 Développement d'une application web de gestion de jeux de rôle destinée aux "Maître de jeu". L'application permettra de générer des fiches de personnages génériques et numériques de façon semi-automatique avec la possibilité d'utiliser l'intelligence artificielle. Ces fiches posséderont des caractéristiques et pourront être attribuées à des scénarios de l'utilisateur. L'application sera développée avec le Framework Next.js qui utilise la librairie React.js

Travail de Bachelor réalisé en vue de l’obtention du Bachelor HES

par :

Enzo Jolidon Marte

Conseiller au travail de Bachelor :

James Keller

Genève, le 26 août 2024

Haute École de Gestion de Genève (HEG-GE)

Filière Informatique de Gestion

Déclaration

Ce travail de Bachelor est réalisé dans le cadre de l’examen final de la Haute école de gestion de Genève, en vue de l’obtention du titre Bachelor of Science HES-SO en Informatique de gestion.

L’étudiant a envoyé ce document par email à l'adresse remise par son directeur de mémoire afin qu’il l’analyse à l’aide du logiciel de détection de plagiat COMPILATIO.

L’étudiant accepte, le cas échéant, la clause de confidentialité. L'utilisation des conclusions et recommandations formulées dans le travail de Bachelor, sans préjuger de leur valeur, n'engage ni la responsabilité de l'auteur, ni celle du conseiller au travail de Bachelor, du juré et de la HEG.

« J’atteste avoir réalisé seul le présent travail, sans avoir utilisé des sources autres que celles citées dans la bibliographie. »

Fait à Genève le 26 août 2024

Enzo Jolidon Marte

< Signez la déclaration ici >

Remerciements

Si vous avez des remerciements à formuler, à l’entreprise ou à toute autre personne qui a pu vous aider dans la réalisation du travail.

Les remerciements sont rédigés dans le style « **Corps de texte** ».

Résumé

Le résumé ne doit pas dépasser une page. Il est rédigé dans le style « **Corps de texte** ».

Table des matières

[Déclaration i](#_Toc170465910)

[Remerciements ii](#_Toc170465911)

[Résumé iii](#_Toc170465912)

[Liste des tableaux v](#_Toc170465913)

[Liste des figures v](#_Toc170465914)

[1. Introduction 1](#_Toc170465915)

[1.1 Le jeu de rôle 1](#_Toc170465916)

[1.2 Maître du jeu 1](#_Toc170465917)

[1.3 Fiche de personnage 1](#_Toc170465918)

[2. Etat de l’art 2](#_Toc170465919)

[2.1 Noobliees Chronique 2](#_Toc170465920)

[2.2 CO-DRS 2](#_Toc170465921)

[2.3 Dyslexic Character Sheets 2](#_Toc170465922)

[3. Technologie 3](#_Toc170465923)

[3.1 Frontend 3](#_Toc170465924)

[3.2 Backend 3](#_Toc170465925)

[3.3 Base de données 3](#_Toc170465926)

[3.4 Déploiement 3](#_Toc170465927)

[3.5 Versioning 3](#_Toc170465928)

[4. Fonctionnalités 4](#_Toc170465929)

[4.1 Scénario 4](#_Toc170465930)

[4.2 Personnage 4](#_Toc170465931)

[4.3 Arme 4](#_Toc170465932)

[Bibliographie 5](#_Toc170465933)

[Annexe 1 : Titre de l’annexe 6](#_Toc170465934)

Liste des tableaux

[Tableau 1 : Titre du tableau **Erreur ! Signet non défini.**](#_Toc381707012)

Liste des figures

[Figure 1 : Titre de la figure **Erreur ! Signet non défini.**](file:///Y:\BODER\1-Cours\160-travail_de_bachelor\Modeles_TDB_HEG_2014.dotx#_Toc381707014)

# Introduction

## Le jeu de rôle

Le jeu de rôle, souvent abrégé en JDR, est une activité ludique et interactive où les participants incarnent des personnages fictifs dans un cadre narratif défini par un maître du jeu. Cette pratique, née avec des jeux emblématiques tels que Donjons et Dragons, a évolué pour englober une vaste gamme de genres, d’univers et de systèmes de règles.

## Maître du jeu

Le maître du jeu (MJ), également connu sous le nom de Game Master (GM), occupe un rôle central et complexe dans le jeu de rôle, en étant responsable de la création et de la gestion de l'univers de jeu. L'une des principales difficultés auxquelles le MJ est confronté réside dans l'imprévisibilité des actions des joueurs.

Chaque décision et chaque interaction peuvent dévier de manière inattendue le déroulement de l'histoire, exigeant du MJ une capacité exceptionnelle à improviser et à adapter le scénario en temps réel. De plus, le MJ doit concevoir et animer une multitude de personnages non-joueurs (PNJ), chacun doté de sa propre personnalité, de ses motivations et de ses objectifs.

Cette création massive de PNJ requiert non seulement une grande créativité, mais aussi une cohérence et une profondeur qui enrichissent l'univers de jeu, rendant l'expérience immersive pour les joueurs. Le MJ doit ainsi jongler avec une multitude de fils narratifs, anticiper les actions des joueurs et maintenir un équilibre entre structure et flexibilité, ce qui fait de ce rôle un véritable défi intellectuel et artistique.

## Fiche de personnage

Afin d’avoir un point de référence tout au long du JDR, les fiches de personnage sont essentielles pour le MJ pour suivre l’état, se souvenir des différentes capacités et en garder en tête les traits de caractère. Suivant les univers dans lesquels se déroule les scénarios, les fiches peuvent variées quant aux champs nécessaires à remplir en adéquation avec les besoins du MJ. En fonction de la longueur et/ou des événements du scénario, il en devient chronophage de créer des dizaines de personnage.

# Etat de l’art

Aujourd’hui (1er juin), il existe diverses applications qui permettent d’obtenir des fiches préremplies de façon aléatoire ou non. Elles possèdent chacune leurs avantages et leurs inconvénients.

## Noobliees Chronique

« Noobliees chronique » est un site français qui possèdent une fonctionnalité de génération de fiche de personnage. Il est possible de parcourir différente section de la fiche mais il n’est parfois pas possible de les remplir aléatoirement. De plus, après avoir parcouru tous les champs, le rendu de la fiche n’est pas fait et donc nous n’obtenons rien de réutilisable.

## CO-DRS

« CO-DRS » est un site qui propose des fiches de personnage interactive. Il offre la possibilité de créer, d’éditer et effectuer des actions sur la fiche. Il également possible de les partager à d’autre utilisateur, comme des joueurs, afin qu’il puisse également suivre l’avancée de leur personnage. Le point le plus freinant est qu’il n’est possible d’avoir uniquement 5 fiche active pour des raisons de stockage. Cette limitation peut être frustrante lorsqu’on a un JDR avec beaucoup de joueur ou un nombre important de personnage non-joueur (PNJ).

## Dyslexic Character Sheets

« Dyslexic Character Sheets » offre des fiches de personnage extrêmement complète, personnalisable et facile d’extraction. Toutefois, ce processus est plutôt long pour créer ne serait-ce qu’un personnage. De plus, les fiches sont spécifiques aux univers en question. De ce fait, on se retrouve bloqué si notre JDR ne se déroule pas dans l’univers proposé par le site. Ajouté à cela, le fait que les fiches sont immuables et ne peuvent donc pas être modifiée.

# Technologie

## Frontend

Afin de développer cette application, il faut choisir parmi les différentes librairie Javascript à disposition. Dans les plus populaire, on retrouve React.js maintenu par le groupe Meta, Vue.js et Angular maintenu par Google et qui a apporté des changements significatifs depuis la version 16. Toutes ces librairies et Framework sont unique mais restent sur la même base à savoir Javascript. De ce fait, elles sont toutes suffisamment qualifié pour produire cette application. Il est possible d’arriver au même résultat indépendamment de la librairie choisie.

Après comparaison des évolutions de ces différentes opportunité, et de la popularité de chacun, j’ai décidé de porter mon choix sur la librairie React.js. Ce choix s’exprime d’une affection particulière dans la façon dont la communauté soutient cette librairie et des différents outils émergents qui propose très souvent des intégrations avec React. Aussi, cette librairie est actuellement sur une pente ascendante, ce qui annonce une bonne prospérité pour le projet.

## Backend

Afin d’alimenter la partie applicative, il est nécessaire de mettre en place une architecture backend complète et fiable. A l’instar du frontend, plusieurs choix s’offre à nous pour développer le côté serveur de l’application. Différent langage et Framework mature comme Django (Python), Ruby on Rails (Ruby) et Laravel (PHP) sont capable de fournir les outils nécessaires.

Cependant, afin de rester sur une perspective et pour maximiser l’intégration avec React, j’ai choisi d’opter pour Next.js. Next.js est un Framework développé et maintenu par Vercel et qui se base sur React. Tout comme ce dernier Next.js utilise le Javascript et met à disposition une multitude d’outils pour alimenter l’application en donnée, définir des « endpoint API », gérer la navigation, etc. La plus grande plus-value qu’il apporte est les « React Server Component ».

### Server Side Rendering

Le **rendu côté serveur (SSR)** est une technique où le contenu de l'application est rendu sur le serveur avant d'être envoyé au client. Cela contraste avec le rendu côté client (Client-Side Rendering ou CSR) où le contenu est principalement généré par JavaScript exécuté dans le navigateur. SSR améliore le temps de chargement initial et le référencement (SEO) en fournissant une page complètement rendue au navigateur dès la première requête. Cela permet d'obtenir des données dynamiques sur le serveur avant de rendre la page, ce qui est particulièrement utile pour les applications nécessitant des données à jour dès le chargement initial.

### React Server Component

Contrairement aux composants classiques de React, les « React Server Components » sont exécutés sur le serveur. Cela signifie que le code de ces composants n'est pas envoyé au client, ce qui réduit la quantité de JavaScript que le navigateur doit télécharger et exécuter. En pratique, cela permet de rendre les parties statiques ou prévisibles de l'interface utilisateur plus rapidement et de manière plus efficace.

## Base de données

Enfin, pour que l’application persiste dans le temps, une base de données fiable et durable sera indispensable. Au vu des schémas de données, il n’y pas de place au doute. Une base de données relationnel (SQL) semble être le choix le plus approprié.

Afin de s’adapter au divers service existent sur le marché, j’utiliserais une base de données PostgreSQL hébergé par le Saas « Neon Tech ». Ce service est reconnu et approuvé par différent organisme notamment Vercel.

## Déploiement

En ce qui concerne l’hébergement de l’application, Vercel propose des services d’hébergement pour les applications Next.js. Ce service est le choix idéal pour déployer l’application en production est avoir un service fiable, disponible et sans engager de coût.

D’autre service plus réputé comme AWS, Infomaniak, Hostinger et autre Saas aurais également pu être de bonne option si un budget avait été mis à disposition pour le déploiement de l’application.

## Versioning

Enfin, pour développer l’application et implémenté les fonctionnalités au fur et à mesure, j’utiliserais Git qui permettra de gérer les états de fichier (Modifié, ajouté, supprimé, non traqué) ainsi que de garder un historique des versions pour voir l’évolution et de revenir en arrière.

Il permettra aussi de séparer différent état de l’application afin de développé différente fonctionnalité sans en entraver d’autre encore en développement. Cette façon de procédé étant plus approprié dans un projet qui engage plusieurs développeurs, elle peut néanmoins avoir certains aspect pratique.

### Github

Pour sauvegarder ce développement et pouvoir le partager, un espace sur Github sera en place. Il permettra d’avoir un graphique de l’ensemble du projet. Un espace sur Gihlab aurait également pu faire l’affaire mais les points clés de cet outil comme la CI/CD pour l’automatisation n’avaient pas de plus-value pour le projet.

# Fonctionnalités

## Authentification

Les utilisateurs de cette application seront dans l’obligation de se connecter. Après étude de la meilleure façon d’attribué de façon privé les différentes données créer par l’utilisateur, un système d’authentification est la solution la plus judicieuse. Pour garder une application simple dans son fonctionnement et utilisé des méthodes récentes et sécurisées, j’implémenterais une fonctionnalité de Single-Sign on (SSO) : OpenIDConnect. Cette méthode permet limiter l’infrastructure de base de données à mettre en place tout en gardant des authentifications sécurisé et fédéré par des services reconnu comme Google, Github, Notion, etc.

### OAuth

## Scénario

Les scénarios seront les objets racines de la bibliothèque de données de l’utilisateurs. Ils agiront comme conteneur des personnages. Ils serviront de point de repère et pour les utilisateurs afin qu’ils puissent s’y retrouver dans leur gestion de différent JDR.

### Création

Depuis la page d’accueil, les utilisateurs pourront et devront créer leur scénario au travers d’un formulaire simple et accessible avec 3 champs (nom, univers, description). Une fois créé, ils apparaîtront sur cette même page et pourront être accéder plus en détail pour la création de personnage au moyen d’un clic.

### Suppression

Un bouton sera à disposition pour supprimer le scénario si nécessaire. A noter que si les personnages présents ne sont pas réutilisés dans d’autre scénario alors ils seront, eux aussi, complètement supprimés.

## Personnage

Les personnages sont les données qu’on cherche en particulier à gérer. De ce fait, ils intègrent un peu plus de fonctionnalité que les autres aspects de l’application. Ils pourront être crée de façon manuelle, en remplissant tous les champs à la main, de façon semi-automatique, en laissant l’aléatoire décidé de certains paramètres, ou complétement aléatoire avec l’aide de l’intelligence artificielle.

### Création

Une fois sur la page de détail d’un scénario l’utilisateur pourra accéder à l’interface de création d’un personnage pour ce même scénario. A partir de ce point, il sera amené à rentrer les champs obligatoires pour pouvoir soumettre le formulaire et créer son personnage. Cette option laisse tout le soin à l’utilisateur de construire un personnage qu’il juge adapté à son JDR.

Afin de répondre à un de besoin principal des utilisateurs, qui est de pouvoir créer des PNJ rapidement, certains champs disposent de bouton qui permettent d’inscrire une valeur aléatoire.

### Modification

Pour persister le personnage dans le temps et le faire évoluer, il est toujours possible de modifier certaines caractéristiques de ce dernier. Au travers des différentes aventure, il se peut que le personnage change et, s’il est réutilisé dans un autre scénario, on souhaite garder sa fiche comme elle devrait l’être. C’est pourquoi il est indispensable de pouvoir avoir la possibilité de le modifié en tout temps.

### Importation

Comme mentionné précédemment, après des aventures, les MJs pourraient avoir envie de réintégré un personnage dans une nouveau scénario (suite d’aventure, référence, PNJ). Pour ce faire, les utilisateurs auront la possibilité de pouvoir importer des personnages déjà existant dans d’autre scénario. Ces personnages de sont pas de copie conforme. Il s’agit de la même fiche, ce qui veut dire que tout changement apporté à cette dernière se verront répercuté autant pour un scénario qu’un autre.

### Suppression

Une page sera dédiée aux personnages créer par l’utilisateur. Elle a pour objectif d’offrir une vue d’ensemble sur eux afin de s’y retrouver. C’est également depuis cette page que l’utilisateur pourra faire du tri dans ses personnages s’il décide de vouloir en retirer pour diverses raisons.

## Arme

Les armes sont une composante complémentaire aux personnages. Elles s’attribuent uniquement à ces derniers. Le but est de pouvoir garder la référence de l’équipement directement sur eux. A l’image de certains attributs que le personnage possède, les armes disposent elles aussi de leur attribut propre.

### Création

A la différence des personnages, les armes pourront être crée dans une page qui englobe la création et la visualisation d’attribut que j’aborderais dans le point suivant. Ces armes pourront donc être crée dans une page de gestion au travers d’un formulaire similaire à celui des personnages.

### Suppression

Depuis la même vue, il est possible de supprimer ces mes armes à l’aide d’un bouton. Les armes sont des éléments propres à l’utilisateurs qui les a créés tout comme les scénarios et personnages.

## Attribut d’arme/de personnage

Enfin, les attributs d’arme et/ou de personnage sont les seules variables communes aux différents utilisateurs. En effet, afin d’avoir les mêmes bases sur ce qu’il est disponible de créer, les attributs sont globaux. Ils ne peuvent être créé que par les utilisateurs disposant des droits suffisants pour exécuter cette action. Néanmoins, même si la majorité des utilisateurs ne seront pas habilité à pouvoir créer des attributs, ils peuvent tout de même remplir le même formulaire et suggérer un attribut.

### Création

Depuis la même page dans laquelle il est possible de créer des armes, les MJ pourront créer et visualiser les attributs. Certains attributs ont la possibilité d’embarquer une description avec eux. Cette description sera utile sur la fiche de personnage pour avoir rapidement des informations complémentaires lié à l’attribut.

### Suggestion

Pour les utilisateurs qui ne dispose pas de droit suffisant pour créer des attributs, ils peuvent tout de même soumettre leur formulaire ce qui créera une suggestion. Ces suggestions apparaîtront dans une page dédiée uniquement par les utilisateurs disposant de droit privilégié. Ainsi ils pourront visualiser les attributs et jugé par eux même s’il est intéressant de les ajoutées à la bibliothèque d’attribut.

# Bibliographie

Le style de paragraphe appliqué dans les références doit être « **références bibliographiques** » : Arial 11 points, interligne simple, justifié, espace en dessus 6 pt, espace en dessous 6 pt. Voici un exemple :

FOREST, David, 2011. *Droit des données personnelles*. Paris : Gualino, 2011. Droit en action. ISBN 9782297015028.

Annexe 1 : Titre de l’annexe

Votre texte ou le document mis en annexe.

Évitez de mettre deux annexes sur la même page. Faites plutôt un saut de page puis indiquez le numéro de la nouvelle annexe et son titre.