

COMPTE-RENDU AP2

Dépot en ligne: <https://github.com/DemonHuman/Mediatek86>

Sommaire

1Rappel du contexte.....	2
2Rappel de la mission.....	2
3Réalisation.....	2
Étape 1.....	2
Étape 2.....	2
Étape 3.....	3
Étape 4.....	7
Étape 5.....	7
Étape 6.....	7

1 Rappel du contexte

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.

2 Rappel de la mission

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#.

3 Réalisation


Étape 1

-Installation de Wampserver

-Installation de Visual Studio 2019

-Installation de Win'Design

-Génération du script sql (SQL -> extrait windesign.sql)


 [extrait windesign.sql](#)


-Mise en place de la base de données MySQL


-Ajout utilisateur


```
1 CREATE USER 'responsable'@'localhost' IDENTIFIED BY 'Admin123!';
2 GRANT CREATE ON * . * TO 'responsable'@'localhost';
3 GRANT SELECT ON * . * TO 'responsable'@'localhost';
4 GRANT UPDATE ON * . * TO 'responsable'@'localhost';
5 GRANT INSERT ON * . * TO 'responsable'@'localhost';
6 GRANT DELETE ON * . * TO 'responsable'@'localhost';
```

-Alimentation de la base de données (SQL -> remplir motif.sql/ remplir personnel & absence.sql/remplir service.sql / ajout responsable.sql)

 [remplir motif.sql](#)

 [remplir personnel & absence.sql](#)

 [remplir service.sql](#)

 [ajout responsable.sql](#)

Étape 2

-Maquettage de l'application (Dossier Design)

-Création du projet (Projet source.zip)

-Codage de la partie vue de l'application (Projet source -> vue)

- ✖ Form Absence.Designer.cs
- ✖ Form Login.Designer.cs
- ✖ Form Personnel.Designer.cs
- ✖ Form Principale.Designer.cs

```

/// <summary>
/// Créer la vue pour le formulaire d'édition/ajout du personnel
/// </summary>
private void InitializeComponent()
{ ...
}

/// <summary>
/// Remplie les éléments avec les données précédemment sélectionnées (valeurs par défaut si mode ajout)
/// </summary>
public void LoadData(bool edit = false, int id = -1, string debut = "", string fin = "", string motif = "", string nom = "")
{ ...
}

```

Étape 3

-Installation du package MySql.Data

-Codage des requetes bdd (Projet source -> dal.cs)

```

namespace dal
{
    public class sqlreq
    {
        private static string connectionString = "server=localhost;userid=responsable;password=pn995T2S;database=MediaTek86";
        private static MySqlConnection sql = new MySqlConnection(connectionString);

        /// <summary>
        /// Vérifie dans la bdd si les identifiants/mots de passe sont correctes
        /// </summary>
        /// <param name="login"></param>
        /// <param name="pwd"></param>
        public bool checkLogin(string login, string pwd)
        { ...
        }

        /// <summary>
        /// Get les données des services dans la bdd afin de valoriser les classes dans une liste
        /// </summary>
        public List<Service> GetServices()
        { ...
        }

        /// <summary>
        /// Get les données des motifs dans la bdd afin de valoriser les classes dans une liste
        /// </summary>
        public List<Motif> GetMotifs()
        { ...
        }

        /// <summary>
        /// Get le personnel et tout ce qui en découle
        /// </summary>
        public List<Personnel> GetPersonnel()
        { ...
        }
    }
}

```

```

-///-<summary>
-///Ajoute des données dans la table Personnel de la BDD
-///-</summary>
-///-<param name="idservice"></param>
-///-<param name="nom"></param>
-///-<param name="prenom"></param>
-///-<param name="tel"></param>
-///-<param name="mail"></param>
-///-<returns>le nombre de lignes affectées par la requete (normalement 1 ici)</returns>
public int AjoutPersonnel(int idservice, string nom, string prenom, string tel, string mail)
{ ...
}

-///-<summary>
-///Modifie les données d'une ligne de la table Personnel de la BDD
-///-</summary>
-///-<param name="id"></param>
-///-<param name="idservice"></param>
-///-<param name="nom"></param>
-///-<param name="prenom"></param>
-///-<param name="tel"></param>
-///-<param name="mail"></param>
-///-<returns>le nombre de lignes affectées par la requete (normalement 1 ici)</returns>
public int EditPersonnel(int id, int idservice, string nom, string prenom, string tel, string mail)
{ ...
}

-///-<summary>
-///Supprime une ligne de la table Personnel de la BDD
-///-</summary>
-///-<param name="id"></param>
-///-<returns>le nombre de lignes affectées par la requete (normalement 1 ici)</returns>
public int SuppPersonnel(int id)
{ ...
}

-///-<summary>
-///Ajoute des données dans la table absence de la BDD
-///-</summary>
-///-<param name="id"></param>
-///-<param name="debut"></param>
-///-<param name="fin"></param>
-///-<param name="motif"></param>
-///-<returns>le nombre de lignes affectées par la requete (normalement 1 ici)</returns>
public int AjoutAbsence(int id, DateTime debut, DateTime fin, int motif)
{ ...
}

```

-Codage du package modele (projet source -> modele.cs) -Codage des modèles (Projet source.zip -> modele.cs)

```

-///-<summary>
-///Modifie les données d'une ligne de la table absence de la BDD
-///-</summary>
-///-<param name="id"></param>
-///-<param name="debut"></param>
-///-<param name="fin"></param>
-///-<param name="motif"></param>
-///-<param name="oldDebut"></param>
-///-<returns>le nombre de lignes affectées par la requete (normalement 1 ici)</returns>
public int EditAbsence(int id, DateTime debut, DateTime fin, int motif, DateTime oldDebut)
{ ...
}

-///-<summary>
-///Supprime une ligne de la table absence de la BDD
-///-</summary>
-///-<param name="id"></param>
-///-<param name="debut"></param>
-///-<returns>le nombre de lignes affectées par la requete (normalement 1 ici)</returns>
public int SuppAbsence(int id, DateTime debut)
{ ...
}
}

```

```

1  using System;
2  using System.Collections.Generic;
3
4  namespace modele
5  {
6      ... public class absence
7  >      ... { ...
43      ... }
44
45      ... public class Personnel
46  >      ... { ...
109      ... }
110
111      ... public class motif
112  >      ... { ...
144      ... }
145
146      ... public class service
147  >      ... { ...
179      ... }
180  }

```

Étape 4


- Codage des fonctionnalités l'application (controleur etc)
- Testes et corrections de bugs

Étape 5

- Création d'une vidéo de démonstration.

Étape 6

- Extraction du projet (dossier Programme)
- Génération du script complet de la bdd (SQL -> genererTout.sql)

 genererTout.sql