RolistMaker – Admin

Documentation technique

**Rappel du projet :**

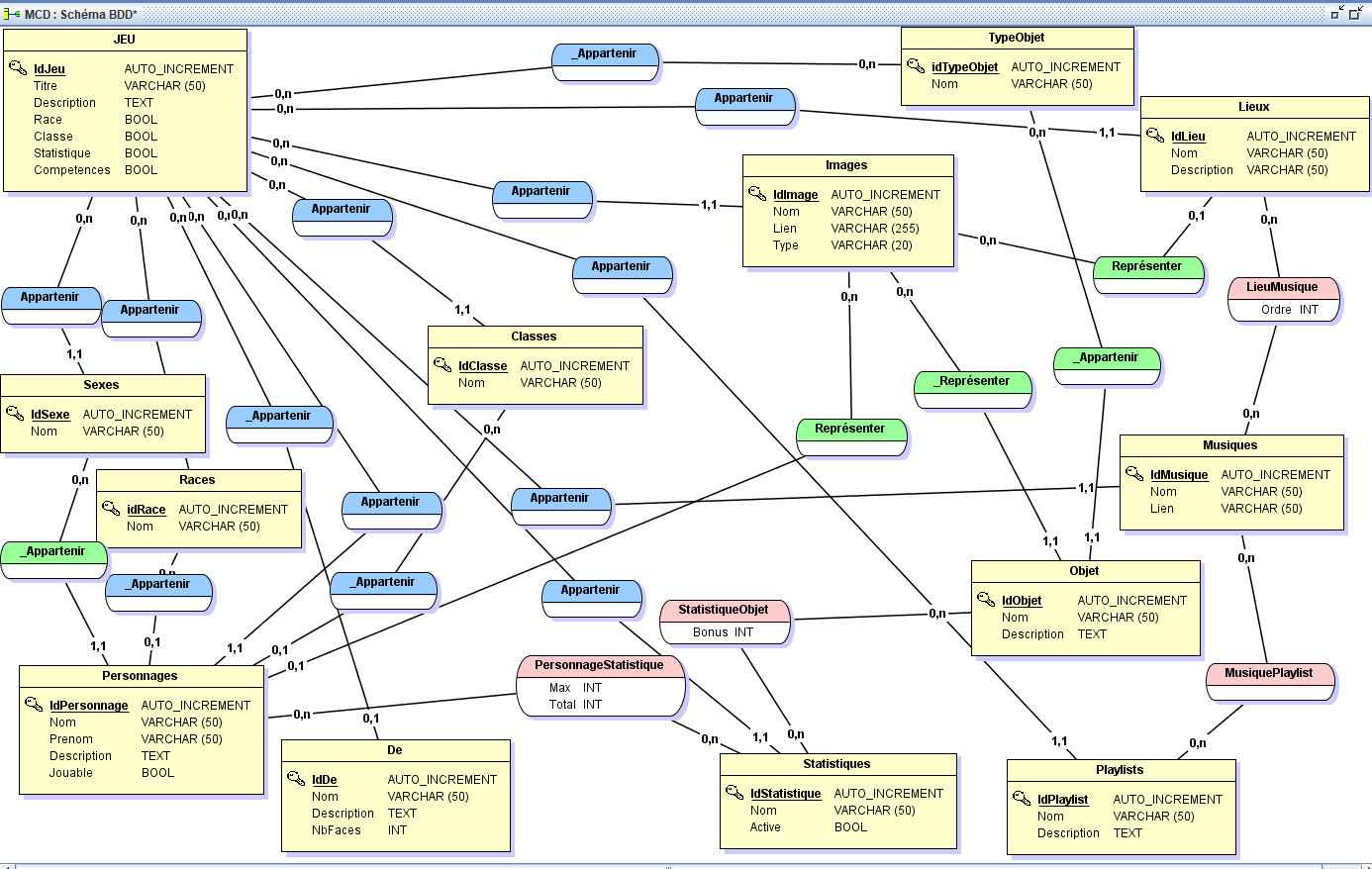
**RolistMaker est un éditeur de JDR papier jouable sur ordinateur. Il permet de créer son JEU et d’y créer son univers et ses règles. L’administrateur aura une palette d’outils variés à sa disposition pour choisir quelle direction prendra son univers (Médieval / fantastique, polar, thriller,…) ainsi que les détails.**

Ce document liste les différents aspects techniques et fonctionnalités attendues dans l’application « ADMIN » de RolistMaker

Des annotations en rouge peuvent apparaître pour des modifications à venir ou des pistes de réflexions à creuser.

# Base de données

*MCD conçu avec JMERISE.*

**Signification des couleurs :**

Bleu : Les liens basiques entre l’entité « JEU » ou « GAME »

Toutes les entités sont reliées au JEU sauf contre-indication dans la documentation

Vert : Les liens entre deux entités autres que « JEU »

Rose : Les tables associatives.

## Description des entités :

Clé Primaire

Clé étrangère

Jeu : L’entité principales qui sera créée en tout premier lors de la création de jeu

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUT** | **DESCRIPTION** | **TYPE** | **OBLIGATOIRE** |
| Titre | Titre du jeu | 50 caractères | OUI |
| Description | Sert de descriptif au jeu. Visible par tous les joueurs | TEXT, pas de limite | NON |
| Race | Indique la présence ou non de race différentes dans le jeu | Boolean | OUI |
| Classe | Indique la présence ou non de classes de personnages différentes dans le jeu | Boolean | OUI |
| Statistique | Indique la présence ou non de statistiques dans le jeu | Boolean | OUI |
| Compétences | Indique la présence ou non de compétences dans le jeu | Boolean | OUI |

Remarques : Les champs Race / Classe/ Statistique / Compétences ouvrent des menus de créations possibles dans la partie ADMIN s’ils sont activés.

**Classes**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUT** | **DESCRIPTION** | **TYPE** | **OBLIGATOIRE** |
| idClasse | Identifiant | INT | OUI |
| Nom | Nom de la classe | 50 caractères | OUI |
| idGame | L’identifiant du jeu auquel l’entité est rattachée | INT | OUI |

**Race**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUT** | **DESCRIPTION** | **TYPE** | **OBLIGATOIRE** |
| idRace | Identifiant | INT | OUI |
| Nom | Nom de la race | 50 caractères | OUI |
| idGame | L’identifiant du jeu auquel l’entité est rattachée | INT | OUI |

**Statistiques**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUT** | **DESCRIPTION** | **TYPE** | **OBLIGATOIRE** |
| id Statistiques | Identifiant | INT | OUI |
| Nom | Nom de la stat | 50 caractères | OUI |
| Active | Définit si une statistique peut évoluer sur du très court terme (Active) ou si elle représente une caractéristique du personnage qui évoluera très peu, voire pas du tout.  Ex : 12 points de vie, 4 points de magie 🡺 Active : TRUE  Ex : Physique, mentale, intelligence,… 🡺 Passive : FALSE | Boolean | OUI |
| idGame | L’identifiant du jeu auquel l’entité est rattachée | INT | OUI |

**Sexe**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUT** | **DESCRIPTION** | **TYPE** | **OBLIGATOIRE** |
| idSexe | Identifiant | INT | OUI |
| Nom | Nom du sexe | 50 caractères | OUI |
| idGame | L’identifiant du jeu auquel l’entité est rattachée | INT | OUI |

**TypeObjet**

*Les types objets permettent de classer les objets. Ex : « Arme », « arc », « arc long »*

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUT** | **DESCRIPTION** | **TYPE** | **OBLIGATOIRE** |
| idTypeObjet | Identifiant | INT | OUI |
| Nom | Nom du type d’objet | 50 caractères | OUI |
| idGame | L’identifiant du jeu auquel l’entité est rattachée | INT | OUI |

**Objet**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUT** | **DESCRIPTION** | **TYPE** | **OBLIGATOIRE** |
| idObjet | Identifiant | INT | OUI |
| Nom | Nom de l’objet | 50 caractères | OUI |
| Description | Description de l’objet | Text | NON |
| idTypeObjet | L’identifiant du type d’objet auquel l’objet est rattaché. Il est unique | INT | OUI |
| idGame | L’identifiant du jeu auquel l’entité est rattachée | INT | OUI |

**Personnage**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUT** | **DESCRIPTION** | **TYPE** | **OBLIGATOIRE** |
| idPersonnage | Identifiant | INT | OUI |
| Nom | Nom du personnage  (Non obligatoire pour des personnages *mystérieux*) | 50 caractères | NON |
| Prenom | Prénom du personnage  (Obligatoire pour des personnages *mystérieux*) | 50 caractères | OUI |
| Jouable | Activer si le personnage est jouable | Boolean | OUI |
| idGame | L’identifiant du jeu auquel l’entité est rattachée | INT | NON |
| idRace | L’identifiant de race auquel l’entité est rattachée (si activée) | INT | NON |
| idClasse | L’identifiant de classe auquel l’entité est rattachée (si activée) | INT | NON |
| idSexe | L’identifiant du sexe auquel l’entité est rattachée (si activée) | INT | NON |

Remarque : Les personnages Joueurs et Non Joueurs seront enregistrés dans cette table

**Lieux**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUT** | **DESCRIPTION** | **TYPE** | **OBLIGATOIRE** |
| idLieu | Identifiant | INT | OUI |
| Nom | Nom du lieu | 50 caractères | OUI |
| Description | Description du lieu | TEXT | NON |
| idImage | L’identifiant de l’image « Représentation » qui s’affichera lorsque les joueurs seront dans ce lieu | INT | OUI |
| idGame | L’identifiant du jeu auquel l’entité est rattachée | INT | OUI |

/ ! \ Ajouter une playlist de lieux pour pouvoir naviguer plus facilement, voir une carte du monde ?

**Musiques**

Jeter un œil aux APIs Youtube / deezer /spotify pour voir si la musique n’est pas utilisable directement dans le logiciel pour éviter un surcout de mémoire

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUT** | **DESCRIPTION** | **TYPE** | **OBLIGATOIRE** |
| idMusique | Identifiant | INT | OUI |
| Nom | Nom de la musique telle qu’elle apparait dans la sélection  Ex : A one winged angel 🡪 musique\_boss\_1 | 50 caractères | OUI |
| Lien | Lien où est stockée la musque dans les dossiers du jeu  / ! \ Un blob storage ou une conversion en bytes[] ne serait pas mieux  et moins couteux en mémoire ? | 255 caractères | OUI |
| idGame | L’identifiant du jeu auquel l’entité est rattachée | INT | OUI |

**Playlist**

Une playlist est une bibliothèque de musiques où l’on peut piocher pour agrémenter les lieux visités.

On ne génère pas une playlist pour un lieu mais bien comme une réserve où se servir.

Exemple : Une playlist *« combats »* peut contenir 50 musiques de ce genre mais un **Lieu** n’en consommera que certaines choisies. Voir d’autres musiques d’autres playlists.

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUT** | **DESCRIPTION** | **TYPE** | **OBLIGATOIRE** |
| idPlaylist | Identifiant | INT | OUI |
| Nom | Nom de la playlist | 50 caractères | OUI |
| Description | Description de la playlist | TEXT | NON |
| idGame | L’identifiant du jeu auquel l’entité est rattachée | INT | OUI |

**Image**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUT** | **DESCRIPTION** | **TYPE** | **OBLIGATOIRE** |
| idImage | Identifiant | INT | OUI |
| Lien | Lien où est stockée l’image dans les dossiers du jeu  / ! \ Un blob storage ou une conversion en bytes[] ne serait pas mieux ? | 255 caractères | OUI |
| Nom | Nom de l’image | 50 caractères | OUI |
| TypeImage | Un enum qui correspond à l’image :   * Icone = Pour les objets * Portrait = pour les personnages * Representation = pour les lieux | Enum | OUI |
| idGame | L’identifiant du jeu auquel l’entité est rattachée | INT | OUI |

**Dé**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUT** | **DESCRIPTION** | **TYPE** | **OBLIGATOIRE** |
| idDé | Identifiant | INT | OUI |
| Nom | Nom du dé | 50 caractères | OUI |
| Description | Description du dé | TEXT | NON |
| NbFaces | Nombre de face du dé  Ajouter une table « FACES » pour créer des dés personnalisés ? | INT | OUI |
| idGame | L’identifiant du jeu auquel l’entité est rattachée | INT | OUI |

## Tables associatives

**MusiquePlaylist**

|  |  |  |  |
| --- | --- | --- | --- |
| idMusique | Identifiant de la musique | INT | OUI |
| idPlaylist | Identifiant de la playlist | INT | OUI |

**LieuMusique**

|  |  |  |  |
| --- | --- | --- | --- |
| idLieu | Identifiant du lieu | INT | OUI |
| idMusique | Identifiant de la musique | INT | OUI |
| Ordre | Ordonne la diffusion des musiques sur le lieu. | INT | OUI |

**StatistiqueObjet**

Table reliant les statistiques aux objets. Les objets ayant une ou plusieurs stats ajoutées y sont consignés.

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUT** | **DESCRIPTION** | **TYPE** | **OBLIGATOIRE** |
| idStatistique | Identifiant de la statistique | INT | OUI |
| idObjet | Identifiant de l’objet | 50 caractères | OUI |
| Bonus | Valeur apportée par l’objet | INT | OUI |

**StatistiquePersonnage**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUT** | **DESCRIPTION** | **TYPE** | **OBLIGATOIRE** |
| idStatistique | Identifiant de la statistique | INT | OUI |
| idPersonnage | Identifiant du personnage | INT | OUI |
| Max | Le Max correspond à la valeur maximum que peut prendre une statistique active  (Ex : points de vie = 12)  S'il s'agit d'une stat passive, c'est la valeur qui sera retenue au jet de dé  (Ex : sur un jet où une valeur supérieure au dé est un succès. Si la stat passive testée est de 40 et que le dé fait 35, c’est une réussite) | INT | OUI |
| Total | Il s'agit de la valeur qu'une stat prend à l'instant T  (Ex : 6 points de vie sur 12) | INT | OUI |

# Environnement technique

Cette partie a été conçue lorsque que le développement se faisait en Winform. Il faut prendre en compte des modifications drastiques à venir.

**Rolist Maker – Admin :**

Application découpée en modèle 3 tiers.

Une couche DAL qui fait les opérations avec la base de données

Une couche BLL qui effectue les traitements et contrôles. Elle fait appel à la couche DAL

Une couche GUI qui affiche les interfaces. Elle fait appel à la couche métier BLL.

L’application en développée en C# . NET avec le pattern Singleton\* sous Visual Studio.

La bases de données est en SQLite et utilise les ORM Entity Framework Core.

Les appels à la base de données se font essentiellement avec LINQ.

\*Etant une première expérience avec Entity Framework, il se peut que le singleton ait été mal utilisé lors de la première version de l’interface de l’application. La deuxième version en WPF le corrigera.

**Rolist Maker – MJ / Rolist Maker - Player**

**Application en cours de réflexion.**

Privilégier une application en web peut-être pour favoriser les parties distantes. SignalR peut apporter les appels en BDD importants.

* Problèmes d’une application, le nombre de fenêtre qui doit être accessible

Une application client lourd ? Plus pratiques pour la multiplicité des fenêtres mais pour faire communiquer les ordinateurs entre eux…

Dossier partagé ? Dropbox ?

Appels FTP ou web service ?

Mettre un site en place qui hébergerait la base de données de la partie qui recevrait les appels ?

# Fonctionnalités attendues

## Prioritaires

* La première fenêtre visible doit être une fenêtre de sélection de jeu avec la possibilité de CUD (**Creation, Update, Delete**) ainsi qu’une copie de l’ensemble d’un jeu.
* Mettre au point une installation facile avec génération de la base de données 🡺 Formation sur Entity framework à approfondir.
* La grille de sélection doit afficher le titre et la description du jeu.
* Lors de la conversion des menus en WPF, afficher soit la description du jeu en « *CharacterEllipsis* » soit en détails dans un détail de ligne.
* Les menus de créations ne doivent pas être accessibles tant que l’entité « JEU » n’a pas été créée.
* Les menus « Compétence », « Statistiques », « Race » et « Classe » ne sont accessibles que selon le paramétrage du « JEU ».
* Par défaut à la création d’un jeu, les sexes « Homme », « Femme », « Non binaire » sont ajoutés mais peuvent être supprimés.
* Les *enum* d’images (portait, icone doivent être dans enregistrés dans le code avant de commencer le menu « Image »
* Les objets ne peuvent être créés tant qu’il n’existe pas au moins un « TypeObjet » (*ex : potion, épée, …)*
* Les playlists ne peuvent être générées tant qu’aucune musique n’a été ajoutée.

## Non-prioritaires mais nécessaires

* Une musique doit pouvoir être écoutée dans les menus de sélection de musique. Seul les boutons « play », « pause », « stop », « next », « previous » ainsi qu’une barre de défilement et le timer sont nécessaire.
* Le menu d’ajout d’images doit être accessible rapidement depuis les menus qui en utilisent (Objet, Personnage, Lieu)
* Définir des tailles d’images pour chaque type.
* Créer un menu de redimensionnement des images pour correspondre aux tailles

## Fonctionnalités de confort

* Créer un menu « Historia » pour rédiger une histoire et chapitrer le jeu qui serait consultable pour les joueurs.
* Rendre une intégration par Office Word possible.
* Donner la possibilité de rendre accessible seulement certaines parties selon les joueurs
* Peupler un lieu avec des personnages non joueurs.
* Mettre certaines options possibles sur un lieu comme une boutique ou des personnages ayant des lignes de dialogues pour que les joueurs puissent effectuer des actions « hors-partie »