APPLICATION DE GESTION DU PERSONNEL

L'application que je dois créer est un logiciel de gestion du personnel destiné aux ordinateurs de bureau. Elle doit permettre aux administrateurs de gérer les informations du personnel, les affectations aux services et les absences.

FONCTIONNALITES PRINCIPALES ET OBJECTIFS TECHNIQUES

- Chaque administrateur peut gérer les informations du personnel.
- Gestion des affectations du personnel à divers services.
- Gestion des absences (congés, maladies) et notifications.
- Développement en C# avec une architecture MVC.

3.3. MA TACHE

DETAIL DES ETAPES DE DEVELOPPEMENT

1. **Étape 1** : Préparation de l'environnement de travail et création de la base de données.
2. **Étape 2** : Dessin des interfaces et structuration de l'application en MVC.
3. **Étape 3** : Codage des modèles de données et des outils de connexion.
4. **Étape 4** : Implémentation des fonctionnalités de gestion du personnel.
5. **Étape 5** : Déploiement de l'application et rédaction du compte rendu.

PLAN DE TESTS POUR CHAQUE ETAPE

- Définir des tests unitaires pour vérifier la fonctionnalité de chaque composant.
- Utiliser des scénarios de test pour valider les interactions entre les composants.