

1. Mediatek86

- 1. Mediatek86
 - 1.1. Description
 - 1.2. Avertissement
 - 1.3. Licence
 - 1.3.1. BddManager.cs
 - 1.4. Documentation du code source
 - 1.4.1. SandCastle pour Visual Studio 2022
 - 1.5. CI/CD
 - 1.6. Architecture
 - 1.7. Contexte de l'application
 - 1.8. Fonctionnalités
 - 1.8.1. Contrôle d'accès à la base de données
 - 1.8.2. Contrôle d'accès à l'application
 - 1.8.2.1. Dépannage de la connexion ou modification des paramètres de connexion
 - 1.8.3. Gestion du personnel
 - 1.9. Processus de développement
 - 1.9.1. Installation des outils de développement
 - 1.9.2. Conception de la base de données

1.1. Description

Une solution pour la médiathèque de Vienne

1.2. Avertissement

Ce projet est un projet scolaire réalisé dans le cadre de la formation BTS-SIO première année.

Il est fourni tel quel et n'est pas destiné à être utilisé en production.

L'application n'est pas conforme aux normes de sécurité et de qualité attendues pour une application professionnelle.

Elle est diffusée ici à titre d'exemple pour illustrer les compétences acquises par l'auteur dans le cadre de sa formation.

L'auteur ne fournit aucune garantie quant à son fonctionnement et ne pourra être tenu pour responsable de tout dommage causé par son utilisation.

1.3. Licence

Ce projet est sous licence MIT. Cela signifie que vous pouvez l'utiliser, le modifier et le distribuer comme bon vous semble, à condition de conserver la licence MIT dans les fichiers modifiés.

1.3.1. BddManager.cs

Le fichier BddManager.cs est une adaptation du code fourni par le professeur et n'est pas concerné par la licence MIT. Il reste la propriété de l'auteur original et ne doit pas être utilisé en dehors du cadre de ce projet.

1.4. Documentation du code source

Le code source est documenté en utilisant le format XMLDoc. Le compilateur C# génère automatiquement un fichier XML contenant la documentation du code source. Ce fichier est disponible dans le répertoire `bin\Debug` ou `bin\Release` après la compilation du projet.

1.4.1. SandCastle pour Visual Studio 2022

Conformément au cahier des charges, la documentation du code source est également générée avec SandCastle. Le projet SandCastle est disponible dans la solution sous le nom `Documentation`. Elle est déployée dans GitHub Pages à l'adresse suivante : [Documentation](#)

1.5. CI/CD

Le projet est configuré pour utiliser GitHub Actions pour la CI/CD.

Le workflow est défini dans le fichier `.github/workflows/ci.yml`.

La publication de la documentation est effectuée automatiquement à chaque push sur la branche `main` via le workflow `.github/workflows/static-pages.yml`.

1.6. Architecture

Cette application est écrite en C# et utilise le framework .NET 8.0 accompagné de la couche interface utilisateur Microsoft WPF.

Ses données sont stockées dans une base de données MySQL installée sur le poste de l'utilisateur.

L'accès à la base de données se fait via Entity Framework Core et la classe d'accès `BddManager`.

1.7. Contexte de l'application

Dans le cadre de la formation BTS-SIO Première année il est proposé de réaliser une application imaginaire pour la médiathèque de Vienne.

Voici comment est présenté le projet.

Nous avons été intégré dans l'entreprise `ESN InfoTech Services 86` en tant que développeur junior.

L'entreprise nous a alors confié la réalisation d'une application Windows pour la médiathèque de Vienne.

Cette application doit permettre de gérer les absences du personnel dans une médiathèque.

L'application est mono-utilisateur et doit permettre de :

- Consulter la liste des employés
- Ajouter un employé
- Modifier un employé
- Consulter la liste des absences
- Ajouter une absence
- Modifier une absence
- Supprimer une absence

1.8. Fonctionnalités

Une étude du cahier des charges fourni par l'entreprise nous a permis de définir les fonctionnalités suivantes :

1.8.1. Contrôle d'accès à la base de données

La connexion entre la base de données et l'application est sécurisée par un contrôle d'accès. Un utilisateur de la base de données doit être créé avec les droits nécessaires pour accéder à la base de données. La chaîne de connexion est stockée dans le fichier `App.config` situé dans le répertoire de l'application.

Voici par exemple une configuration valide pour la chaîne de connexion et les paramètres de runtime:

```
<?xml version="1.0" encoding="utf-8" ?>

<configuration>
  <configSections>
    <section name="system.data"
type="System.Data.Common.DbProviderFactoriesConfigurationSection, System.Data,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
    <section name="entityFramework"
type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection,
EntityFramework, Version=6.0.0.0, Culture=neutral" requirePermission="false" />
  </configSections>
  <entityFramework>
codeConfigurationType="MySQL.Data.EntityFramework.MySqlEFConfiguration,
MySQL.Data.EntityFramework">
    <defaultConnectionFactory
type="System.Data.Entity.Infrastructure.SqlConnectionFactory, EntityFramework"/>
    <providers>
      <provider invariantName="MySQL.Data.MySqlClient"
type="MySQL.Data.MySqlClient.MySqlProviderServices,
MySQL.Data.EntityFramework"/>
    </providers>
  </entityFramework>
  <connectionStrings>
    <add name="MyMediatek86DbContext"
providerName="MySQL.Data.MySqlClient"

connectionString="server=localhost;database=mediatek86;user=mediatek86;password=me
diatek86pwd" />
  </connectionStrings>
</configuration>
```

1.8.2. Contrôle d'accès à l'application

L'accès à l'application est sécurisé par un contrôle d'accès vérifiant les identifiants de l'utilisateur. Cette vérification est effectuée en comparant les identifiants saisis par l'utilisateur avec ceux stockés dans la base de données. Par souci de sécurité il a été convenu de ne pas stocker les mots de passe en clair dans la base de données. Les mots de passe sont donc hashés avant d'être stockés. Le hashage est effectué avec l'algorithme SHA256. Le hash est stocké sous forme de chaîne hexadécimale dans la base de données.

1.8.2.1. Dépannage de la connexion ou modification des paramètres de connexion

Il n'a pas été prévu dans le cahier des charges de l'application de permettre à l'utilisateur de modifier les paramètres de connexion à l'application. Cependant, il est possible de modifier les paramètres de connexion en modifiant le contenu de la base de données. Pour ce faire, il est nécessaire de se connecter à la base de données avec un outil tel que MySQL Workbench. Il est alors possible de modifier les paramètres de connexion dans la table `responsable` de la base de données.

Voici comment créer l'utilisateur `admin` avec le mot de passe `adminpwd` :

```
INSERT INTO `responsable` (`login`, `pwd`) VALUES ('admin', SHA2('adminpwd', 256));
```

Voici comment modifier le mot de passe de l'utilisateur `admin` pour le remplacer par `newpwd` :

```
UPDATE `responsable` SET `pwd` = SHA2('newpwd', 256) WHERE `login` = 'admin';
```

1.8.3. Gestion du personnel

L'application permet de consulter la liste des employés, d'ajouter un employé, de modifier un employé.

Le fonctionnement est simple; sur la page d'accueil de l'application, la liste des employés est affichée.

Un bouton `Ajouter` permet d'ajouter un employé.

Après avoir sélectionné un employé dans la liste, le bouton `Modifier` permet de modifier les informations de l'employé. Après avoir sélectionné un employé dans la liste, le bouton `Supprimer` permet de supprimer l'employé.

1.9. Processus de développement

Le développement de l'application a été réalisé en plusieurs étapes.

1.9.1. Installation des outils de développement

- Installation de Visual Studio 2022
 - Les options C# et .NET 8.0 sont installées
- Installation de MariaDB 11.4.2 [depuis le site officiel](#)
- Installation de Looping MCD 4.0 [depuis le site officiel](#)

1.9.2. Conception de la base de données

Le modèle conceptuel de données a été réalisé avec Looping MCD 4.0.

La table `responsable` a été créée dans Looping puis [le script de création des tables](#) a été généré et a été complété pour créer la base de données appelée `mediatek86` dans le même script.