

MediaTek 86 - Gestion du personnel

Nom	Prenom	Service	Tel	Email
Dima	Amadou	administratif	02 55 00 44 00	ama33@society.mail
Dima	John	administratif	01 33 00 00 00	dima01@email.mail
Johnson	Lea	prêt	04 00 00 00 08	1984@ad.join.fr
Mado	Anna	prêt	08 21 00 00 21	Mado@society.com.fr
Madys	Ygor	médiation	07 04 08 04 00	gorma@society.fr
manier	Jacques	administratif	07 06 05 03 04	manier@synths.email
Marau	Myriam	prêt	09 00 00 00 20	MYmara@monemail.com.fr
Nashamura	Takumi	administratif	06 07 08 09 10	taku@initial-d.email
Nurry	Franck	prêt	09 00 00 01 88	NurrFranck@email.com.fr
Piotrsky	Lise	médiation	01 00 02 03 04	lise@euro.email
Piotrsky	Pierre	prêt	01 00 02 03 08	pierrepotr@euro.email
Poliak	Clara	administratif	07 00 02 03 08	MissClara@email.com.fr
Ramb	John	administratif	07 01 02 03 04	Johnny@1984.email
Sanchez	Marco	médiation	09 00 00 01 88	MarcoSan@email.com
Scatt	Ridley	prêt	01 88 00 00 08	1984@society.mail.fr
Steinbrick	William	prêt	04 00 00 00 33	willstein@php.mail

Ajouter/Modifier personnel

Nom
Poliak

Prenom
Clara

Service
administratif

Téléphone
07 00 02 03 08

Email
MissClara@email.com.fr

Ajouter

Modifier

Supprimer

Afficher absences

MediaTek 86 - Rapport

Atelier 2 - première année BTS SIO

Paul Thorel

Sommaire

- Étape 1 - Mise en place de l'environnement.
- Étape 2 - Créer et coder le visuel.
- Étape 3 - Générer une documentation, les outils de connexion et modèles.
- Étape 4 - Coder les fonctionnalités en se basant sur un dossier et des user cases.
- Étape 5 - Documentation vidéo pour les utilisateurs.
- Étape 6 - Déploiement - compte rendu - documentation.
- Bilan.

Rappel contexte:

Afin de développer votre projet dans un contexte réaliste d'entreprise, nous vous proposons le contexte MediaTek86, 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.

Vous travaillez en tant que technicien développeur junior pour l'ESN InfoTech Services 86 qui vient de remporter le marché pour différentes interventions au sein du réseau MediaTek86, dont certaines dans le domaine du développement d'application.

Il vous a été confié le développement de l'application de bureau qui va permettre de gérer le personnel de chaque médiathèque, leur affectation à un service et leurs absences.

Cette application est monoposte et sera installée sur un poste du service administratif.

Au niveau du SGBDR, vous pouvez travailler avec MySQL ou MariaDB. Au niveau du langage, vous avez le choix entre Java et C#.

Étape 1 - Mise en place de l'environnement.

Il a fallu tout d'abord installer les différents outils nécessaires:

- Wampserver - Plateforme de développement web avec phpmyadmin et mysql [Pour installer la base de données]
- VisualStudio 2022 - IDE pour programmer en C# et créer l'application (ainsi que le débogage)
- Doxygen - Outil de création de documentation au format HTML

Ensuite, j'ai dû ajouter un nouvel utilisateur dans mysql pour gérer la future base de données (il faut éviter d'utiliser root qui a tous les pouvoirs)

```
CREATE USER 'compteacces'@'%' IDENTIFIED BY [REDACTED];
GRANT USAGE ON gestionperso.* TO 'compteacces';
GRANT ALL PRIVILEGES ON gestperso.* TO 'compteacces';
FLUSH PRIVILEGES;
```

Puis, en se basant sur le contexte et le dossier, j'ai créé les tables nécessaire dans la base de donnée "gestionperso" avec phpmyAdmin

- responsable (avec le mot de passe crypté et le login)
- absence
- motif
- personnel
- service

Table ▾	Action
<input type="checkbox"/> absences	☆ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> motif	☆ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> personnel	☆ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> responsable	☆ Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/> service	☆ Parcourir Structure Rechercher Insérer Vider Supprimer
5 tables	Somme

[↑](#) Tout cocher ▾

[Imprimer](#) [Dictionnaire de données](#)

[Créer une nouvelle table](#)

Une fois que tout fonctionne, je passe à la partie 2.

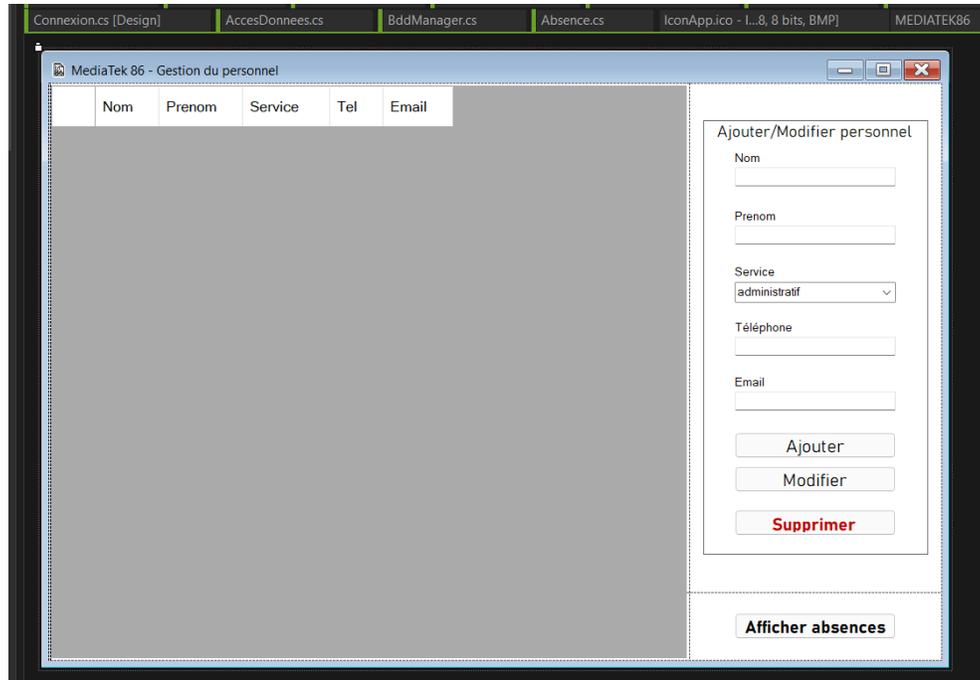
Étape 2 - Créer et coder le visuel

Les objectifs sont ici assez clair, il s'agit de designer la partie visuelle de l'application (du front-end donc), et j'ai utilisé l'outil intégré à Visual Studio 2022, le "Designer".

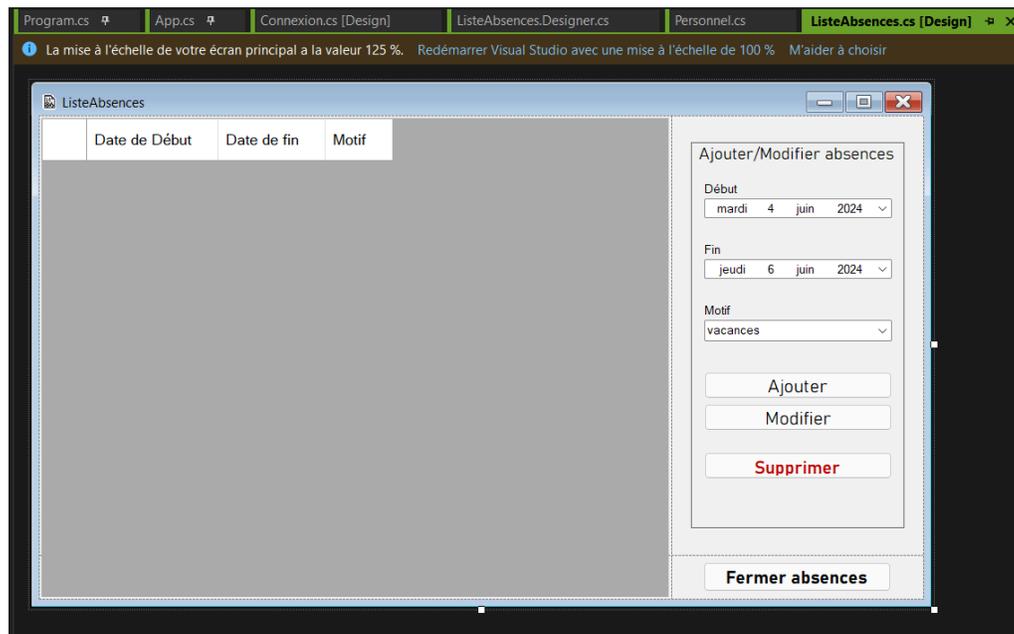
Voici des captures d'écrans de l'état du travail accompli à cet étape:



Fenêtre de connexion



Fenêtre principale avec la liste des personnels et les outils à dispositions



Fenêtre des absences avec la liste des absences et les outils à dispositions

A ce stade, les interfaces sont dessinées dans l'outil, l'application est créée et le visuel des interfaces est codé (en partie généré par Designer par ailleurs)

Étape 3 - Générer une documentation, créer les outils de connexion et les modèles.

Ensuite, j'ai codé les modèles nécessaires et j'ai généré une première documentation technique avec doxygen (je préfère cet outil car il fonctionne assez facilement en invite de commandes).

J'ai utilisé une classe BddManager assez simple pour "dialoguer" avec la base de donnée mysql distante. (Par ailleurs, il a fallu ajouter une référence pour le connecteur mysql nécessaire afin d'accéder aux bases de données).

J'ai créé un package (Dal) et j'y ai inséré une classe pour répondre au demande d'un controleur en utilisant la classe BddManager (qu accède à la base de données , pour faire simple)

Puis je commence à déboguer mon application et corriger les éventuelles erreurs, tout en créant les modèles que le controleur et l'application allaient utiliser.

Étape 4 - Coder les fonctionnalités en se basant sur un dossier et des use cases.

Ici, c'est la partie la plus importante de l'application: son fonctionnement.

J'ai créé une classe Controle qui va permettre de communiquer entre le Dal et les actions utilisateurs (la Vue).

L'utilisation de phpmyadmin pour vérifier les requêtes utilisées dans le Dal et que le contrôleur appelait m'a permis ici de déboguer certaines erreurs (champs mal remplis par l'utilisateur)

A ce stade, l'application est plus ou moins finie, il ne reste plus qu'à la rendre utilisable (étape 6) pour les utilisateurs lambdas.

Nom	Prenom	Service	Tel	Email
Dima	Amadou	administratif	02 55 00 44 00	ama33@society.mail
Dima	John	administratif	01 33 00 00 00	dima01@email.mail
Johnson	Lea	prêt	04 00 00 00 08	1984@ad.join.fr
Mado	Anna	prêt	08 21 00 00 21	Mado@society.com.fr
Madys	Ygor	médiation	07 04 08 04 00	gorma@society.fr
manier	Jacques	administratif	07 06 05 03 04	manier@synths.email
Marau	Myriam	prêt	09 00 00 00 20	MYmara@monemail.com.fr
Nashamura	Takumi	administratif	06 07 08 09 10	taku@initial-d.email
Nurry	Franck	prêt	09 00 00 01 88	NurrFranck@email.com.fr
Piotrsky	Lise	médiation	01 00 02 03 04	lise@euro.email
Piotrsky	Pierre	prêt	01 00 02 03 08	pierrepiootr@euro.email
Poliak	Clara	administratif	07 00 02 03 08	MissClara@email.com.fr
Ramb	John	administratif	07 01 02 03 04	Johnny@1984.email
Sanchez	Marco	médiation	09 00 00 01 88	MarcoSan@email.com
Scatt	Ridley	prêt	01 88 00 00 08	1984@society.mail.fr
Steinbrick	William	prêt	04 00 00 00 33	willstein@ohh.mail

Ajouter/Modifier personnel

Nom
Nurry

Prenom
Franck

Service
prêt

Téléphone
09 00 00 01 88

Email
NurrFranck@email.com.fr

Ajouter

Modifier

Supprimer

Afficher absences

Fenêtre principale avec le tableau des personnels et les outils disponibles.

Étape 5 - Documentation vidéo pour les utilisateurs.

J'ai utilisé l'outil capture d'écran de Windows 11 pour enregistrer la vidéo, et j'ai fait une démonstration de l'application avec les différentes fonctionnalités.

J'ai utilisé Ableton pour l'enregistrement, et KDeLive pour le montage, où j'ai rajouté des textes de descriptions pour "illustrer" l'audio.

Étape 6 - Déploiement - compte rendu - documentation.

Afin de pouvoir “finir” cette application, je dois la rendre utilisable pour les futurs utilisateurs, en la déployant.

Je commence par récupérer un script SQL de la base de données complète, avec toutes les tables (qui ne sera pas effacée si elle existe déjà sur la machine hôte).

J’y ajoute l’instruction précisée en partie 1 (création du compte pour accéder à cette base), et je met le tout dans un fichier sql qu’il faudra exécuter avant de lancer l’application (tout est précisé dans la documentation même si celle ci est assez simple ;))

Puis je rédige ce compte-rendu et c’est à ce moment là que mon antivirus ainsi que plusieurs vm ont commencé à alerter sur le fait que j’avais compilé une application contenant des trojans.

J’ai donc supprimé les fichiers demandés, et j’ai décidé de ne pas fournir normalement le msi qui avait commencé à tout “abimer”, mais simplement un zip de la source ainsi qu’un zip de l’application à extraire. (et un msi dans les releases)

*Le bilan est à la page suivante.

Bilan.

J'ai apprécié travailler sur cette application, que ce soit sur le front-end (certes Winforms ne propose pas des miracles mais remplit sa tâche sur des applications de bureau simples) et sur le back-end (surtout en SQL où j'ai apprécié la syntaxe parfois déconcertante et simpliste) .

C'est la première fois que je code une application de bureau en C# (et quasiment la plus) et cela confirme que je ne me suis pas trompé en choisissant cette filière.

C'était assez complexe de gérer tous les cas d'usages et les situations possibles (champs vides, erreurs de dates) mais j'ai vu tout ça comme un défi et je suis satisfait du résultat.

(Je pensais même que je ferais sans doute une version 2 en dehors de ce devoir avec un aspect graphique plus "moderne")

J'aurais néanmoins aimé que cette histoire de faux-positif/virus à la fin ne m'empêche pas de rendre rapidement le devoir comme demandé.