

INSCRIPTION DES PARTICIPANTS LOG/IN/OUT

Documentation Technique

1. Cahier des charges

Inscription des participants :

Pour chaque participant, on conserve aussi nom, prénom, adresse, @mail, numéro de portable. Les intervenants et les bénévoles sont parfois aussi licenciés, mais ce n'est pas systématique. Un licencié peut être bénévole un jour et participer aux ateliers un autre jour. En revanche, les intervenants assurent l'animation ou l'intervention sur un seul atelier pendant les deux jours. Un atelier est animé par un seul intervenant (toujours le même qui est le seul à avoir le titre d'animateur). Plusieurs autres intervenants peuvent éventuellement intervenir sur ce même atelier. Fonctionnalités attendues de l'application :

-L'inscription des participants, incluant les ateliers auxquels ils participent ou dans lesquels ils interviennent, et demi-journées d'intervention des bénévoles.

-La visualisation des participants, par nom, par atelier, par type.

Authentification :

Le gestionnaire devra s'authentifier au lancement de l'application. Ensuite son login sera rappelé sur tous les formulaires. Les informations concernant un gestionnaire sont le login et le mot de passe.

2. Information de connexion

Authentification application :

Username : user

Password : user

3. CLASSES

Atelier.cs

Classe permettant la création d'objet(s) Atelier cette classe comprend les propriétés privées, le constructeur et les accesseurs.

Connect.cs

Classe permettant la création d'objet(s) Connect.

Cette classe contient la méthode `seConnecter`.

`seConnecter(string id, string pw)` prend en compte deux paramètres et renvoie un booléen.

//La méthode est utilisée pour vérifier à l'aide d'une requête SQL SELECT l'existence du login et du password entré par l'utilisateur. Si les données existent dans la base de données la variable interne « compteur » prend +1 en valeur et renvoie true sinon elle renvoie false.

HorrairesBenevoles.cs

Classe permettant la création d'objet(s) HorrairesBenevoles cette classe comprend les propriétés privées, le constructeur et les accesseurs.

Participant.cs

Classe permettant la création d'objet(s) Participant cette classe comprend les propriétés privées, le constructeur et les accesseurs.

TypeParticipant.cs

Classe permettant la création d'objet(s) TypeParticipant cette classe comprend les propriétés privées, le constructeur et les accesseurs.

DAOFactory.cs

Classe permettant la création d'objet(s) DAOFactory.

Cette classe contient les données de connexions à la base de données SQLServer.

Cette classe contient les méthodes suivantes : `connecter`, `déconnecter`, `SqlDataReader`, `execSQLWrite`.

`connecter()` permet la connexion à la base de données.

`deconnecter()` permet la déconnexion à la base de données.

SqlDataReader(string requete) permet l'exécution d'une requête de lecture (SELECT) retourne un Datareader.

execSQLWrite(string requete) permet l'exécution d'une requête d'écriture (INSERT ou UPDATE) ne retourne rien.

DAOAtelier.cs

Classe permettant la création d'objet(s) DAOAtelier.

Cette classe contient la méthode getAllAteliers.

getAllAtelier() retourne une liste lesAteliers composée d'objet(s) Atelier at, les données sont récoltées dans la base de données grâce à une requête SELECT.

//DAOParticipant.cs

Classe permettant la création d'objet(s) DAOParticipant.

Cette classe contient la méthode getAllParticipants, AjouterParticipant.

getAllParticipants() retourne une liste lesParticipants composée d'objet(s) Participant p, les données sont récoltées dans la base de données grâce à une requête SELECT.

```
AjouterParticipant (string unNom, string unPrenom, int
unIdType, string uneAdresse, string unMail, string
unNumPortable, int unIdAtelier, int
unIdHorraireBenevoles)
```

Permet d'ajouter des participants dans la base de données grâce à une requête INSERT elle ne retourne rien.

DAOHorrairesBenevoles.cs

Classe permettant la création d'objet(s) DAOHorrairesBenevoles.

Cette classe contient la méthode getAllHorrairesBenevoles.

getAllHorrairesBenevoles() retourne une liste lesHorrairesBenevoles composée d'objet(s) HorrairesBenevoles hb, les données sont récoltées dans la base de données grâce à une requête SELECT.

DAOTypeParticipant.cs

Classe permettant la création d'objet(s) DAOTypeParticipant.

Cette classe contient la méthode `getAllTypeParticipant`.

`getAllTypeParticipant()` retourne une liste `lesTypesParticipants` composée d'objet(s) `TypeParticipant tp`, les données sont récoltées dans la base de données grâce à une requête `SELECT`.

4. FORM

Login.cs

Designer : permet la création de l'interface graphique de Login.

Code :

- Evènement :

- 1) `btnConnexion_Click` : Bouton permettant lors du clique de cacher le Form Login pour faire par la suite apparaitre le Form APPLI_PPE.

APPLI_PPE.cs

Designer : permet la création de l'interface graphique d'APPLI_PPE.

Code :

- Evènement :

- 1) `buttonCreateParticipant_Click` : Prend en compte tout les champs nécessaire à l'inscription d'un participant. Selon certaines conditions qui doivent être respectées créer un objet `DAOParticipant` en lui appliquant la fonction `AjouterParticipant` (classe `DAOParticipant`)
- 2) `textBoxNom_KeyPress/textBoxPrenom_KeyPress/textBoxTel_KeyPress` : Sont utiles pour le contrôles des caractères entrés dans les champs.

- 3) **btnEnd_Click** : Est utilisé pour fermer l'application (Quitter application)
- 4) **btnLogOut_Click** : Est utilisé pour cacher le form APPLI_PPE faisant apparaître par la suite le form Login (Se déconnecter de l'application)

5. METHODES SUPPLEMENTAIRES FORM APPLI_PPE

1) **GetLibForAtelier(int atelier) : string**
Remplace idAtelier par LibelleAtelier.

2) **GetLibForHorraireBenevoles(int Horraire) : string**
Remplace idHorraire par LibelleHorraire.

3) **GetLibForType(int Type) : string**
Remplace idType par LibelleType.

4) **refreshDgvParticipant()**
Refresh le Dgv.

5) **GetIdForType(string type) : int**
Remplace LibelleType par IdType.

6) **GetIdForAtelier(string atelier) :int**
Remplace LibelleAtelier par NumAtelier.

7) **GetIdForHorraireBenevoles(string Horraire) : int**

Remplace LibelleHorraires par IdHorraires.