LAFFI RAYAN

Rayan.laffi@icof.fr

|  |
| --- |
|  |

|  |
| --- |
| Accueil - ICOF - Lycée technologique secondaire et supérieur privé sous  contrat |

Résumé

Cette application permet à la MDL la mise en place d’une couverture informatique d'une manifestation biennale appelée "Assises du Sport Lorrain".

**Documentation technique**

**d’une application de gestion des Assises du Sport Lorrain**

Table des matières

[Informations 2](#_Toc72793401)

[Description de l’application 2](#_Toc72793402)

[Technologies utilisées 2](#_Toc72793403)

[Description de la base de données 2](#_Toc72793404)

[Description des classes utilisées 3](#_Toc72793405)

[Description de la structure utilisée 4](#_Toc72793406)

# Informations

Nom du projet : application de gestion des Assises du Sport Lorrain

Type de document : Documentation technique

Auteur : LAFFI RAYAN

# Description de l’application

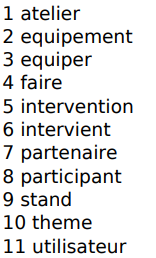
Cette application permet à la MDL la mise en place d’une couverture informatique d'une manifestation biennale appelée "Assises du Sport Lorrain". Elle permet de visualiser les différentes informations des ateliers mais aussi de les modifier.

# Technologies utilisées

Le logiciel de gestion est codé en C# (l’architecture utilise le modèle DAO orientée objet).

# Description de la base de données

Elle contient 11 tables :



[**VOIR LES TABLES ET LE SCHEMA RELATIONNEL**](description%20de%20la%20BDD.pdf)

# Description des classes utilisées

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASSES** | | | |
| Atelier | Theme | utilisateur | Participant | |
| **ATTRIBUTS** | | | |
| id | nPoste | id | indIP | |
| capaciteMax | nomPoste | nbPoste | nomSegment | |
| nom | indIP | indIP | etage | |
| heureFin | ad |  | nbSalle | |
| heureDebut | typePoste |  | nbPoste | |
| description | idSalle |  |  | |
| id\_Participant | nbLog |  |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLASSES** | | | |
| Atelier | Theme | utilisateur | Participant |
| **ATTRIBUTS** | | | |
| id | id | id | id |
| capaciteMax | nom | Nom | Nom |
| nom | id\_Atelier | Prenom | Prenom |
| heureFin |  | Mdp | adresse |
| heureDebut |  | email | mail |
|  |  |  | numero |
|  |  |  | club |
| **METHODE** | | | |
| listeAtelier | listeTheme | getUtilisateur | listeParticipant |
| getDicoAteliers | listeThemeByAtelier |  | getParticipant |
| getAtelier |  |  | getAllIntervenantParAtelier |
| updateAnimateur |  |  | getIntervenantsNonAffecter |
| updateHeure |  |  | supprimerIntervenant |
| updateCapacite |  |  | insertIntervenant |
|  |  |  | getDicoIntervenantDansAtelier |
|  |  |  | getDicoIntervenants |
|  |  |  | getDicoIntervenantsNonAffecter |
|  |  |  | getNombreParticipantNonAffecter |
|  |  |  | getNombreParticipantDansAtelier |
|  |  |  |  |

|  |  |
| --- | --- |
| **METHODES DE LA CLASSE ATELIER** | |
| listeAtelier | Cette méthode retourne une collection d’objet atelier |
| getAtelier | Cette méthode prend 1 paramètre et retourne un objet atelier |
| getDicoAteliers | Renvoie un dictionnaire avec les ateliers |
|  |  |
| updateAnimateur | Cette méthode prend en paramètre les informations pour mettre à jour un animateur dans un atelier |
| updateHeure | Méthode pour mettre à jour l’heure de début et de fin. |
| updateCapacite | Méthode pour mettre à jour la capacité. |
|  |  |

|  |  |
| --- | --- |
| **METHODES DE LA CLASSE THEME** | |
| listeTheme | Cette méthode retourne une collection d’objets THEME |
| listeThemeByAtelier | Cette méthode prend 1 paramètre et retourne une collection d’objets postes par atelier |

|  |  |
| --- | --- |
| **METHODES DE LA CLASSE UTILISATEUR** | |
| getUtilisateur | Cette méthode prend en paramètre l’email de l’utilisateur et retourne un objet utilisateur |

|  |  |
| --- | --- |
| **METHODES DE LA CLASSE PARTICIPANT** | |
| listeParticipant | Cette méthode retourne une collection d’objets participant |
| getParticipant | Cette méthode prend 1 paramètre et retourne un objet participant |
| getAllIntervenantParAtelier | Cette méthode prend en paramètre l’id de l’atelier et retourne une collection d’objets participant |
| getIntervenantsNonAffecter | Cette méthode prend en paramètre l’id de l’atelier et retourne une collection d’objets participant qui n’ont pas été affectée |
| supprimerIntervenant | Cette méthode prend en paramètre l’id du participant et supprime l’intervenant |
| insertIntervenant | Cette méthode insert un nouvel intervenant à l’atelier. Il prend deux paramètres en compte : idIntervenant, idAtelier |
| getIntervenantsNonAffecter | Cette méthode renvoie une liste d’intervenants non affecter dans un atelier. |
| getDicoIntervenantDansAtelier | Renvoie un dictionnaire des intervenants dans l’atelier |
| getDicoIntervenants | Renvoie un dictionnaire des intervenants |
| getDicoIntervenantsNonAffecter | Renvoie un dictionnaire des intervenants non affecter |
| getNombreParticipantNonAffecter | Renvoie le nombre de participant non affecter (inutile pour le moment) |
| getNombreParticipantDansAtelier | Renvoie le nombre de participants dans l’atelier. |

# Description de la structure utilisée

Le projet utilise une architecture DAO orientée objet. Le code à été commentée pour mieux le comprendre.