

Compte rendu d'activité

Atelier de professionnalisation — 1^{ère} année

SEN Monica Tevy

BTS SIO SLAM

04 Juin 2021

SOMMAIRE

INTRODUCTION	4
LANGAGE & TECHNOLOGIES	4
ÉTAPE 1	5
1. Modèle conceptuel de données (MCD)	6
2. Modèle logique de données (MLD)	6
3. Modèle physique de données (MPD)	7
4. Création de la base de données (BDD)	7
5. Création d'un utilisateur ayant les droits d'accès	8
6. Ajouter une table 'responsable' (login/pwd)	9
7. Alimenter la base de donnée	10
ÉTAPE 2	13
1. Prototype GUI	14
2. Création du dépôt distant	17
3. Structurer l'application suivant le modèle MVC	17
4. Coder la partie Vue de l'application	18
5. Sauvegarde sur le dépôt distant	20
ÉTAPE 3.....	22
1. Configurer l'IDE pour accéder à la BDD	22
2. Création du package connexion	23
3. Création du package dal	24
4. Création des classes du package Modèle	25
5. Génération de la documentation technique	27
ÉTAPE 4	30

1. La démarche	30
2. Les difficultés rencontrées	33
ÉTAPE 5	37
ÉTAPE 6	37
1. Déploiement de l'application	37
2. Création de la page du portfolio	38
ATELIER DE PROFESSIONNALISATION	39
BILAN.....	39

INTRODUCTION

Mediatek86 est un réseau qui regroupe les responsables des médiathèques du département de Vienne (86). Les médiathèques sont des établissements qui conservent et mettent à disposition différents types de médias (livres, journaux, livres numériques, DVD, cédérom etc...), permettant la consultation sur place et l'emprunt à domicile. Avec le développement de ses activités, MediaTek86 a fait appel à l'ESN (Entreprise de Services Numériques) InfoTech Service 86 (ITS86) afin de gérer le parc informatique des médiathèques, son infrastructure système et réseau, et de développer son portail web et ses applications internes. Les activités d'ITS86 s'organisent autour du pôle Développement ainsi que le pôle Système et réseau.

En tant que développeur junior, je serai accompagnée par l'équipe ITS86 de la conception au déploiement d'une application de bureau qui permettra de gérer le personnel de chaque médiathèque, leur affectation à un service et leurs absences. Les collaborateurs et partenaires d'ITS86 apporteront leur expertise afin de proposer une solution adaptée et sur mesure à MediaTek86.

LANGAGE & TECHNOLOGIES

Langage de programmation : C#

Environnement de développement (IDE) : Visual Studio Code (VS)

Serveur : WAMPServer (Windows, Apache, MySQL, PHP)

Système de gestion de base de données (SGBDR) : MySQL

Modélisation : Langage de modélisation unifié (UML)

Outil de versionning : Github, Git

ÉTAPE 1

Préparer l'environnement de travail
Créer la base de données



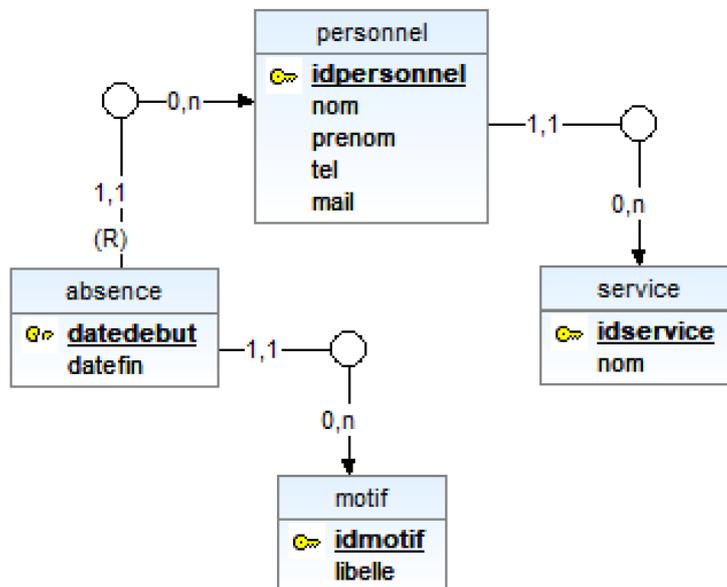
La création d'une application pose de nombreuses problématiques auxquelles il est important de savoir anticiper et y répondre avant de se lancer dans l'écriture du code. Après analyse de celles-ci, les équipes de développement ont élaboré un modèle conceptuel de données (MCD), dans lequel les données à utiliser et les relations entre elles sont définies.

Le processus de modélisation se poursuit avec la traduction du modèle conceptuel en modèle relationnel, cela peut être découpée en deux étapes :

- la conception du modèle logique de données (MLD): représentation en tables indépendantes du SGBD
- la traduction en un modèle physique de données (MPD): propre à un SGBD.

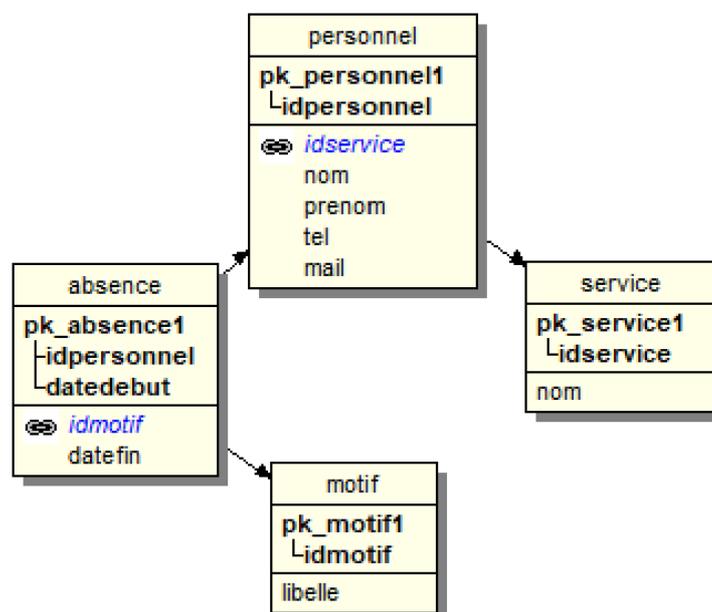
Certains outils facilitent la conception lors du développement d'une application, tel que WinDesign, qui permet de générer automatiquement le MLD ainsi que le MPD sous forme de script SQL, à partir du MCD.

1. Modèle conceptuel de données (MCD)



MCD récupéré

2. Modèle logique de données (MLD)



MLD généré par WinDesign

3. Modèle physique de données (MPD)

```
#-----  
#      TABLE : absence  
#-----  
  
CREATE TABLE IF NOT EXISTS absence  
(  
    idpersonnel INTEGER NOT NULL ,  
    datedebut DATETIME NOT NULL ,  
    idmotif INTEGER NOT NULL ,  
    datefin DATETIME NULL  
    , PRIMARY KEY (idpersonnel, datedebut)  
)  
ENGINE=InnoDB;  
#-----  
#      TABLE : motif  
#-----  
  
CREATE TABLE IF NOT EXISTS motif  
(  
    idmotif INTEGER NOT NULL AUTO_INCREMENT ,  
    libelle VARCHAR(128) NULL  
    , PRIMARY KEY (idmotif)  
)  
ENGINE=InnoDB;  
#-----  
#      TABLE : service  
#-----  
  
CREATE TABLE IF NOT EXISTS service  
(  
    idservice INTEGER NOT NULL AUTO_INCREMENT ,  
    nom VARCHAR(50) NULL  
    , PRIMARY KEY (idservice)  
)  
ENGINE=InnoDB;  
  
#-----  
#      TABLE : personnel  
#-----  
  
CREATE TABLE IF NOT EXISTS personnel  
(  
    idpersonnel INTEGER NOT NULL AUTO_INCREMENT ,  
    idservice INTEGER NOT NULL ,  
    nom VARCHAR(50) NULL ,  
    prenom VARCHAR(50) NULL ,  
    tel VARCHAR(15) NULL ,  
    mail VARCHAR(128) NULL  
    , PRIMARY KEY (idpersonnel)  
)  
ENGINE=InnoDB;  
  
#-----  
#      CREATION DES REFERENCES DE TABLE  
#-----  
  
ALTER TABLE absence  
    ADD FOREIGN KEY FK_absence_motif (idmotif)  
        REFERENCES motif (idmotif) ;  
  
ALTER TABLE absence  
    ADD FOREIGN KEY FK_absence_personnel (idpersonnel)  
        REFERENCES personnel (idpersonnel) ;  
  
ALTER TABLE personnel  
    ADD FOREIGN KEY FK_personnel_service (idservice)  
        REFERENCES service (idservice) ;
```

Script SQL généré par WinDesign

4. Création de la base de données (BDD)

Le script SQL nous permet de créer la base de données correspondant au schéma conceptuel de données du projet MediaTek86. Les étapes de création sont les suivantes :

- Lancement de WAMPServer
- Ouverture de phpMyAdmin, connexion avec nos identifiants
- Création de la base de donnée dans l'onglet dédié à cet effet :

nom de la BDD : "mediatek86", encodage : "utf_unicode_ci"

- Exécution du script SQL dans l'onglet "SQL" de la base de données

5. Création d'un utilisateur ayant les droits d'accès

L'interface graphique (1) phpMyAdmin offre de nombreuses possibilités, la création d'un utilisateur, l'attribution ou la révocation des privilèges, l'ajout de tables, la modification du type des champs, la suppression etc... L'exécution des scripts SQL (2) permet également d'effectuer ces actions à condition de connaître le langage, cela permet de gagner davantage de temps sur de nombreuses manipulations.

The image shows two screenshots of the phpMyAdmin interface. The top screenshot, labeled '1', is the 'Ajouter un compte d'utilisateur' (Add user account) page. It features a navigation bar with options like 'Bases de données', 'SQL', 'État', 'Comptes utilisateurs', 'Exporter', 'Importer', 'Paramètres', 'Réplication', and 'Plus'. The main form is divided into two sections: 'Informations pour la connexion' and 'Base de données pour ce compte d'utilisateur'. In the first section, the 'Nom d'utilisateur' is 'mediatekadmin', 'Nom d'hôte' is 'localhost', and the password strength is 'Fort'. The second section has three checkboxes: 'Créer une base portant son nom...', 'Accorder tous les privilèges à un nom passe-partout...', and 'Donner tous les privilèges sur la base de données mediatek86.', with the last one checked. The bottom screenshot, labeled '2', shows the 'Exécuter une ou des requêtes SQL' page for the 'mediatek86' database. The SQL query entered is:

```
1 CREATE USER 'mediatekadmin'@'localhost' IDENTIFIED BY '8Medi@Tek6$m';
2 GRANT USAGE ON *.* TO 'mediatekadmin'@'localhost';
3 GRANT ALL PRIVILEGES ON `mediatek86`.* TO 'mediatekadmin'@'localhost';
```

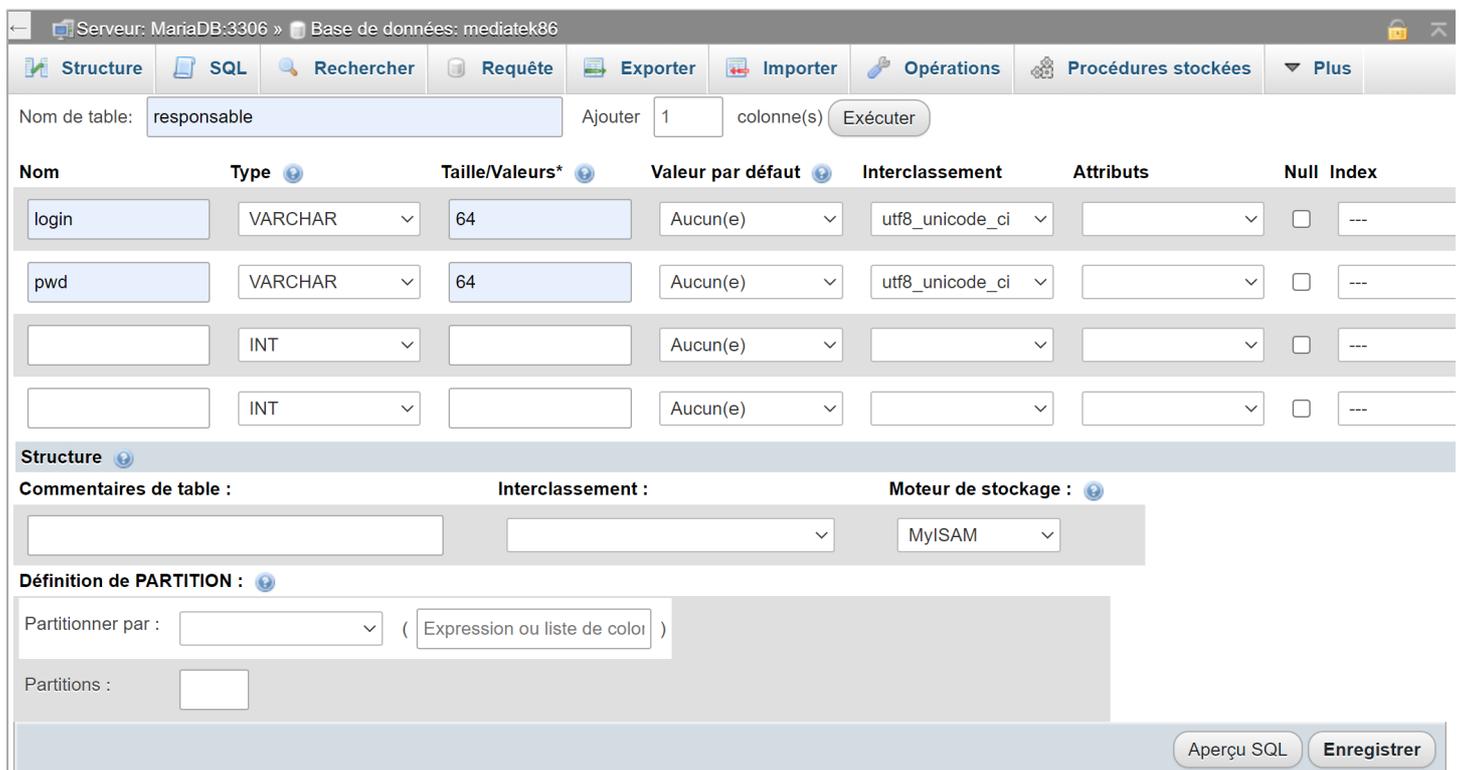
 The interface includes buttons for 'Effacer', 'Format', and 'Récupérer la requête auto-sauvegardée', and an 'Exécuter' button at the bottom right.

Pour vérifier que tout s'est bien passé, nous nous déconnecterons et tenterons de nous connecter avec les nom d'utilisateur et mot de passe nouvellement créés.



The image shows the phpMyAdmin welcome screen. At the top, there is a logo for phpMyAdmin featuring a sailboat. Below the logo, the text "Bienvenue dans phpMyAdmin" is displayed. There are two main sections: "Langue - Language" with a dropdown menu set to "Français - French", and "Connexion" with input fields for "Utilisateur :" (containing "mediatekadmin") and "Mot de passe :". An "Exécuter" button is located at the bottom right of the connection section.

6. Ajouter une table 'responsable' (login/pwd)



The image shows the phpMyAdmin interface for creating a new table named "responsable". The table name is entered in the "Nom de table:" field. Below this, there are options to "Ajouter 1 colonne(s)" and an "Exécuter" button. A table structure is shown with columns: "login" (VARCHAR, 64), "pwd" (VARCHAR, 64), and two empty rows (INT, 0). Below the table structure, there are sections for "Structure", "Commentaires de table", "Interclassement", "Moteur de stockage" (set to MyISAM), and "Définition de PARTITION". At the bottom right, there are buttons for "Aperçu SQL" and "Enregistrer".

Nom	Type	Taille/Valeurs*	Valeur par défaut	Interclassement	Attributs	Null	Index
login	VARCHAR	64	Aucun(e)	utf8_unicode_ci		<input type="checkbox"/>	---
pwd	VARCHAR	64	Aucun(e)	utf8_unicode_ci		<input type="checkbox"/>	---
	INT		Aucun(e)			<input type="checkbox"/>	---
	INT		Aucun(e)			<input type="checkbox"/>	---

7. Alimenter la base de donnée

Les tables 'personnel' et 'absence' seront remplis d'exemples aléatoires générés par Generatedata.com. La démarche est simple, après s'être rendu sur le site (<https://www.generatedata.com>), il faut entrer un certain nombre d'informations tel que : le nom de la colonne correspondant au nom du champ de notre table, son type, spécifier le format de sortie, ici, SQL de type MySQL, puis cliquer sur Générer.

Generate About News Donate

Les personnels SAVE

COUNTRY-SPECIFIC DATA
France

DATA SET

Order	Table Column	Data Type	Examples	Options	Help	Del
1	idpersonnel	Number Range	No examples available.	Between 1 and 10	?	
2	idservice	Number Range	No examples available.	Between 1 and 3	?	
3	nom	Names	Smith (surname)	Surname	?	
4	prenom	Names	Alex (any gender)	Name	?	
5	tel	Phone / Fax	France	0X xx xx xx xx	?	
6	mail	Email	No examples available.	No options available.	?	

Add 1 Row(s)

EXPORT TYPES
CSV Excel HTML JSON LDIF Programming Language SQL XML - hide data format options

Database table name: personnel
Database Type: MySQL
Misc Options:
 Include CREATE TABLE query
 Include DROP TABLE query
 Enclose table and field names with backquotes

Statement Type: INSERT INSERT IGNORE UPDATE
INSERT batch size: 10
Primary Key: None Add default auto-increment column

Generate 100 rows
 Generate in-page New window/tab Prompt to download Zip? **Generate**

Generate About News Donate

Generated 100 of 100 results cancel

```
1 INSERT INTO `personnel` (`idpersonnel`,`idservice`,`nom`,`prenom`,`tel`,`mail`) VALUES (9,1,"Bauer","Bevis","05 79 72 53 71","faucibus@felisorciadipiscing.net"),(2,3,
2 INSERT INTO `personnel` (`idpersonnel`,`idservice`,`nom`,`prenom`,`tel`,`mail`) VALUES (5,1,"Lancaster","Brynn","06 55 26 90 84","commodo@sem.edu"),(4,2,"Johnson",
3 INSERT INTO `personnel` (`idpersonnel`,`idservice`,`nom`,`prenom`,`tel`,`mail`) VALUES (6,3,"Gray","Paki","02 92 15 97 94","eget.mollis.lectus@pedenecante.edu"),(2,2,
4 INSERT INTO `personnel` (`idpersonnel`,`idservice`,`nom`,`prenom`,`tel`,`mail`) VALUES (10,1,"Banks","Ali","09 35 71 73 34","lorem.vitae@Pellentesquetincidunt.com"),(
5 INSERT INTO `personnel` (`idpersonnel`,`idservice`,`nom`,`prenom`,`tel`,`mail`) VALUES (9,2,"Carney","Knox","04 42 73 42 97","euismod.in@ornaregestasligula.org"),(10
6 INSERT INTO `personnel` (`idpersonnel`,`idservice`,`nom`,`prenom`,`tel`,`mail`) VALUES (4,1,"York","Declan","01 76 56 56 41","feugiat.nec@orciadipiscingnon.ca"),(5,1,
7 INSERT INTO `personnel` (`idpersonnel`,`idservice`,`nom`,`prenom`,`tel`,`mail`) VALUES (7,3,"Olson","Rooney","05 55 12 86 18","in@sed.ca"),(1,3,"Cross","Joelynn","05
8 INSERT INTO `personnel` (`idpersonnel`,`idservice`,`nom`,`prenom`,`tel`,`mail`) VALUES (1,3,"Meadows","Linda","01 89 25 42 44","Curabitur@ulputatemaaurissagittis.org"
9 INSERT INTO `personnel` (`idpersonnel`,`idservice`,`nom`,`prenom`,`tel`,`mail`) VALUES (9,3,"McMahon","Denton","03 99 32 45 66","ullamcorper.Duis.at@rutrum.ca"),(2,
10 INSERT INTO `personnel` (`idpersonnel`,`idservice`,`nom`,`prenom`,`tel`,`mail`) VALUES (10,1,"Mayo","Lila","08 84 53 60 19","eros@molestie.net"),(8,2,"Alford","Neve"),(2,
11
```

La version gratuite génère un total de 10x10 inserts, nous copierons la première ligne générée et l'exécuterons dans la partie SQL de notre base de donnée pour ajouter une dizaine de personnels. Même démarche pour ajouter une cinquantaine d'absence dans la table absence, nous copierons 5 lignes parmi celles générées et exécuterons le script SQL.

Generate About News Donate

Les absences SAVE   

COUNTRY-SPECIFIC DATA 

France 

DATA SET 

Order	Table Column	Data Type	Examples	Options	Help	Del
1	idpersonnel	Number Range	No examples available.	Between 1 and 10	?	<input type="checkbox"/>
2	datedebut	Date	MySQL datetime	From: 06/01/2020 To: 06/01/2022 Format code: Y-m-d	?	<input type="checkbox"/>
3	idmotif	Number Range	No examples available.	Between 1 and 3	?	<input type="checkbox"/>
4	datefin	Date	MySQL datetime	From: 06/01/2020 To: 06/01/2022 Format code: Y-m-d	?	<input type="checkbox"/>

Order Table Column Data Type Examples Options Help Del

Add 1 Row(s)

EXPORT TYPES 

CSV Excel HTML JSON LDIF Programming Language SQL XML - hide data format options

Database table name: absence Statement Type: INSERT INSERT IGNORE UPDATE

Database Type: MySQL INSERT batch size: 10

Misc Options: Include CREATE TABLE query Include DROP TABLE query Enclose table and field names with backquotes

Primary Key: None Add default auto-increment column

Generate 100 rows Generate in-page New window/tab Prompt to download Zip? Generate

Generate About News Donate

Generated 100 of 100 results cancel 

```

1 INSERT INTO `absence` (`idpersonnel`,`datedebut`,`idmotif`,`datefin`) VALUES (4,"2021-11-05",1,"2021-11-22"),(6,"2021-09-24"
2 INSERT INTO `absence` (`idpersonnel`,`datedebut`,`idmotif`,`datefin`) VALUES (9,"2021-04-25",1,"2020-11-28"),(3,"2022-01-15"
3 INSERT INTO `absence` (`idpersonnel`,`datedebut`,`idmotif`,`datefin`) VALUES (7,"2021-09-01",2,"2021-11-10"),(5,"2020-11-18"
4 INSERT INTO `absence` (`idpersonnel`,`datedebut`,`idmotif`,`datefin`) VALUES (9,"2021-07-18",2,"2022-03-12"),(5,"2021-11-13"
5 INSERT INTO `absence` (`idpersonnel`,`datedebut`,`idmotif`,`datefin`) VALUES (1,"2020-10-31",2,"2021-03-24"),(7,"2021-02-11"
6 INSERT INTO `absence` (`idpersonnel`,`datedebut`,`idmotif`,`datefin`) VALUES (6,"2020-06-07",3,"2022-05-28"),(3,"2021-04-01"
7 INSERT INTO `absence` (`idpersonnel`,`datedebut`,`idmotif`,`datefin`) VALUES (10,"2021-05-12",2,"2020-07-09"),(4,"2021-07-23"
8 INSERT INTO `absence` (`idpersonnel`,`datedebut`,`idmotif`,`datefin`) VALUES (10,"2022-01-19",2,"2021-09-23"),(1,"2021-08-17"
9 INSERT INTO `absence` (`idpersonnel`,`datedebut`,`idmotif`,`datefin`) VALUES (1,"2021-11-29",2,"2022-05-05"),(3,"2022-03-06"
10 INSERT INTO `absence` (`idpersonnel`,`datedebut`,`idmotif`,`datefin`) VALUES (2,"2022-04-28",3,"2021-05-11"),(6,"2021-04-27"
11

```

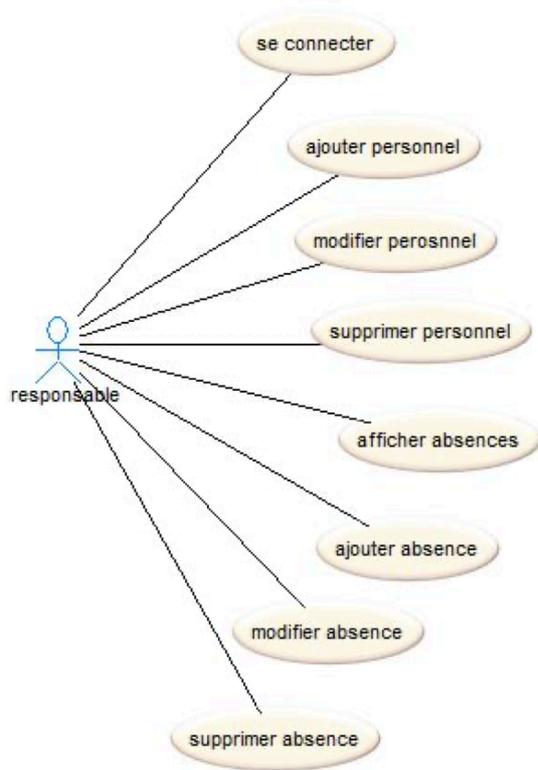
Le problème que peut poser la génération aléatoire des dates est de générer des dates qui soient incohérentes : datefin inférieure à datedebut. Pour résoudre ce problème : j'ai exporté les données relatives aux absences au format CSV, importé ces données dans un tableur, j'ai exécuté un test if pour comparer les dates, effectué des modifications (glissement des colonnes), exporté ces nouvelles données pour les importer de nouveau dans la base de données après avoir supprimé les anciennes données.

Les tables "responsable", "motif", "service" seront alimentés sans utiliser le site de génération automatique de données.

```
# -----  
#      TABLE : responsable  
# -----  
INSERT INTO responsable (login, pwd)  
VALUES ('responsable', SHA2('RespMedi@2021',256));  
  
# -----  
#      TABLE : motif  
# -----  
INSERT INTO motif (libelle)  
VALUES ('vacances'), ('maladie'), ('motif familial'), ('congé parental');  
  
# -----  
#      TABLE : service  
# -----  
INSERT INTO service (nom)  
VALUES ('administratif'), ('médiation culturelle'), ('prêt');
```

ÉTAPE 2

Dessiner les interfaces
Structurer l'application en MVC
Créer un dépôt, coder le visuel



Cas d'utilisation n°1	Se connecter
Acteur	Responsable du personnel
Événement déclencheur	aucun
Intérêt	Accéder aux fonctionnalités de gestion du personnel
Précondition	aucune
Scénario nominal	<ol style="list-style-type: none"> 1. Le système affiche les zones de saisie du login et du mot de passe. 2. Le responsable saisit son login, son mot de passe et valide. 3. Le système affiche la liste des personnels et des boutons de commande pour pouvoir ajouter un nouveau personnel, supprimer un personnel sélectionné dans la liste, modifier un personnel sélectionné dans la liste, accéder à la gestion des absences d'un personnel sélectionné dans la liste.
Scénario alternatif	<ol style="list-style-type: none"> 3a. Les champs ne sont pas tous remplis : retour au point 2. 3b. Le login et/ou le mot de passe ne correspondent pas à ceux enregistrés dans la base de données, le système affiche une alerte : retour au point 2.

Cas d'utilisation n°2	Ajouter un personnel
Acteur	Responsable du personnel
Événement déclencheur	aucun
Intérêt	Ajout d'un nouveau personnel dans la base de données
Précondition	Être connecté
Scénario nominal	<ol style="list-style-type: none"> 1. Le responsable demande l'ajout d'un personnel. 2. Le système affiche les zones de saisie pour le personnel (nom, prénom, tel, mail) et une liste déroulante pour sélectionner un service d'affectation. 3. Le responsable saisit les informations, sélectionne un service d'affectation et demande d'enregistrer le nouveau personnel. 4. Le système enregistre le nouveau personnel dans la base de données et met à jour l'affichage de la liste des personnels.
Scénario alternatif	<ol style="list-style-type: none"> 3a. Le responsable annule sa demande d'ajout : fin du traitement. 4a. Tous les champs ne sont pas remplis, le système affiche une alerte : retour au point 3.

Afin de nous accompagner dans la suite du projet, les équipes de développement ont constitué un dossier documentaire contenant un diagramme d'utilisation suivi du descriptif de chaque cas d'utilisation. Ces informations nous permettront d'élaborer une première ébauche des interfaces de l'application.

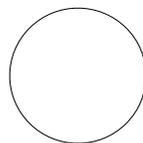
Nous utiliserons l'outil de maquettage Pencil pour réaliser notre prototype GUI (graphical user interface). Les interfaces graphiques seront créés avec Windows Forms, sous Visual Studio Code.

1. Prototype GUI

a. Charte graphique



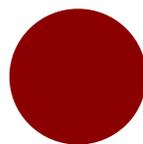
Dark Slate Blue
HEX #483D8B
RGB (72, 61, 139)



White
HEX #FFFFFF
RGB (255, 255, 255)



Indigo
HEX #4B0082
RGB (75, 0, 130)



Dark Red
HEX #8B0000
RGB (139, 0, 0)



Dim Gray
HEX #696969
RGB (105, 105, 105)

Yu Gothic *(Police d'écriture)*

b. Première ébauche des interfaces

Mon idée de départ était de partir sur la création de 3 fenêtres :

- (1) fenêtre d'authentification "Login" répondant au cas d'utilisation n°1
- (2) fenêtre "Gestion du personnel" répondant aux cas d'utilisation n°2, 3, 4
- (3) fenêtre "Gestion des absences" répondant aux cas d'utilisation n°5, 6, 7, 8.

Selon les scénarios, j'ai également ajouté les 'MessageBox' contenant les messages d'information.

Les fenêtres (2) et (3) ont de nombreuses similarités car elles offrent la possibilité de réaliser les mêmes actions : affichage d'une liste, ajouter, modifier et supprimer un élément. Cependant certains éléments sont présents dans une fenêtre et pas dans une autre, ce qui m'a poussé à vouloir créer 3 fenêtres par simplicité. C'est en codant que j'ai réalisé que ça n'était pas la chose la plus facile à gérer et qu'elle ne donnait pas non plus la meilleure expérience utilisateur. Les modifications ont été apportées par la suite directement dans Visual Studio.

Le dossier complet Pencil sera disponible en PDF dans le dépôt distant.

1



MediaTek86

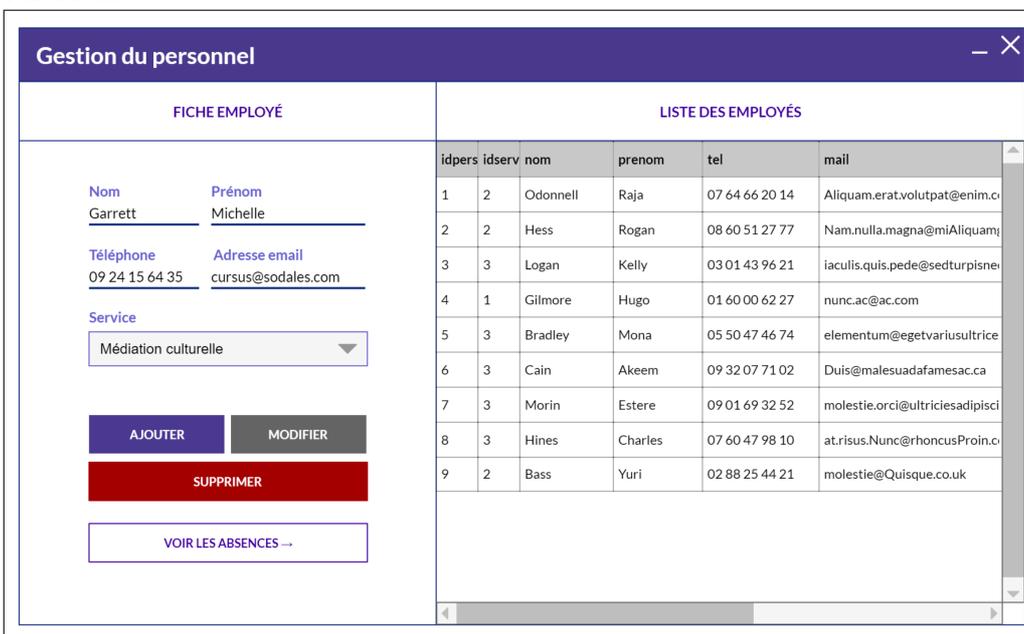
Nom d'utilisateur

Mot de passe

Afficher le mot de passe

Se connecter

2



Gestion du personnel

FICHE EMPLOYÉ

Nom: Garrett Prénom: Michelle

Téléphone: 09 24 15 64 35 Adresse email: cursus@sodales.com

Service: Médiation culturelle

AJOUTER **MODIFIER**

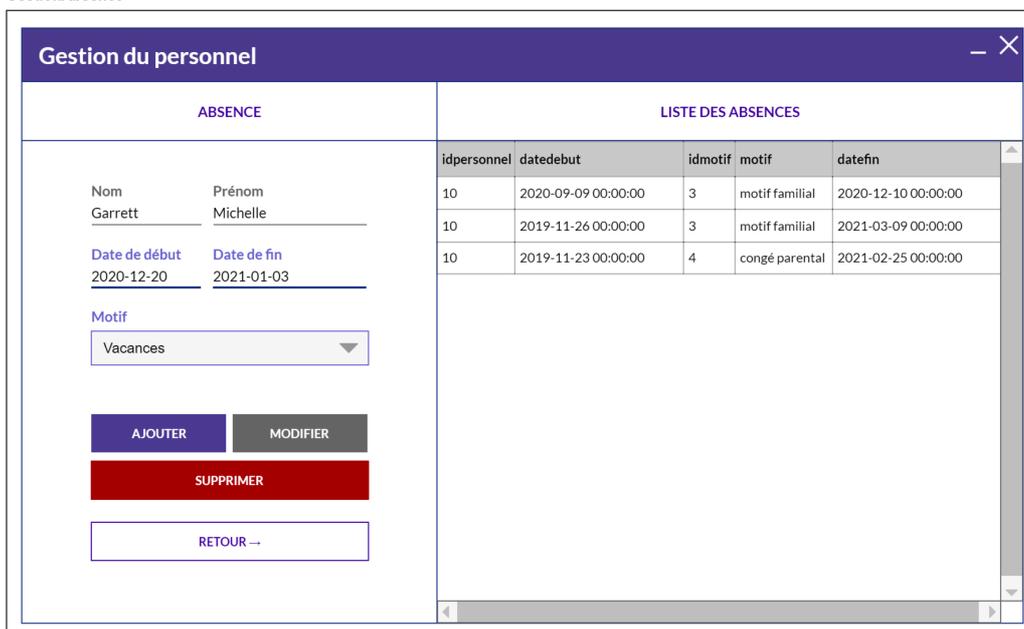
SUPPRIMER

[VOIR LES ABSENCES →](#)

LISTE DES EMPLOYÉS

idpers	idserv	nom	prenom	tel	mail
1	2	Odonnell	Raja	07 64 66 20 14	Aliquam.erat.volutpat@enim.c
2	2	Hess	Rogan	08 60 51 27 77	Nam.nulla.magna@miAliquam
3	3	Logan	Kelly	03 01 43 96 21	iaculis.quis.pede@sedturpisnei
4	1	Gilmore	Hugo	01 60 00 62 27	nunc.ac@ac.com
5	3	Bradley	Mona	05 50 47 46 74	elementum@egetvariusultrice
6	3	Cain	Akeem	09 32 07 71 02	Duis@malesuadafamesac.ca
7	3	Morin	Estere	09 01 69 32 52	molestie.orci@ultricesadipisci
8	3	Hines	Charles	07 60 47 98 10	at.risus.Nunc@rhoncusProin.c
9	2	Bass	Yuri	02 88 25 44 21	molestie@Quisque.co.uk

3



Gestion du personnel

ABSENCE

Nom: Garrett Prénom: Michelle

Date de début: 2020-12-20 Date de fin: 2021-01-03

Motif: Vacances

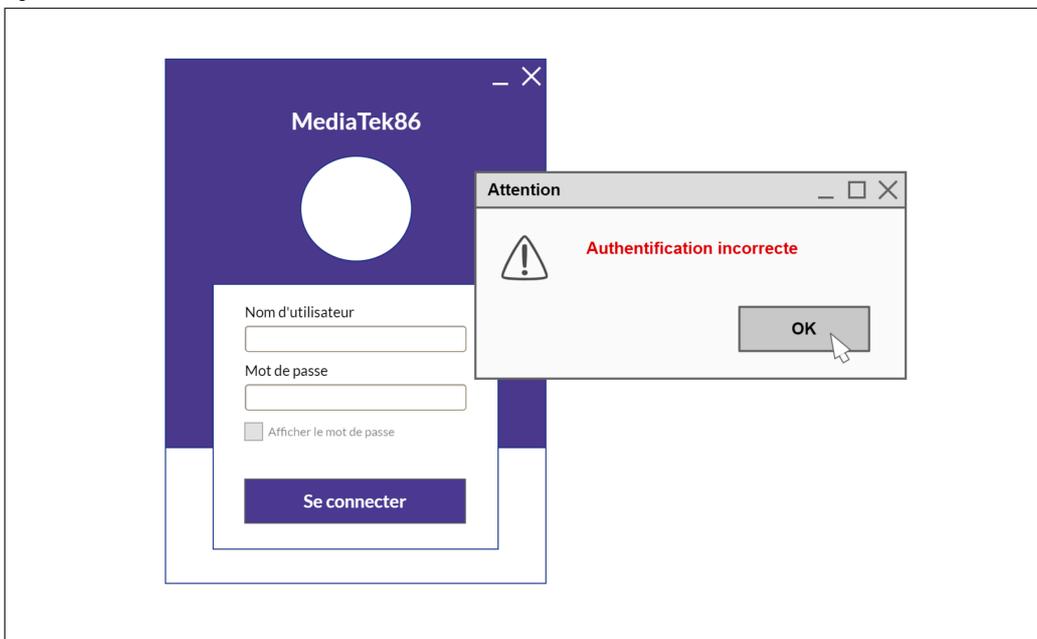
AJOUTER **MODIFIER**

SUPPRIMER

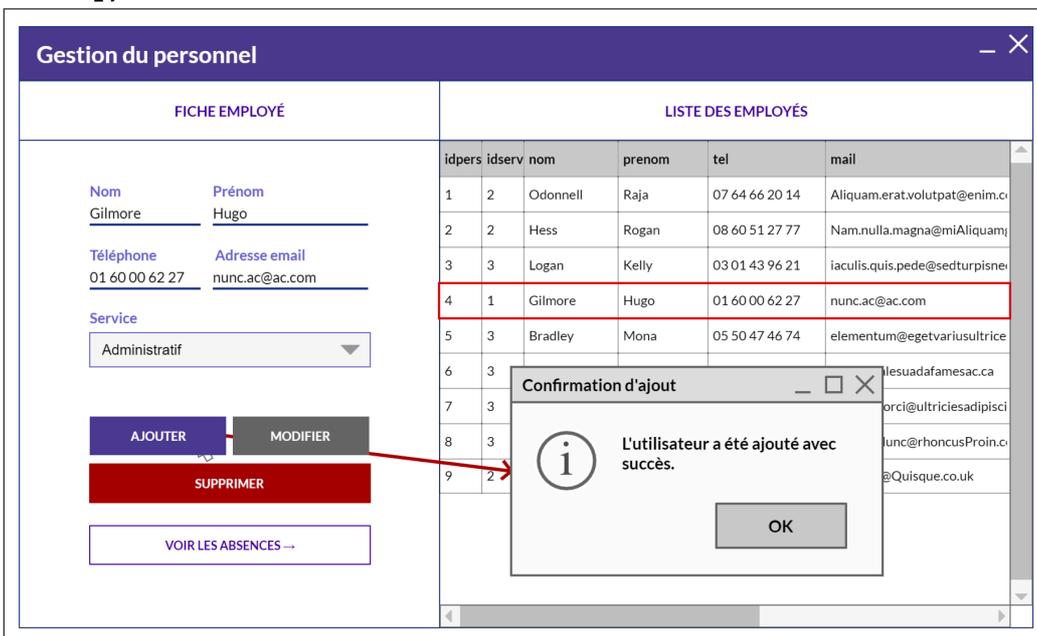
[RETOUR →](#)

LISTE DES ABSENCES

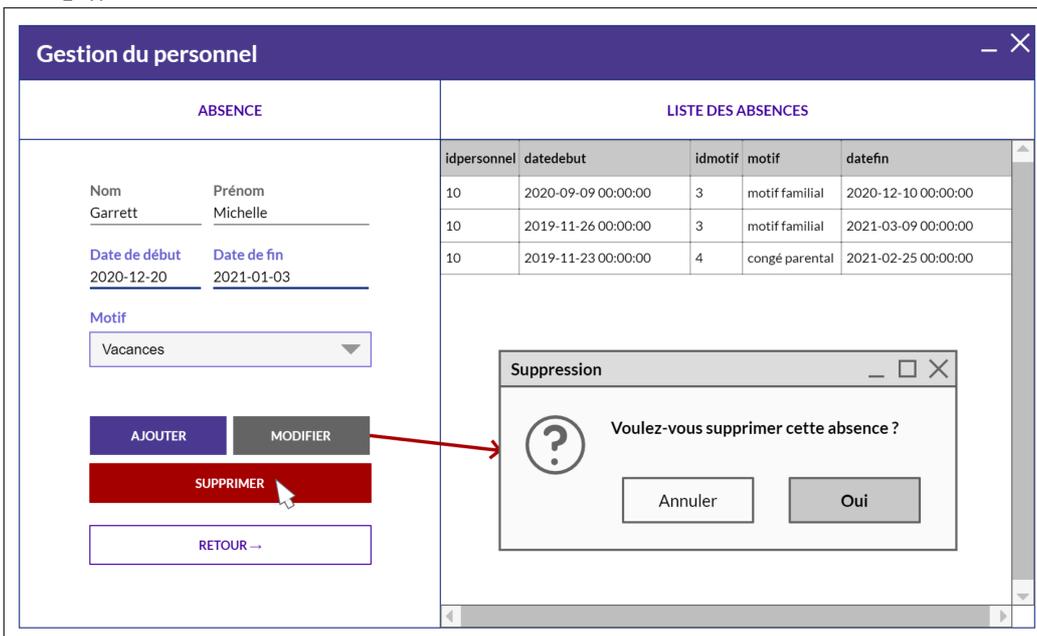
idpersonnel	datedebut	idmotif	motif	datefin
10	2020-09-09 00:00:00	3	motif familial	2020-12-10 00:00:00
10	2019-11-26 00:00:00	3	motif familial	2021-03-09 00:00:00
10	2019-11-23 00:00:00	4	congé parental	2021-02-25 00:00:00



confirmation_ajout



absence_suppression



2. Création du dépôt distant

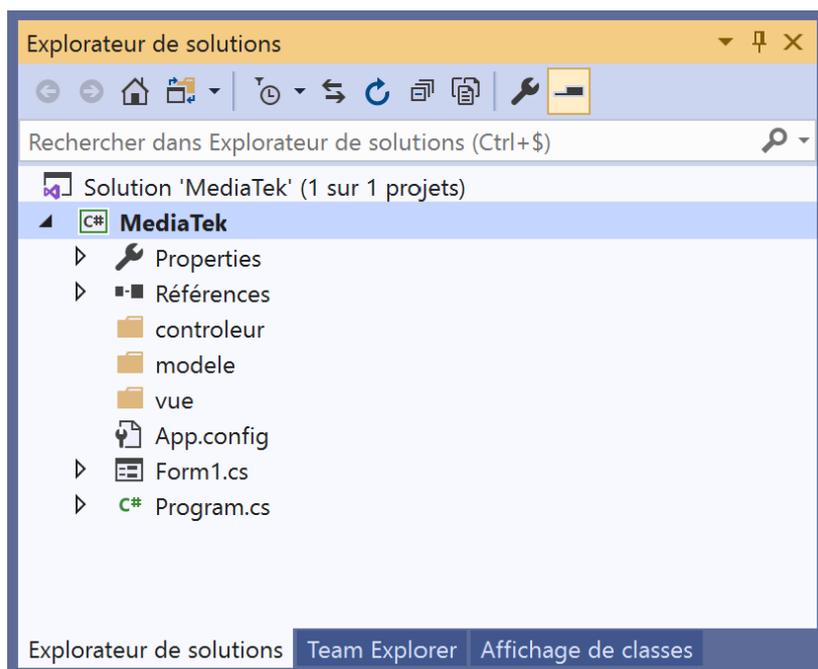
Après m'être connectée sur mon compte Github (<https://github.com>), les étapes de création du dépôt sont les suivantes :

- Create a new repository
- Code > Clone > Copier l'adresse du dépôt
- Dans Visual Studio > Cloner un projet > Coller l'adresse

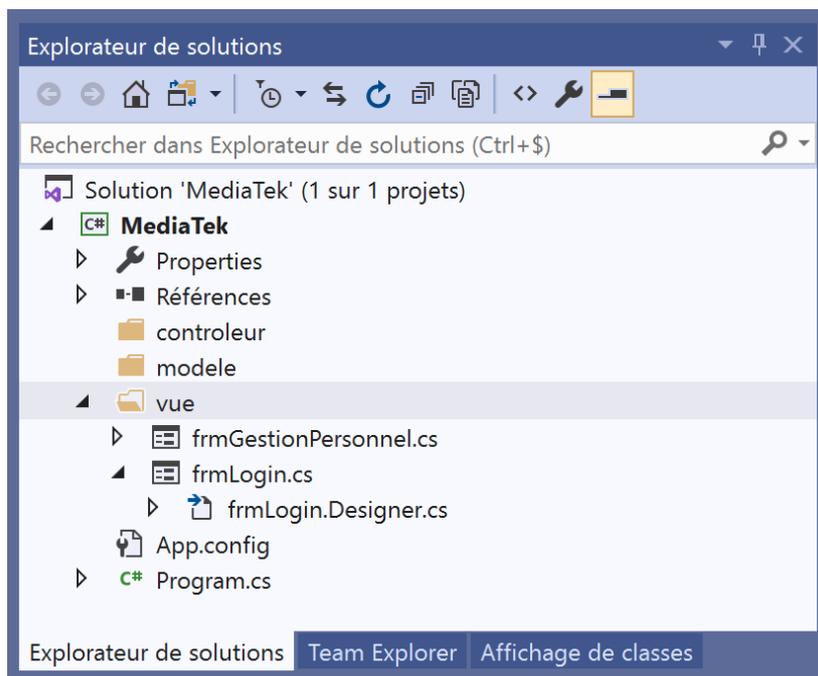
Le dépôt distant est créé, il est maintenant possible de sauvegarder le travail et de consulter tous les changements apportés sur le Github du repository.

3. Structurer l'application suivant le modèle MVC

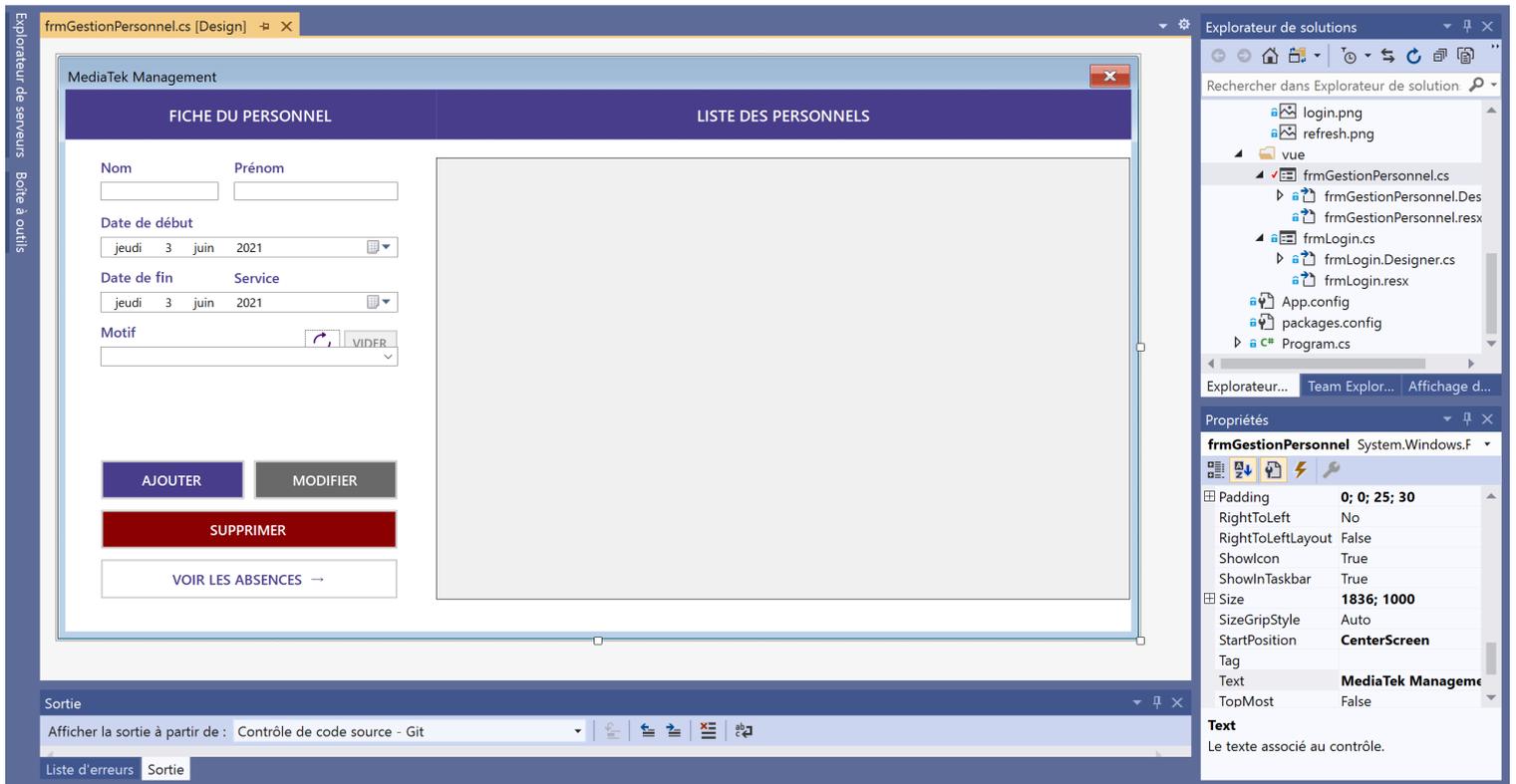
L'avantage de structurer notre application suivant le modèle MVC (Model-View-Controller ou en français Modèle-Vue-Contrôleur) nous permet d'être plus efficient et offre une meilleure maintenance du projet. Par exemple, un développeur back-end peut mettre en place la structure sans intervenir sur le front-end et vice-versa. Cela s'explique par le fait que seul le Contrôleur gère les interactions entre le Modèle et la Vue, les autres composants ne communiquent pas entre eux, le Contrôleur fait office d'intermédiaire.



4. Coder la partie Vue de l'application



Les forms seront créés dans le package Vue. Avec la maquette comme support, 3 forms devront être créés correspondant aux fenêtres du point 1.b, cependant cette idée ne sera pas retenue car elle aura pour effet de créer des redondances dans le code au niveau des boutons, de plus il sera plus compliqué de gérer les données si elles proviennent de forms différents. Un autre point qui m'a convaincu de changer de stratégie est l'expérience utilisateur : l'ouverture et la fermeture des fenêtres lorsqu'il faut passer de la Gestion du personnel à la Gestion des absences et vice-versa provoque un sentiment désagréable, il faut donner l'illusion que les changements s'opèrent dynamiquement, dans un même endroit, sans efforts. La fenêtre "Gestion des absences" étant un duplicata de la fenêtre "Gestion du personnel", j'ai fait le choix de réunir les composants graphiques sur une même fenêtre, et de créer une variable de type booléen qui permettra de déterminer l'affichage.



L'application démarrera sur la fenêtre d'authentification, une fois l'utilisateur connecté, une nouvelle fenêtre s'ouvre et affiche la liste des personnels ainsi que les boutons permettant la gestion du personnel, les composants pour la gestion des absences sont masqués. Lorsque l'utilisateur clique sur "Voir les absences", les composants permettant la gestion du personnel seront à leur tour masqués et ceux utiles à la gestion des absences seront affichés.

```

73     /// <summary>
74     /// Affiche/Cache les composants graphiques
75     /// </summary>
76     /// <param name="show"></param>
77     public void ShowComponents(bool show)
78     {
79         // GESTION DU PERSONNEL
80         txtNom.Enabled = !show;
81         txtPrenom.Enabled = !show;
82         lblService.Visible = !show;
83         cboServices.Visible = !show;
84         dgvPersonnels.Visible = !show;
85
86         // GESTION DES ABSENCES
87         dgvAbsences.Visible = show;
88         lblDateDebut.Visible = show;
89         lblDateFin.Visible = show;
90         DPickDebut.Visible = show;
91         DPickFin.Visible = show;
92         lblMotif.Visible = show;
93         cboMotifs.Visible = show;
94
95         if (!show)
96         {
97             // GESTION DU PERSONNEL
98             picRefresh.Location = new Point(205, 214);
99             btnVider.Location = new Point(240, 214);
100            lblNom.ForeColor = System.Drawing.Color.Indigo;
101            lblPrenom.ForeColor = System.Drawing.Color.Indigo;
102            btnSwitch.Text = "VOIR LES ABSENCES ->";
103            lblGauche.Text = "FICHE DU PERSONNEL";
104            lblDroit.Text = "LISTE DES PERSONNELS";
105        }
106        else
107        {
108            // GESTION DES ABSENCES
109            picRefresh.Location = new Point(205, 263);
110            btnVider.Location = new Point(240, 263);
111            lblNom.ForeColor = System.Drawing.Color.DimGray;
112            lblPrenom.ForeColor = System.Drawing.Color.DimGray;
113            btnSwitch.Text = "RETOUR ->";
114            lblGauche.Text = "DÉTAIL DE L'ABSENCE";
115            lblDroit.Text = "LISTE DES ABSENCES";
116        }
117    }

```

5. Sauvegarde sur le dépôt distant

Les sauvegardes peuvent être réalisés directement dans VS Code dans l'onglet Team Explorer après avoir installé l'extension Github pour Visual Studio (<https://visualstudio.github.com>). Cependant j'ai fait le choix par préférence, d'effectuer les sauvegardes à l'aide du Git Bash et de la commande 'git'.

(1) `git log` permet de lister les commits

```
MINGW64:/c/Users/Monica Tevy Sen/Desktop/MediaTek/MediaTek
Monica Tevy Sen@DESKTOP-960AOHM MINGW64 ~/Desktop/MediaTek/MediaTek (master)
$ git log
commit 7b10585e96b1ee1ff00813f6774bef7e0325b82a (HEAD -> master, origin/master)
Author: monicatevy <monicatevy.sen@pm.me>
Date: Tue Jun 1 03:49:52 2021 +0700

    (Authentification) L'utilisateur doit se connecter pour accéder à la fenêtre principale

commit eead78c2844514fd6ecc53b43f84c3316d0e378d
Author: monicatevy <monicatevy.sen@pm.me>
Date: Mon May 31 22:38:48 2021 +0700

    (documentation technique) Génération du fichier XML

commit 8da606f60581797ec685c2c78acc882e8f87a685
Author: monicatevy <monicatevy.sen@pm.me>
Date: Mon May 31 15:26:53 2021 +0700

    (dal, connexion, modele, vue) Création des méthodes nécessaires

commit cfc21ef86bdb5d0f42aa631c4ba7e55f9e4370cf
Author: monicatevy <monicatevy.sen@pm.me>
Date: Mon May 24 16:38:17 2021 +0700
```

(2) `git status` affiche les fichiers modifiés

(3) `git add .` permet d'ajouter tous les fichiers dans le répertoire local

```
MINGW64:/c/Users/Monica Tevy Sen/Desktop/MediaTek/MediaTek
Monica Tevy Sen@DESKTOP-960AOHM MINGW64 ~/Desktop/MediaTek/MediaTek (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .vs/MediaTek/v16/.suo
        modified:   MediaTek/bin/Debug/MediaTek.exe
        modified:   MediaTek/bin/Debug/MediaTek.pdb
        modified:   MediaTek/connexion/ConnexionBDD.cs
        modified:   MediaTek/obj/Debug/MediaTek.csprojAssemblyReference.cache
        modified:   MediaTek/obj/Debug/MediaTek.exe
        modified:   MediaTek/obj/Debug/MediaTek.pdb
        modified:   MediaTek/vue/frmGestionPersonnel.cs

no changes added to commit (use "git add" and/or "git commit -a")
Monica Tevy Sen@DESKTOP-960AOHM MINGW64 ~/Desktop/MediaTek/MediaTek (master)
$ git add .
```

- (4) `git commit -m` valide les modifications avec un message entre guillemets
- (5) `git push` envoie les modifications locales à la branche principale

```
MINGW64:/c:/Users/Monica Tevy Sen/Desktop/MediaTek/MediaTek

Monica Tevy Sen@DESKTOP-960A0HM MINGW64 ~/Desktop/MediaTek/MediaTek (master)
$ git commit -m "(connexionBDD) Les messages d'erreur sont masqués"
[master 6aea3a5] (connexionBDD) Les messages d'erreur sont masqués
 8 files changed, 8 insertions(+), 8 deletions(-)
 rewrite .vs/MediaTek/v16/.suo (80%)

Monica Tevy Sen@DESKTOP-960A0HM MINGW64 ~/Desktop/MediaTek/MediaTek (master)
$ git push
Enumerating objects: 35, done.
Counting objects: 100% (35/35), done.
Delta compression using up to 8 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (18/18), 8.78 KiB | 1.10 MiB/s, done.
Total 18 (delta 10), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (10/10), completed with 10 local objects.
To https://github.com/monicatevy/MediaTek.git
 7b10585..6aea3a5 master -> master
```

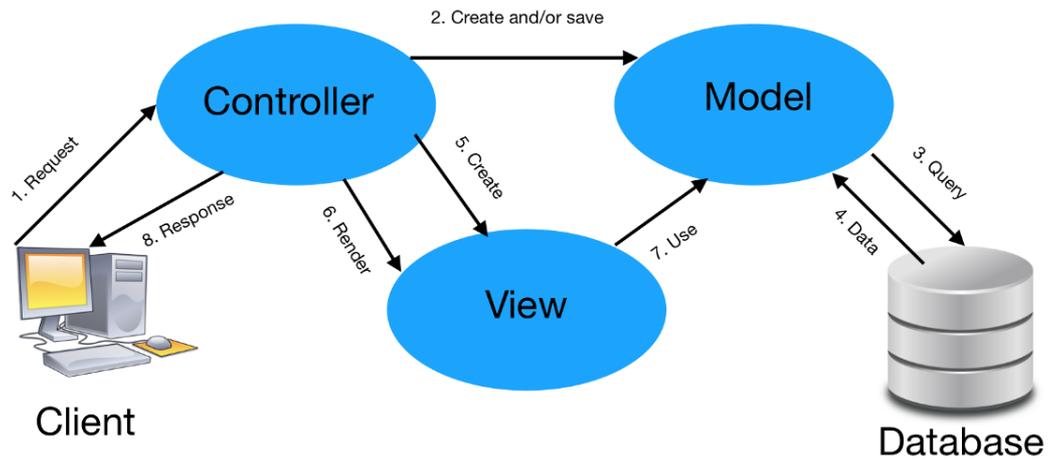
Ce commit pourra être consulté sur le Github du repository.

The screenshot shows the GitHub interface for the repository 'MediaTek'. At the top, there are navigation tabs: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation, there are buttons for 'Go to file', 'Add file', and 'Code'. The main content area shows a list of commits. The most recent commit is by 'monicatevy' with the message '(connexionBDD) Les messages d'erreur sont masqués', commit hash '6aea3a5', made 1 minute ago, and has 7 commits in the history. Below this, there is a list of files and folders:

File/Folder	Commit Message	Time Ago
.vs/MediaTek/v16	(connexionBDD) Les messages d'erreur sont masqués	1 minute ago
MediaTek	(connexionBDD) Les messages d'erreur sont masqués	1 minute ago
packages	(dal, connexion, modele, vue) Création des méthodes nécessaires	3 days ago
MediaTek.sln	Initial commit	10 days ago

ÉTAPE 3

Coder le modèle et les outils de connexion
Générer la documentation technique



1. Configurer l'IDE pour accéder à la BDD

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Explorateur de serveurs' pane shows the 'localhost(mediatek86)' server. In the center, a 'Modifier la connexion' dialog box is open, showing the configuration for a new connection:

- Source de données: .NET Framework Data Provider for MySQL
- Server name: localhost
- User name: mediatekadmin
- Password: [masked]
- Save my password:
- Database name: mediatek86

On the right, the 'Propriétés' pane shows the connection string for 'localhost(mediatek86) Connexion':

```
Chaîne de connexion server=localhost;user id=mediatekadmin;persistsecurityinfo=True;database=mediatek86
État Ouvrir
Fournisseur .NET Framework Data Provider for MySQL
Type MySQL
Version 5.5.5-10.4.13-MariaDB
```

On ajoute une nouvelle connexion avec les informations de notre BDD.

2. Création du package connexion

Ce package contiendra la classe technique singleton de connexion à la base de données avec toutes les méthodes nécessaires.

```
1 using MySql.Data.MySqlClient;
2 using System;
3 using System.Collections.Generic;
4 using System.Windows.Forms;
5
6 namespace MediaTek.connexion
7 {
8     /// <summary>
9     /// Connexion à la base de données et exécution des requêtes
10    /// </summary>
11    public class ConnexionBDD
12    {
13        /// <summary>
14        /// Objet de connexion à la BDD à partir d'une chaîne de connexion
15        /// </summary>
16        private MySqlConnection connection;
17
18        /// <summary>
19        /// Constructeur privé pour créer la connexion à la BDD et l'ouvrir
20        /// </summary>
21        /// <param name="stringConnect">chaîne de connexion</param>
22        private ConnexionBDD(string stringConnect)
23        {
24            try
25            {
26                connection = new MySqlConnection(stringConnect);
27                connection.Open();
28            }
29            catch (Exception e)
30            {
31                MessageBox.Show(e.Message, "Erreur");
32                Application.Exit();
33            }
34        }
35    }
```

L'utilisation d'un singleton est adaptée pour la connexion à une base de donnée car le singleton ne peut être instancié qu'une seule fois, ce qui signifie que si l'instance de connexion n'est pas déjà créé, elle sera créé, sinon, l'instance sera récupérée avec la méthode GetInstance, son utilisation constitue donc un gain de temps.

Pour se connecter à notre base de données MySQL, il faut au préalable ajouter les "références", autrement dit les libraries qui vont nous permettre de travailler avec MySQL, puis ajouter "using MySql.Data.MySqlClient", cette assembly sera nécessaire dans notre classe "ConnexionBDD".

Les méthodes nécessaires à la gestion de l'accès des données sont :

- `ReqSelect()` : exécution d'une requête de type SELECT, le résultat de ces requêtes seront mémorisés dans le curseur objet nommé "reader".
- `Read()` : détermine si la fin du curseur est atteint.
- `Field()` : si la fin du curseur n'est pas atteint, le contenu du champ dont le nom est passé en paramètre est retourné.
- `Close()` : fermeture du curseur.
- `ReqUpdate()` : exécution d'une requête de type autre que SELECT, c'est à dire les INSERT, UPDATE ou DELETE.

3. Création du package dal

Ce package contiendra la classe technique "AccesDonnees" et toutes les méthodes (selon les besoins de l'application) qui répondront aux demandes du Contrôleur. La chaîne de connexion est utilisée pour établir la connexion avec la base de données et contient les paramètres : Server (serveur de la BDD), Database (nom de la BDD), User ID (nom d'utilisateur) et le Password (mot de passe).

```
7 using MediaTek.connexion;
8 using MediaTek.modele;
9
10 namespace MediaTek.dal
11 {
12     /// <summary>
13     /// Classe permettant de gérer les demandes concernant les données distantes
14     /// </summary>
15     public class AccesDonnees
16     {
17         /// <summary>
18         /// Chaîne de connexion à la BDD
19         /// </summary>
20         private static string connectionString = "Server=localhost;Database=mediatek86;User Id=mediatekadmin;Password=8Medi@Tek6$m;";
21     }
22 }
```

4. Création des classes du package Modèle

Le package Modèle doit contenir les classes métiers correspondant aux tables de notre base de données : "Absence", "Motif", "Personnel" et "Service".

```
3 namespace MediaTek.modele
4 {
5     /// <summary>
6     /// Classe métier correspondant à la table "Absence" de la base de données
7     /// </summary>
8     public class Absence
9     {
10         private DateTime datedebut;
11         private DateTime datefin;
12         private int idpersonnel;
13         private int idmotif;
14         private string motif;
```

Prenons l'exemple de la classe "Absence", les propriétés correspondent aux noms des champs de la table "Absence" de notre BDD et sont privées pour interdire la valorisation de ces propriétés par une simple affectation.

```
16     /// <summary>
17     /// Getter : datedebut (format: jj/mm/aaaa)
18     /// </summary>
19     public DateTime Datedebut { get => datedebut.Date; }
20
21     /// <summary>
22     /// Getter : datefin (format: jj/mm/aaaa)
23     /// </summary>
24     public DateTime Datefin { get => datefin.Date; }
25
26     /// <summary>
27     /// Getter : idpersonnel
28     /// </summary>
29     public int Idpersonnel { get => idpersonnel; }
30
31     /// <summary>
32     /// Getter : idmotif
33     /// </summary>
34     public int Idmotif { get => idmotif; }
35
36     /// <summary>
37     /// Getter : motif
38     /// </summary>
39     public string Motif { get => motif; }
```

Afin d'accéder aux données, des fonctions membres de type "Getter" sont créés. Par exemple pour accéder à la date de début d'une absence on écrira : "Absence.Datedebut;"

```

42     /// <summary>
43     /// Constructeur : valorise les propriétés
44     /// </summary>
45     /// <param name="datedebut"></param>
46     /// <param name="datefin"></param>
47     /// <param name="idpersonnel"></param>
48     /// <param name="idmotif"></param>
49     /// <param name="motif"></param>
50     public Absence(DateTime datedebut, DateTime datefin, int idpersonnel, int idmotif, string motif)
51     {
52         this.datedebut = datedebut;
53         this.datefin = datefin;
54         this.idpersonnel = idpersonnel;
55         this.idmotif = idmotif;
56         this.motif = motif;
57     }

```

Enfin, l'utilisation d'un constructeur sera utile dans cette classe notamment pour l'ajout des absences. Le constructeur est appelé, ses paramètres sont envoyés et les propriétés sont valorisées.

```

31         /// <summary>
32         /// Définit l'information à afficher : nom
33         /// </summary>
34         /// <returns>nom du service</returns>
35         public override string ToString()
36         {
37             return this.nom;
38         }

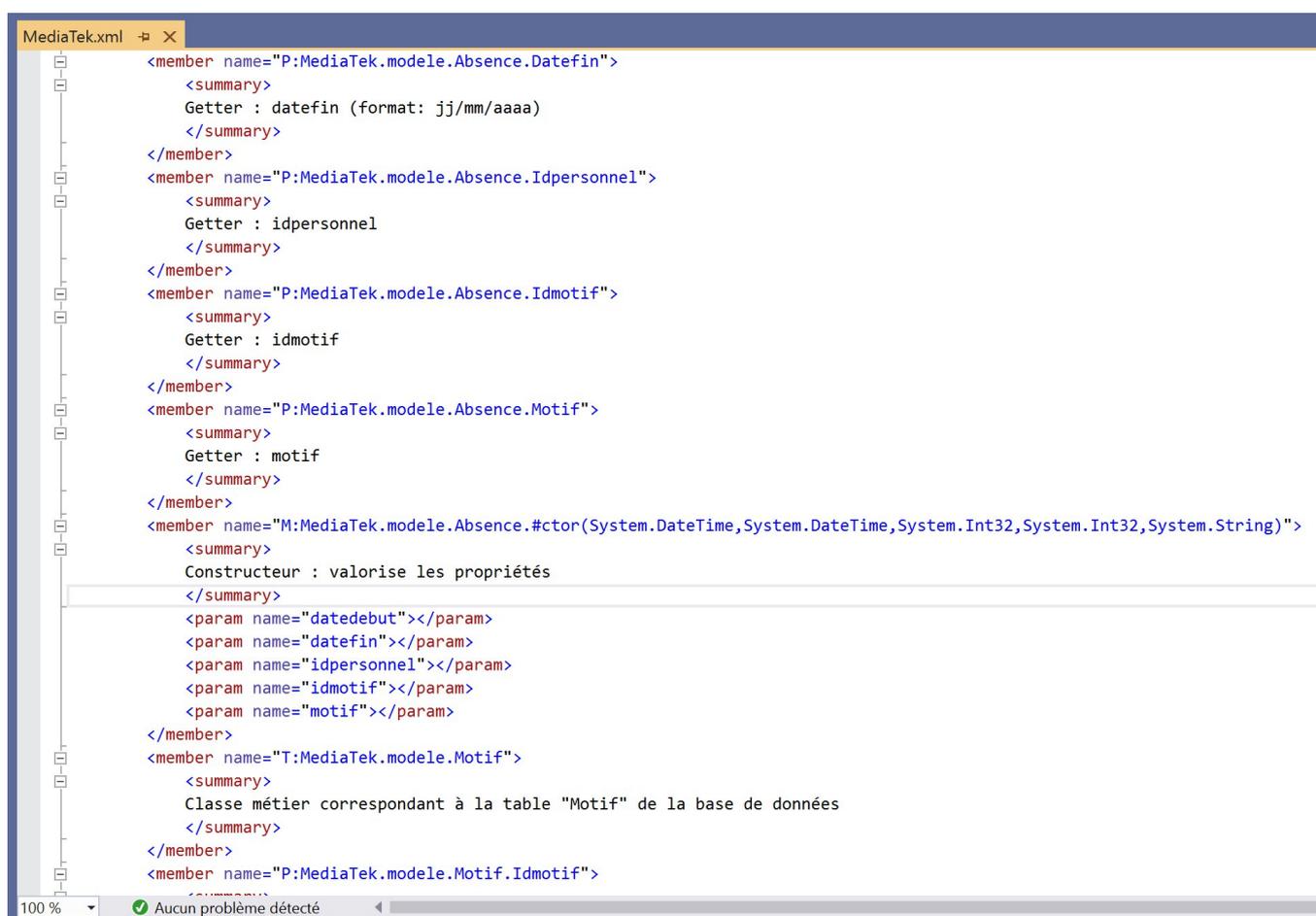
```

Les classes "Motif", "Service" et "Personnel" suivront la même structure que la classe "Absence". Les classes "Motif" et "Service" auront en plus des propriétés privées, getter et constructeur, une méthode ToString() redéfinie qui permettra de définir l'information à afficher.

5. Génération de la documentation technique

a. Fichier XML

Il faut avant tout s'assurer d'avoir sélectionné la génération de documentation XML dans Projet > Propriétés > Fichier de documentation XML. Ce fichier contiendra les classes et méthodes visibles avec leur commentaires normalisés.



```
MediaTek.xml
<member name="P:MediaTek.modele.Absence.Datefin">
  <summary>
    Getter : datefin (format: jj/mm/aaaa)
  </summary>
</member>
<member name="P:MediaTek.modele.Absence.Idpersonnel">
  <summary>
    Getter : idpersonnel
  </summary>
</member>
<member name="P:MediaTek.modele.Absence.Idmotif">
  <summary>
    Getter : idmotif
  </summary>
</member>
<member name="P:MediaTek.modele.Absence.Motif">
  <summary>
    Getter : motif
  </summary>
</member>
<member name="M:MediaTek.modele.Absence.#ctor(System.DateTime,System.DateTime,System.Int32,System.Int32,System.String)">
  <summary>
    Constructeur : valorise les propriétés
  </summary>
  <param name="datedebut"></param>
  <param name="datefin"></param>
  <param name="idpersonnel"></param>
  <param name="idmotif"></param>
  <param name="motif"></param>
</member>
<member name="T:MediaTek.modele.Motif">
  <summary>
    Classe métier correspondant à la table "Motif" de la base de données
  </summary>
</member>
<member name="P:MediaTek.modele.Motif.Idmotif">
  <summary>
```

L'inconvénient de ce fichier est qu'il est peu lisible. Il existe des outils tel que SandCastle Help File Builder qui permet de produire un site web à partir du fichier XML du projet et ainsi d'afficher toute la documentation sur une page. Le développeur qui consulte la documentation pourra naviguer entre les classes et méthodes et voir leur commentaires.

b. SandCastle Help File Builder

Après avoir installé et lancé le logiciel (<https://github.com/EWSoftware/SHFB/releases>), les étapes à suivre sont les suivantes :

- Fichier > Nouveau projet
- Sélectionner un dossier vide où la documentation en ligne sera générée
- Choisir le format et spécifier la syntaxe utilisée
- Cliquez-droit sur Documentation Source > Add
- Sélectionner le fichier XML du projet dans le dossier bin > Debug
- Cliquer sur le bouton "Build"

The image shows the Sandcastle Help File Builder (SHFB) interface on the left and a Windows File Explorer window on the right. The SHFB interface displays the documentation for the `ConnexionBDD.ReqSelect Method`. It includes the following information:

- Namespace:** MediaTek.connexion
- Assembly:** MediaTek (in MediaTek.exe) Version: 1.0.0.0 (1.0.0.0)
- Syntax:**

```
C#
public void ReqSelect(
    string stringQuery
)
```
- Parameters:**
 - `stringQuery`: Type: `System.String`, requête "SELECT"
- See Also:**
 - Reference: [ConnexionBDD Class](#), [MediaTek.connexion Namespace](#)

The File Explorer window shows the contents of the "Help" folder, which has been created. The folder contains the following files and subfolders:

- fti
- html
- icons
- scripts
- styles
- toc
- index (selected)
- LastBuild
- search
- SearchHelp.aspx
- SearchHelp.inc
- SearchHelp
- Web.Config
- WebKl.xml

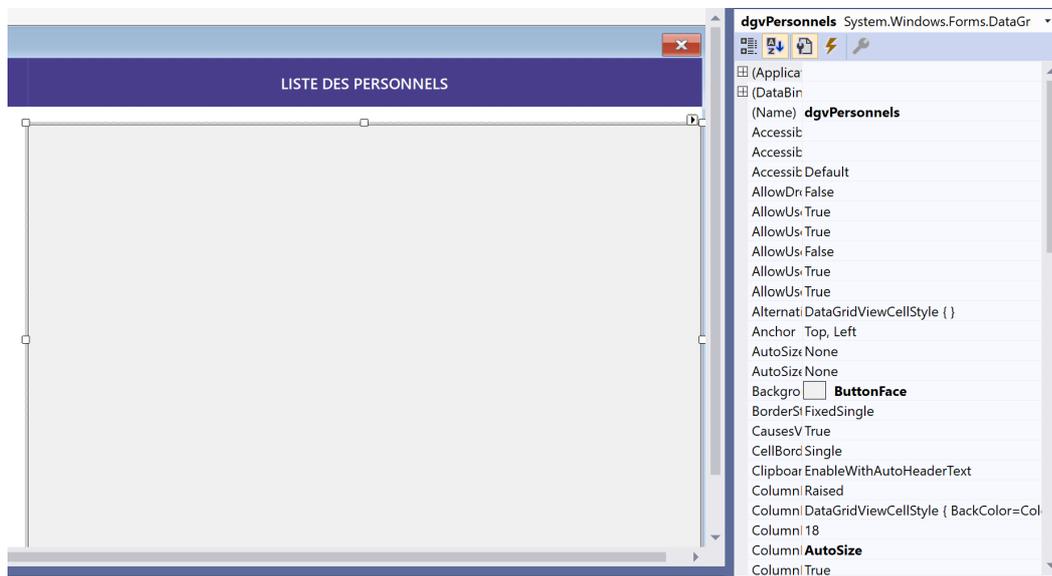
Une fois le build terminé, un dossier "Help" sera créé dans le dossier précisé précédemment, et contiendra le fichier index.html, qu'il sera possible d'ouvrir avec un navigateur web.

ÉTAPE 4

Coder les fonctionnalités de l'application à partir des cas d'utilisation

1. La démarche

(1) J'analyse en premier lieu, selon les cas d'utilisations, les composants graphiques qui recevront les données (textBox, comboBox, dataGridView), puis je détermine les données qui seront affichés.



(2) Ces informations me permettront d'écrire les requêtes SQL correspondant à chaque cas d'utilisation. Ces requêtes seront au préalable testées dans phpMyAdmin afin de s'assurer qu'elles fonctionnent.

✓ Affichage des lignes 0 - 8 (total de 9, traitement en 0,0005 seconde(s).)

```
SELECT p.idpersonnel AS idpersonnel, p.nom AS nom, p.prenom AS prenom, p.tel AS tel, p.mail AS mail, s.idservice AS idservice, s.nom AS service FROM personnel p JOIN service s USING (idservice) ORDER BY nom, prenom
```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

+ Options

idpersonnel	nom	1	prenom	2	tel	mail	idservice	service
9	Bass	Yuri	02 88 25 44 22	bassyuri@Quisque.co.uk	3	prêt		
6	Cain	Akeem	09 32 07 71 02	Duis@malesuadafamesac.ca	1	administratif		
8	Funes	Charles	07 60 47 98 10	at.risus.Nunc@rhoncusProin.com	3	prêt		
4	Gilmore	Hugo	01 60 00 62 27	nunc.ac@ac.com	3	prêt		
2	Hess	Rogan	08 60 51 27 77	Nam.nullamagna@miAliquamgravida.com	2	médiation culturelle		
3	Martin	Kelly	03 01 43 96 21	iaculis.quis.pede@sedturpisnec.org	3	prêt		
7	Morin	Estere	09 01 69 32 52	molestie.orci@ultricesadipiscingenim.com	3	prêt		
1	Odonnell	Raja	07 64 66 20 14	Aliquam.erat.voluptat@enim.co.uk	2	médiation culturelle		
10	Ung	Theary	07 46 27 19 01	theary@gmail.com	2	médiation culturelle		

(3) Dans la classe "AccesDonnees" du package dal, suivant les fonctionnalités, je crée la méthode contenant : la requête SQL, des paramètres s'il y a lieu (ReqUpdate), la récupération de l'instance de connexion, l'exécution de la requête et un *return* pour retourner les données s'il s'agit d'une requête de type SELECT.

```

50  /// <summary>
51  /// Récupère et retourne les personnels provenant de la BDD
52  /// </summary>
53  /// <returns>liste des personnels</returns>
54  public static List<Personnel> GetLesPersonnels()
55  {
56      List<Personnel> lesPersonnels = new List<Personnel>();
57      string req = "SELECT p.idpersonnel as idpersonnel, p.nom AS nom, p.prenom AS prenom, p.tel AS tel, p.mail AS mail, s.idservice AS idservice, s.nom AS nom A";
58      req += "FROM personnel p JOIN service s USING (idservice)";
59      req += "ORDER BY nom, prenom;";
60
61      ConnexionBDD curseur = ConnexionBDD.GetInstance(connectionString);
62      curseur.ReqSelect(req, null);
63      while (curseur.Read())
64      {
65
66
67          Personnel personnel = new Personnel((int)curseur.Field("idpersonnel"),
68              (string)curseur.Field("nom"),
69              (string)curseur.Field("prenom"),
70              (string)curseur.Field("tel"),
71              (string)curseur.Field("mail"),
72              (int)curseur.Field("idservice"),
73              (string)curseur.Field("service"));
74
75          lesPersonnels.Add(personnel);
76
77      }
78      curseur.Close();
79      return lesPersonnels;
80  }

```

(4) La classe "AccesDonnees" ne peut pas être appelé directement dans les classes de notre package Vue. En suivant la structure du MVC, je crée la méthode (get, add, del, update) dans la classe "Controleur" du package controleur, qui gère la demande (affichage, ajout, suppression, modification), celle-ci appelle la méthode (exécutant la requête) de la classe "AccesDonnees".

```

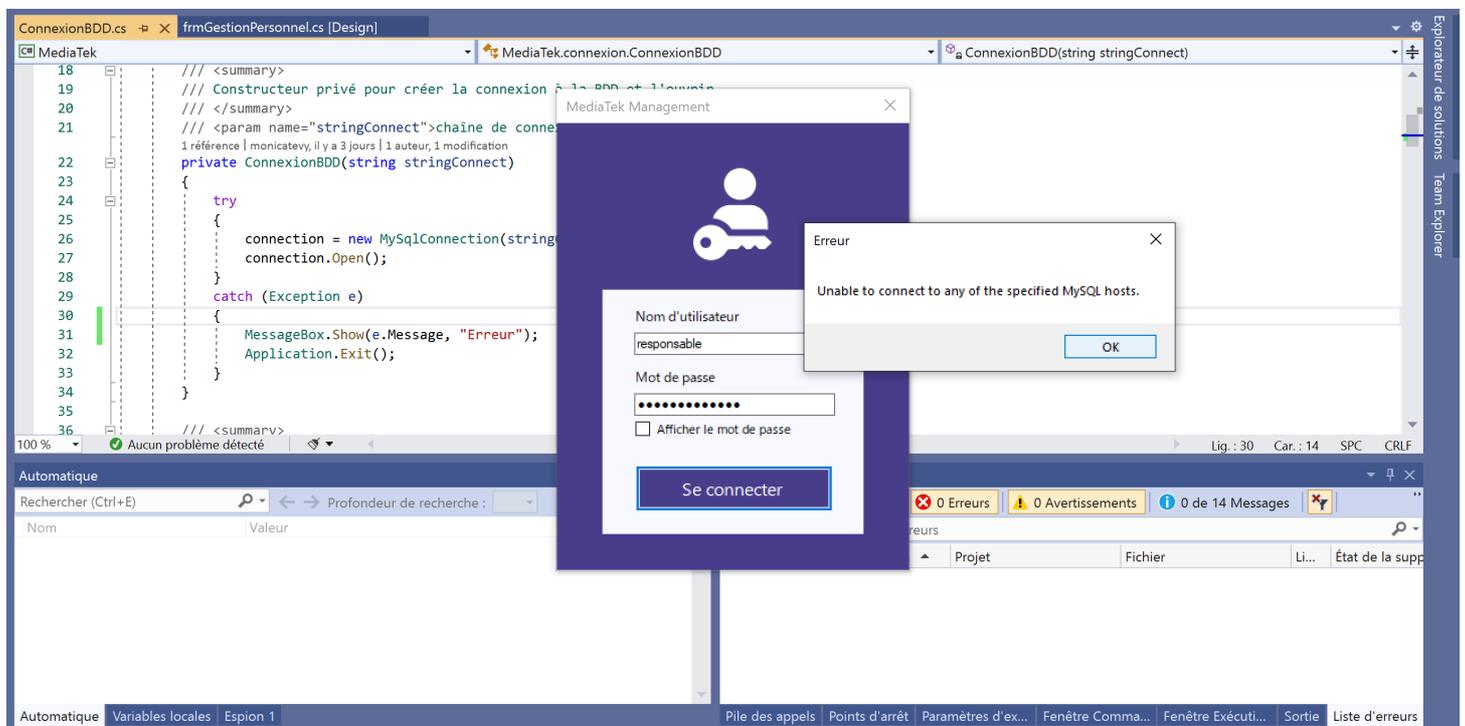
49      /// <summary>
50      /// Récupère et retourne les infos du personnel provenant de la BDD
51      /// </summary>
52      /// <returns>liste des personnels</returns>
53      public List<Personnel> GetLesPersonnels()
54      {
55          return AccesDonnees.GetLesPersonnels();
56      }

```

(5) À ce stade, dans nos classes du package Vue, je crée la procédure événementielle (Click) ou dans notre cas, la méthode qui demande au Contrôleur d'effectuer certaines manipulations et d'afficher les données dans les composants graphiques.

```
119     /// <summary>
120     /// Affiche les personnels
121     /// </summary>
122     public void RemplirDGVPersonnels()
123     {
124         List<Personnel> lesPersonnels = controle.GetLesPersonnels();
125         bsPersonnels.DataSource = lesPersonnels;
126         dgvPersonnels.DataSource = bsPersonnels;
127         dgvPersonnels.Columns["idpersonnel"].Visible = false;
128         dgvPersonnels.Columns["idservice"].Visible = false;
129         dgvPersonnels.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells;
130     }
```

(6) Je teste l'application à chaque fois que j'apporte une modification pour m'assurer que ce que je viens de coder répond aux attentes et que cela n'interfère pas dans le fonctionnement des autres fonctionnalités. J'affiche les messages d'erreur dans un MessageBox pour faciliter le débogage.



2. Les difficultés rencontrées

a. Les propriétés des éléments graphiques

Dans l'ensemble, les principales difficultés sont dues à un manque de connaissances quant aux propriétés existantes d'un élément tel que le DataGridView (DGV). La gestion de l'affichage des informations d'un personnel dans les textBox lorsqu'une ligne du DataGridView est sélectionné n'a pas été évidente. Après avoir fait des dizaines de recherches et les documentations, j'ai testé les évènements "CellClick", "SelectionChanged" et "CellEnter", cette dernière m'a semblé être la plus adaptée.

```
167     /// <summary>
168     /// Affiche les informations du personnel sélectionné
169     /// </summary>
170     /// <param name="sender"></param>
171     /// <param name="e"></param>
172     private void dgvPersonnels_CellEnter(object sender, DataGridViewCellEventArgs e)
173     {
174         DataGridViewRow row = dgvPersonnels.CurrentRow;
175         txtNom.Text = row.Cells["Nom"].Value.ToString();
176         txtPrenom.Text = row.Cells["Prenom"].Value.ToString();
177         txtTel.Text = row.Cells["Tel"].Value.ToString();
178         txtEmail.Text = row.Cells["Mail"].Value.ToString();
179         cboServices.Text = row.Cells["Service"].Value.ToString();
180     }
```

Pour que le tout soit synchronisé, il a fallu gérer le curseur du DGV ainsi que la cellule sélectionnée, souvent avec un foreach () pour se positionner sur la ligne qui contient une valeur donnée.

```
355     // Sélectionne l'élément modifié ou à modifier (idpersonnel)
356     dgvPersonnels.ClearSelection();
357     int rowIndex = -1;
358     foreach (DataGridViewRow row in dgvPersonnels.Rows)
359     {
360         if (row.Cells["idpersonnel"].Value.ToString().Equals(personnel.Idpersonnel.ToString()))
361         {
362             rowIndex = row.Index;
363             break;
364         }
365     }
366     dgvPersonnels.CurrentCell = this.dgvPersonnels[1, rowIndex];
367     dgvPersonnels.CurrentCell.Selected = true;
```

b. Les conflits en base de données

Lors de l'ajout d'un personnel depuis l'application, il m'a semblé important de ne pas pouvoir ajouter plusieurs fois un personnel avec des informations identiques. La base de données gérant l'incrémentation de l'id du personnel (clé primaire), la multiplication des enregistrements est inévitable.

idpersonnel	nom	prenom	tel	mail	idservice
1	Garett	Michelle	07 49 46 27 80	garettm@mail.com	2
2	Garett	Michelle	07 49 46 27 80	garettm@mail.com	2
3	Garett	Michelle	07 49 46 27 80	garettm@mail.com	2
4	Garett	Michel	07 49 46 27 80	garettm@mail.com	2

Pour remédier à ce problème j'ai fait le choix d'autoriser les ajouts ou les modifications à condition que la combinaison nom-prénom n'existe pas déjà dans la base de donnée, en partant du principe qu'il est peu probable de trouver dans le même réseau deux personnes portant le même nom-prénom.

```
101     /// <summary>
102     /// Ajoute un personnel
103     /// </summary>
104     /// <param name="personnel">objet de type personnel à ajouter</param>
105     public static void AddPersonnel(Personnel personnel)
106     {
107         string req = "INSERT INTO personnel(nom, prenom, tel, mail, idservice) ";
108         req += "SELECT * FROM(SELECT @nom AS nom, @prenom AS prenom, @tel AS tel, @mail AS mail, @idservice AS idservice) AS req ";
109         req += "WHERE NOT EXISTS(SELECT nom, prenom FROM personnel WHERE nom = @nom AND prenom = @prenom); ";
110
111         Dictionary<string, object> parameters = new Dictionary<string, object>();
112         parameters.Add("@nom", personnel.Nom);
113         parameters.Add("@prenom", personnel.Prenom);
114         parameters.Add("@tel", personnel.Tel);
115         parameters.Add("@mail", personnel.Mail);
116         parameters.Add("@idservice", personnel.Idservice);
117         ConnexionBDD connexion = ConnexionBDD.GetInstance(connectionString);
118         connexion.ReqUpdate(req, parameters);
119     }
```

Dans la même logique, il ne devrait pas être possible d'ajouter une absence avec une date de début existante pour un même personnel du fait que la clé primaire est composée, la combinaison datedebut-idpersonnel doit être unique dans la base de données.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 idpersonnel	int(11)			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	2 datedebut	datetime			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	3 idmotif	int(11)			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	4 datefin	datetime			Oui	NULL			Modifier Supprimer Plus

Si l'on gère les heures de départ d'absence et les heures de retour, il est possible d'avoir deux dates de début identiques mais avec une heure de début différente. À mon sens il est inutile de gérer l'heure, en général un employé s'absente pendant période, une journée ou une demi-journée. Par simplicité, nous gérons que la date (jj-mm-aaaa) dans notre application.

Le cas d'utilisation n°8 doit permettre la modification d'une absence, ce qui implique la modification de la date de début. Traduire cette modification par une requête de modification ReqUpdate() posera problème car il se peut que la valeur de clé primaire existe déjà.

Une manière d'éviter ces conflits en base de données serait de ne pas utiliser de données réelles comme clé primaire, c'est à dire des valeurs qui pourraient être modifiées par la suite par l'utilisateur, et plutôt privilégier la clé primaire à incrémentation automatique. C'est une précaution à prendre dans les prochains projets.

Il est maintenant trop tard pour revenir à la conception de la base de données et y apporter des modifications. On pourrait rendre la modification de la date de début impossible avec "DateTimePicker.Enabled = false;" ce qui pousserait l'utilisateur à supprimer de lui même l'absence pour en rajouter une nouvelle avec les bonnes informations. D'un point de vue expérience utilisateur, ce n'est sans doute pas la meilleure option.

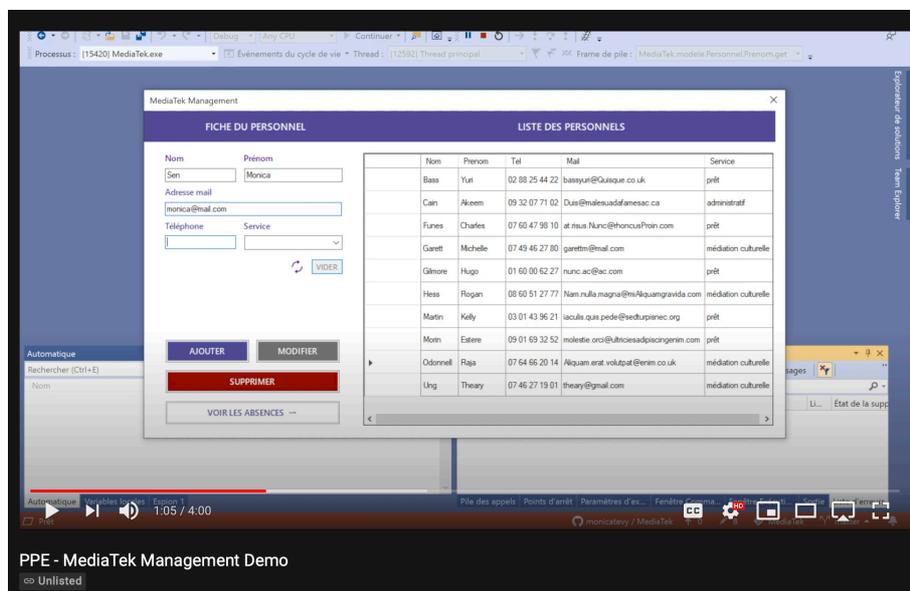
Finalement, la solution que j'ai trouvé est de donner l'illusion d'une modification en supprimant l'absence, ce qui supprimera la clé primaire, et en ajoutant une nouvelle absence qui remplace celle supprimée, à condition que la clé primaire n'existe pas déjà. Dans le cas où la clé primaire existe dans la base de données, l'ajout n'a pas pu se faire, il faudra restaurer l'absence supprimée et informer l'utilisateur.

cas n°1	idpersonnel	datedebut	datefin	idmotif
<i>Absence à modifier</i>	1	01-06-2021	03-06-2021	2
<i>(1) suppression</i>	1	01-06-2021	03-06-2021	2
<i>(2) ajout</i>	1	29-05-2021	03-06-2021	2

cas n°2	idpersonnel	datedebut	datefin	idmotif
<i>Absence à modifier</i>	1	01-06-2021	03-06-2021	2
<i>(1) suppression</i>	1	01-06-2021	03-06-2021	2
<i>(2) pas d'ajout car conflit</i>	1	01-06-2021	07-07-2021	1
<i>(3) ajoute l'absence initiale</i>	1	01-06-2021	03-06-2021	2

ÉTAPE 5

Créer une documentation utilisateur
en vidéo



L'enregistrement de l'écran s'est fait avec le logiciel Bandicam,
l'enregistrement de la voix avec le logiciel de montage vidéo iMovie.
J'ai importé la vidéo sur Youtube faciliter les prochaines manipulations

ÉTAPE 6

Gérer le déploiement
Rédiger le compte rendu
Créer la page du portfolio

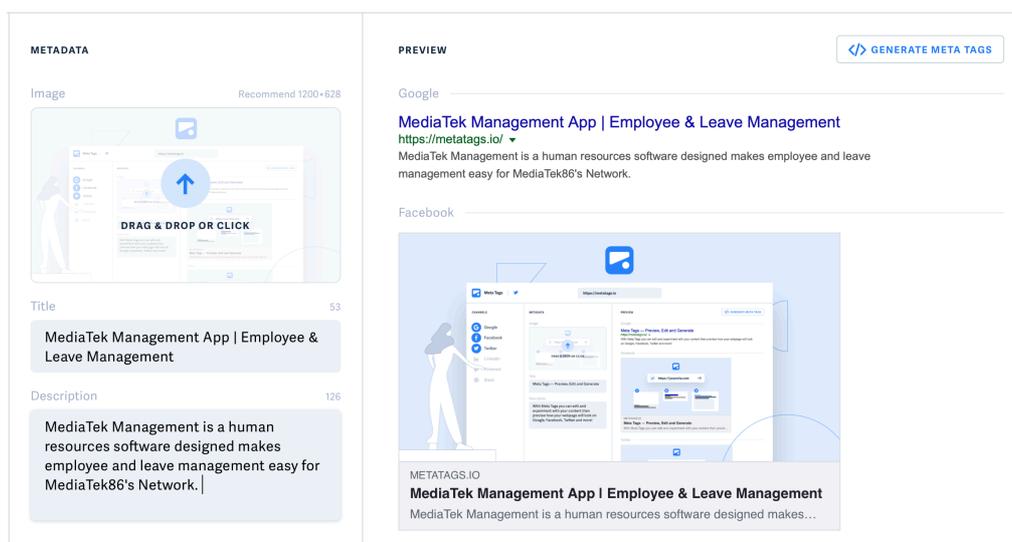
1. Déploiement de l'application

The screenshot shows a file explorer window with a search bar and a list of files. The files are organized into folders: 'Application Files', 'MediaTek', and 'setup'. The 'setup' file is highlighted.

Nom	Modifié le	Type
Application Files	04/06/2021 12:23	Dossier de fichiers
MediaTek	04/06/2021 12:23	Application Manifest
setup	04/06/2021 12:23	Application

2. Création de la page du portfolio

Ayant réalisé quelques projets personnels auparavant, j'ai décidé de coder mon site de portfolio sans l'utilisation de CMS tel que WordPress ou Wix pour des soucis de flexibilités et de rapidités. Mon portfolio, ainsi que la page dédiée à la mission sera hébergé sur Github.



Utilisation d'un outil de pré-visualisation des balises <meta>



WORK

First year project (PPE)

 PPE stands for "Projet Personnalisé Encadré" in French. As part of my first year in Computer Science, I've build this application using C# in Visual Studio Code. I designed the UI and created the Database using MySQL (WAMPServer). You can check my work by downloading the PDF. (Compte rendu d'activité)

[DOWNLOAD PDF](#) [GITHUB REPO](#)

Vous aurez la possibilité de télécharger ce Compte rendu d'activité depuis mon site web et de consulter le dépôt distant du projet.

ATELIER DE PROFESSIONNALISATION

Démonstration de l'application : [en vidéo](#)

Page de portfolio dédiée à la mission : [ici](#)

Dépôt distant du projet : [sur Github](#)

BILAN

Le développement de l'application est maintenant terminé, contre toute attente, j'ai énormément apprécié faire ce projet, de la conception de la base de données, en passant par le design des interfaces utilisateurs, le développement des fonctionnalités jusqu'au déploiement de l'application, la création de la page de portfolio, et même la rédaction de ce compte rendu. J'ai investi du temps, des efforts, qui finalement n'étaient pas motivés par une note, mais par la volonté de créer une application qui facilite les manipulations et qui soit agréable pour l'utilisateur. Je voulais aller au-delà de ce qui était attendue pour ce PPE, notamment pour que ce projet serve d'argument pour des recruteurs potentiels.

Par ailleurs, ce projet étant bien différent que ceux réalisés au cours de l'année, il s'agissait d'un réel challenge. Cela m'a permis de confirmer mes choix d'orientations, car j'aspire à être Web Développeur, le front-end m'attirait déjà mais la réalisation de cette application me fait comprendre que le back-end n'est pas insurmontable, mieux encore, les défis dans ce domaine sont infinis.