 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

Pour commencer, je tiens à remercier tout particulièrement M. Antoine Barone ainsi que M. Matteo Schudel, qui ont été les mandants de cette application, pour leur disponibilité ainsi que leur soutien et leur idée qui m’ont permis de réaliser ce travail et sans qui tout cela n’aurait pas été possible.

Un grand merci à l’équipe de développement de E-xpert Solutions SA, qui même sans le savoir, m’ont aidé dans le développement de cette application de par les compétences que j’ai acquises et que je continuerais d’acquérir.

Mes remerciements vont également à Mme Iris Hughes, sans qui je n’aurais pas pu rendre un travail aussi qualitatif que celui-ci. Elle m’aura permis de prendre du recul sur certains aspects techniques du document.

Je remercie également M. James Keller pour l’intérêt qu’il a témoigné à mon sujet et pour son encouragement qui m’a poussé à aller plus loin que mes idées.

Enfin, mille merci à toutes les personnes qui m’ont apporté du soutien tout au long de mon travail, de nouvelles idées et pistes à explorer ou de leur participation de l’amélioration de ce travail.

Résumé

La gestion de session de jeu dans les jeux de rôle (JDR) peut relever de différents défis que ce soit dans la préparation ou le déroulement de ce dernier. À commencer par l’écriture du scénario jusqu’à la fin de la session de jeu, en passant par la création de personnage, l’attribution de compétence et gestion des décors.

Les maîtres du jeu (MJ) peuvent se heurter à diverses problématiques, qu'elles soient réflexives ou procédurales, où l'informatique pourrait sans doute alléger leur charge de travail. Ainsi, la création d'une application web moderne utilisant des technologies innovantes permettrait de gérer certains aspects d'un jeu de rôle (JDR) de manière plus efficace.

Table des matières

[Déclaration i](#_Toc175334396)

[Remerciements ii](#_Toc175334397)

[Résumé iii](#_Toc175334398)

[Table des matières iv](#_Toc175334399)

[Liste des figures vi](#_Toc175334400)

[Glossaire des termes 1](#_Toc175334401)

[1. Introduction 2](#_Toc175334402)

[1.1 Le jeu de rôle 2](#_Toc175334403)

[1.2 Maître du jeu 2](#_Toc175334404)

[1.3 Fiche de personnage 2](#_Toc175334405)

[2. Etat de l’art 3](#_Toc175334406)

[2.1 Noobliees Chronique 3](#_Toc175334407)

[2.2 CO-DRS 3](#_Toc175334408)

[2.3 Dyslexic Character Sheets 3](#_Toc175334409)

[2.4 Conclusion 3](#_Toc175334410)

[3. Technologie 4](#_Toc175334411)

[3.1 Frontend 4](#_Toc175334412)

[3.2 Backend 4](#_Toc175334413)

[3.2.1 Server Side Rendering 5](#_Toc175334414)

[3.2.2 React Server Component 5](#_Toc175334415)

[3.3 Base de données 5](#_Toc175334416)

[3.4 Déploiement 5](#_Toc175334417)

[3.5 Versioning 5](#_Toc175334418)

[3.5.1 Github 6](#_Toc175334419)

[4. Fonctionnalités 7](#_Toc175334420)

[4.1 Authentification 7](#_Toc175334421)

[4.1.1 OAuth 7](#_Toc175334422)

[4.2 Scénario 7](#_Toc175334423)

[4.2.1 Création 7](#_Toc175334424)

[4.2.2 Suppression 7](#_Toc175334425)

[4.3 Personnage 8](#_Toc175334426)

[4.3.1 Création 8](#_Toc175334427)

[4.3.2 Modification 8](#_Toc175334428)

[4.3.3 Importation 8](#_Toc175334429)

[4.3.4 Suppression 9](#_Toc175334430)

[4.4 Arme 9](#_Toc175334431)

[4.4.1 Création 9](#_Toc175334432)

[4.4.2 Suppression 9](#_Toc175334433)

[4.5 Attribut d’arme/de personnage 9](#_Toc175334434)

[4.5.1 Création 9](#_Toc175334435)

[4.5.2 Suggestion 10](#_Toc175334436)

[5. Intelligence artificielle 11](#_Toc175334437)

[6. Sécurité 13](#_Toc175334438)

[6.1 Authentification 13](#_Toc175334439)

[6.2 API 13](#_Toc175334440)

[6.2.1 Interface REST 14](#_Toc175334441)

[6.2.2 Server Action 14](#_Toc175334442)

[6.2.3 Inconvénients 15](#_Toc175334443)

[6.2.4 Problématique 15](#_Toc175334444)

[6.2.5 Solutions 16](#_Toc175334445)

[7. Avenir de l’application 17](#_Toc175334446)

[7.1 Spécialisation des fiches 17](#_Toc175334447)

[7.2 Software as a Service 17](#_Toc175334448)

[7.3 Interface graphique (shared room) 17](#_Toc175334449)

[8. Conclusion 18](#_Toc175334450)

[Bibliographie 19](#_Toc175334451)

[Annexe 1 : MindMap 20](#_Toc175334452)

[Annexe 2 : Modèle de base de données 21](#_Toc175334453)

Liste des figures

Figure 1 Prompt 11

Glossaire des termes

G1. : JDR Jeu de rôle

G2. : MJ Maître du jeu

G3. : GM Game Master

G4. : PNJ Personnage non-joueur

G5. : SSR Server Side Rendering

G6. : CSR Client Side Rendering

G7. : SSO Single Sign On

G8. : API Application Platform Interface

G9. : REST Representational State Transfer

G10. : SaaS Software as a Service

G11. : IA Intelligence artificielle

G12 : SEO Search Engine Optimization

# Introduction

## Le jeu de rôle

Le jeu de rôle papier, souvent abrégé 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 dans les années 1980-1990 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. (Wikipédia 2024)

## 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, en étant responsable de la création et de la gestion de l'univers de jeu ainsi que le déroulement de l’histoire. 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 d’improvisation et d’adaptation du 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 et les joueurs afin de suivre l’état, se souvenir des différentes capacités et garder en tête les traits de caractère des différents personnages de l’histoire. Suivant l’univers de jeu et le système de règle utilisé, les fiches peuvent varier grandement 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, d’autant plus si le système de règle requiert des fiches de personnage complexe.

# Etat de l’art

En date du 1er juin 2024, 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 différentes sections 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 affiché à l’écran et donc je n’obtiens rien de réutilisable.

## CO-DRS

« CO-DRS » est un site qui propose des fiches de personnage interactives. Il offre la possibilité de créer, d’éditer et effectuer des actions sur la fiche. Il possible également de les partager à d’autres utilisateurs, comme des joueurs, afin qu’il puisse également suivre l’avancée de leur personnage. Le plus gros inconvénient de ce site est le nombre limité de fiches actives en raison de capacité de stockage, c’est-à-dire uniquement cinq fiches actives. Cette limitation peut être frustrante lorsqu’on a un JDR avec beaucoup de joueurs ou un nombre important de personnages non-joueurs (PNJ).

## Dyslexic Character Sheets

« Dyslexic Character Sheets » offre des fiches de personnage extrêmement complètes, personnalisables et faciles d’extraction. Toutefois, ce processus est chronophage pour créer ne serait-ce qu’un personnage. De plus, les fiches sont spécifiques à un système de règle en particulier. De ce fait, on se retrouve bloqué si le JDR ne se déroule pas dans l’univers proposé par le site. Ajoutées à cela, le fait que les fiches sont immuables et ne peuvent donc pas être modifiées.

## Conclusion

En raison de ces contraintes, j'ai été mandaté pour créer une application visant à simplifier et accélérer le processus de création de fiches de personnages complètes et modulables, adaptées à tout type de jeu de rôle et réutilisable. Cette solution aura pour principal avantage de ne pas être fixe à un système de règle défini et donc de pouvoir être flexible aux autres univers en offrant un jeu agnostique.

# Technologie

## Frontend

Afin de développer cette application, il faut choisir parmi les différentes librairies Javascript à disposition. Dans les plus populaires, 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 uniques mais restent sur la même base, à savoir Javascript. De ce fait, elles sont toutes suffisamment qualifiées pour produire cette application. Il est possible d’arriver au même résultat indépendamment de la librairie choisie. Les différences les plus significatives entre ces librairies ne se situe pas forcément dans la façon dont elles ont été construites, mais plutôt dans le support qu’il y a dessus, les intégrations dans d’autres services et la courbe d’apprentissage.

En se basant sur ces connaissances, je peux dès lors me positionner sur la librairie qui me permettra de développer cette application dans le meilleur environnement tout en favorisant la longévité de la maintenance. Malgré le fait qu’elle propose une courbe d’apprentissage plutôt rude, la librairie React.js se présente comme le choix le plus adapté au cas actuel. En effet, les intégrations se font nombreuses au vu de la popularité grandissante de React.js. Ces opportunités permettent de s’adapter à la majorité des situations et représentent donc un argument de taille au sujet du choix de cette librairie.

## Backend

Afin d’alimenter la partie applicative, il est nécessaire de mettre en place une architecture backend complète et fiable. À l’instar du frontend, plusieurs choix s’offre à moi pour développer le côté serveur de l’application. Différents langages et Framework matures comme Django (Python), Ruby on Rails (Ruby) et Laravel (PHP) sont capables de fournir les outils nécessaires à l’application de ce projet.

Cependant, afin de rester sur une perspective avant-gardiste 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 données, 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 dans laquelle le contenu de l'application est rendu sur le serveur avant d'être envoyé au client. En opposition avec le rendu côté client (Client-Side Rendering ou CSR) dans lequel 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 a pas de place au doute. Une base de données relationnelles (SQL) semble être le choix le plus approprié.

Afin de s’adapter aux divers services existant sur le marché, j’utilise une base de données PostgreSQL hébergée par le Saas « Neon Tech ». Ce service est reconnu et approuvé par différents organismes, 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 et avoir un service fiable, disponible et sans engager de coût.

D’autres services plus réputés comme AWS, Infomaniak, Hostinger et autres Saas auraient également pu être de bonnes options si un budget avait été mis à disposition pour le déploiement de l’application.

## Versioning

Enfin, pour développer l’application et implémenter les fonctionnalités au fur et à mesure, j’utilise 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 revenir en arrière.

Il permettra aussi de séparer différents états de l’application afin de développer différentes fonctionnalités sans en entraver d’autres encore en développement. Cette façon de procéder étant plus appropriée dans un projet qui engage plusieurs développeurs, elle peut néanmoins avoir certains aspects pratiques.

### Github

Pour sauvegarder ce développement et pouvoir le partager, je mets en place un espace sur Github. Il permettra d’avoir une vue graphique de l’ensemble du projet. Un espace sur Gihlab aurait également pu être employé, 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 devront se connecter. Après avoir étudié la meilleure manière d'attribuer les données créées par l'utilisateur de manière privée, un système d’authentification est la solution la plus judicieuse. Pour garder une application simple dans son fonctionnement et utiliser des méthodes récentes et sécurisées, j’implémente la fonctionnalité de Single-Sign on (SSO), déjà commune dans beaucoup d’application : OpenIDConnect. Cette méthode permet de limiter l’infrastructure de base de données à mettre en place tout en gardant des authentifications sécurisées et fédérées par des services reconnus comme Google, Github, Notion, etc.

### OAuth

Nombreux sont les utilisateurs à avoir déjà utilisé cette méthode d’authentification sans forcément savoir qu’il s’agit de celle-ci. En effet, il s’agit des boutons qui proposent de s’authentifier en passant par un autre organisme. Typiquement, Digitec AG propose, lors de la création de son compte, de pouvoir s’authentifier avec son compte Google. Cette méthode permet à l’utilisateur de créer son compte rapidement en reprenant automatiquement les informations détenues par Google. Le fait que l’utilisateur possède un compte valable d’une société reconnue, rend le processus d’autant plus sécurisé.(Anicas 2021)

## Scénario

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

### Création

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

### Suppression

Un bouton sera à disposition pour supprimer le scénario si nécessaire. À 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 que l’on cherche en particulier à gérer. De ce fait, ils intègrent un peu plus de fonctionnalités que les autres aspects de l’application. Ils pourront être créés manuellement, en remplissant tous les champs à la main, de façon semi-automatique, en laissant l’aléatoire décider de certains paramètres, ou complétement aléatoirement 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. À partir de ce point, il sera amené à rentrer les champs obligatoires afin de 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 au besoin principal des utilisateurs, qui est de pouvoir créer des PNJ rapidement, certains champs disposent de bouton qui permettent d’inscrire une valeur aléatoire. Ainsi, il pourra survoler certains champs sans se soucier de quelconque réflexion. De plus, il est possible de créer son personnage à partir de l’intelligence artificielle. En écrivant un simple message, l’utilisateur pourra spécifier un certain type de personnage que l’IA générera toute seule.

### Modification

Afin de faire 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 aventures, il se peut que le personnage change et, s’il est réutilisé dans un autre scénario, il est souhaitable garder sa fiche dans son état initial. C’est pourquoi il est indispensable de pouvoir avoir la possibilité d’appliquer des modifications en tout temps.

### Importation

Comme mentionné précédemment, après des aventures, les MJ pourraient avoir envie de réintégrer un personnage dans un nouveau scénario (suite d’aventure, référence, PNJ). Cette action pourra se faire depuis la page de gestion du scénario. L’utilisateur, à l’aide d’un bouton, pourra voir apparaître une liste de personnage n’appartenant pas au scénario en question. À partir ce point, il pourra sélectionner les personnages qu’il désire. Pour ce faire, les utilisateurs auront la possibilité de pouvoir importer des personnages déjà existants dans d’autres scénarios. Il est important de noter que les personnages importés ne sont pas des duplicatas. Il s’agit de la même fiche, tout changement apporté à cette dernière se verront répercutés d’un scénario à l’autre.

### Suppression

Une page sera dédiée aux personnages créés 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 un suivi de l’équipement directement équipé par le personnage. À l’image de certains attributs que le personnage possède, les armes disposent elles aussi de leurs propres attributs.

### Création

À la différence des personnages, les armes pourront être créées dans une page qui englobe la création et la visualisation d’attributs que j’aborderais dans le point suivant. Ces armes pourront donc être créés 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 mêmes armes à l’aide d’un bouton. Les armes sont des éléments propres à l’utilisateur 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 possible, les attributs sont globaux. Ils ne peuvent être créés que par les utilisateurs disposant des droits suffisants pour exécuter cette action. Néanmoins, même si la majorité des utilisateurs ne seront pas habilités 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 liées à l’attribut.

### Suggestion

Pour les utilisateurs qui ne disposent pas de droits suffisants 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 accessible par les utilisateurs disposant de droit privilégié. Ainsi ils pourront visualiser les attributs et juger par eux même s’il est intéressant de les ajouter à la bibliothèque d’attribut.

# Intelligence artificielle

Pour mettre en avant l’évolution et la nouveauté de l’informatique, j’intègre de l’intelligence artificielle dans cette application. Le model d’IA choisi pour l’application est celui d’OpenAI : « gpt4-o ». C’est le modèle, à ce jour, le plus avancé de l’entreprise. Plutôt, j’ai évoqué l’intégration qu’offrait la librairie React.js. Dans ce cas, c’est également ce qui a énormément orienté le choix du modèle. Vercel offre la possibilité d’utiliser un SDK afin de s’interfacer avec certains modèles, dont « GPT », avec aisance. Ces arguments font que c’est ce modèle qui sera intégré à l’une des fonctionnalités principales de l’application qui est la génération de fiches de personnage. Son objectif sera de produire un format de données correspondant à celui utilisé pour créer, mettre à jour et lire les fiches de personnages afin de suggérer le résultat à l’utilisateur pour validation.

L’utilisateur pourra de son côté, lors de la création de personnage, demander à l’IA de remplir automatiquement tous les champs du formulaire de façon intelligente. Ce dernier pourra écrire un message qui sera utilisé en tant que prompt pour l’IA afin de donner des arguments un peu plus précis sur le type de personnage qu’il souhaite avoir.

À ce jour, l’IA remplit uniquement les champs libres tel que le nom, l’origine, l’âge, etc. Il n’est pas possible de demander qu’un personnage puisse avoir tel ou tel caractéristique ou arme. Néanmoins, dans le futur de l’application, il sera intéressant qu’il puisse le faire pour instaurer une certaine cohésion dans la fiche et qu’ainsi le processus soit complet. Pour se prémunir d’éventuels malfrats qui chercheraient à détourner l’IA de son objectif avec des messages trompeurs, un prompt système est implémenté afin que le résultat de la requête soit toujours satisfaisant.

De ce fait, si quelqu’un rentre un message qui donne une instruction contraire à la mission du modèle, ce dernier n’effectuera pas l’action demandée et continuera ce pourquoi il a été programmé. Ainsi, j’évite les attaques sur l’IA et maintient une expérience utilisateur convenable

Figure Prompt

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

(Enzo Jolidon, 2024)

Au niveau du déploiement de cette fonctionnalité, comme il s’agit là d’un service facturé, il ne sera pas disponible sur la version en production sur le web pour autant que les mandants décident de financer les coûts de l’API. Cette fonction ne requiert, pour l’instant, aucun droit particulier, mais il serait judicieux, si implémenté, de limiter cette fonctionnalité à certains utilisateurs. De ce point, l’application pourrait évoluer en « Software as a Service » et proposer de souscrire à un abonnement pour avoir l’accès à cette fonctionnalité.

# Sécurité

## Authentification

En ce qui concerne la sécurité de l’application, il y a deux points importants à mentionner : l’authentification et les appels API. Pour ce qui est du premier point, les authentifications se font exclusivement par Single-Sign on en utilisant OpenIDConnect. Aucune collecte de données n’est faite par l’application auprès de l’utilisateur. Les seules informations à disposition sont celles retournées par les services utilisés pour s’authentifier. Ces informations sont en général un nom, une adresse électronique ainsi qu’une éventuelle image de profil. Ces informations sont disponibles au bon vouloir de l’utilisateur sur sa page de profil. Il peut, à tout moment, supprimer son compte et ses données seront complètement effacées de l’application.

## API

Un deuxième point spécifique à ce projet en termes de sécurité est l’utilisation des « Server Action ».

Tout d’abord, il est important de contextualiser le sujet et d’expliquer les composants de ce dernier afin de pouvoir approfondir par la suite.

Une application web permet à un utilisateur, depuis son propre appareil électronique ayant accès à un navigateur, d’effectuer des tâches en rapport avec la nature de l’entreprise/association fournissant cette application. Si cette dernière est un magasin en ligne, l’utilisateur pourra alors effectuer des achats ou consulter le catalogue des produits. En revanche, si l’application est fournie par un hôpital, cette dernière permettra sûrement de prendre rendez-vous pour une consultation ou même de prendre contact directement avec un professionnel de la santé. Cette application va tout simplement permettre une liaison entre l’entité qui l’a mise en place et l’utilisateur afin d’accomplir des tâches qui les lient. (Amazon 2024)

Dans le cadre du développement informatique, ces dernières sont connues pour être relativement plus accessibles et simples à développer, poussant les entreprises à se positionner vers le développement d’une application web plutôt qu’une autre forme. (OVHcloud 2024)

Afin de lier les deux acteurs dans une interaction sur une application web, qui sont, le serveur qui va fournir les informations demandées et le client qui est celui qui demande les informations. C’est ici que les actions du serveur interviennent.

### Interface REST

Afin de fournir une application qui comporte des données fiables et accessibles, les serveurs sont entretenus par les fournisseurs de l’application. Ce sont ces serveurs qui alimentent les applications visibles par le client. Néanmoins, il faut que ces applications puissent communiquer avec le serveur afin de délivrer les informations à l’utilisateur. De nos jours, la façon la plus commune et traditionnelle de récupérer des données, dans le cadre d’une application web, est de faire des appels API vers l’URL qui retourne dès lors les données. Ces appels peuvent être de différents types : GET, POST, PUT, DELETE. Elles ont chacune leur rôle pour récupérer, ajouter, modifier ou supprimer les données. Ces interfaces sont les points d’entrée vers le serveur pour les applications situées sur le navigateur de l’utilisateur. (*API REST : définition* 2024)

Grâce au « Server Side rendering » (SSR) combiné aux « Server Components », certains « Framework » web, comme celui utilisé pour l’application de ce travail, mettent à disposition la possibilité d’effectuer des actions côté serveur directement depuis le client. Ces actions sont notamment utilisées pour pouvoir accéder à la base de données sans avoir à faire des requêtes API. Cette façon de faire est utilisée pour récupérer des données plus que pour en ajouter, modifier, supprimer. En comparaison à la non-utilisation des Server Component et du SSR, lors du chargement de la page, l’application devrait faire un appel vers le serveur et attendre le retour des données de ce dernier. Cette façon de faire peut-être très lente en fonction de la distance entre le client et le serveur. Avec les Server Component, il est possible réduire cette latence en récupérant plus rapidement les données étant donné que la page sera chargée avec les informations déjà acquises sans avoir à questionner le serveur. (Vercel 2024a)

En ce qui concerne Next.js, pour rappel, il s’agit du « Framework » web maintenu par Vercel qui est autour de React.js. Ce « Framework » utilise les « Server Components » et aussi le SSR. Il met en place sa propre gestion des routes pour déclarer les liens vers le serveur afin de permettre au composant client de faire des appels vers ces derniers. Il intègre aussi une nouvelle façon de faire des appels vers le serveur depuis les dernières versions de React.(Meta - The React Team 2024)

### Server Action

Une des dernières fonctionnalités stables sortie par React.js est les « Server Actions ». À ce jour, Next.js est le seul Framework qui les met à profit. Cette fonctionnalité est une alternative aux appels API traditionnels. En effet, elle permet de déclarer des fonctions qui seront exécutées côté serveur sans « directement » faire des appels API. On peut associer cela à un raccourci pour les développeurs de la manière dont j’écrit du code dans l’application sans avoir à mettre en place des liens.(Eze 2023)

Comme on ne réinvente pas la roue, ces actions fonctionnent en faisant des appels API au niveau du lien de la page sur laquelle on se trouve au moment du clic. À noter que ces appels sont toujours de type « POST » et jamais autrement, peu importe ce que l’action fait au niveau du serveur ou de la base de données. Toutefois, cette façon de procéder engendre quelques problèmes. (Melvynx • Apprendre à coder 2023) (Vercel 2024b)

### Inconvénients

Le problème majeur est que n’importe qui peut faire des appels tant qu’il connaît le lien et possède l’identifiant lié à la « Server Action ». Un second souci est que si la personne derrière cet appel connaît la façon de procéder du serveur (utilisation d’ORM, SQL, etc.), il peut très bien « injecter » du code dans la requête qui pourraient avoir des conséquences désastreuses comme supprimer tous les enregistrements en base de données. En complément, les fonctions définies pour faire ces actions serveur reçoivent des paramètres qui ne sont pas forcément du bon type et malgré le fait qu’elles soient correctement déclarées cela n’y change rien.(Melvynx • Apprendre à coder 2023)

Comme mentionné précédemment, tout l’appel fait au moyen des servers actions sont de type POST. Dans les normes de l’interface REST, la méthode GET sert à récupérer des données, POST sert à ajouter, PUT/PATCH sert à modifier, DELETE sert à supprimer. Toutefois, rien n’empêche d’utiliser des « Server Actions » pour supprimer modifier ou même récupérer des données. Comme il s’agit de fonctions totalement libres pour le développeur, il peut effectuer les actions qu’il désire. De ce fait, il peut créer des actions qui lui permettront de faire autre chose que des ajouts de données malgré le fait que Next.js fait uniquement des appels de type POST.(Vercel 2024c)

### Problématique

Avec tous les éléments actuels à ma disposition, on comprend bien l’enjeu entre de Next.js. D’un côté, une version traditionnelle en établissant ses propres routes pour interagir avec le serveur et, de l’autre côté, l’opportunité d’une solution plus novatrice sans déclaration de route, mais qui entraîne davantage de problèmes de sécurité, ou du moins, des problèmes qui ne sautent pas aux yeux immédiatement.

À partir de là, ce que l’on souhaite, c’est pouvoir réduire l’écart de fiabilité entre l’interface REST et les « Server Action » de Next.js. Il faudrait donc que les actions concernées ne puissent être atteintes que par les utilisateurs étant authentifiés par l’application et qui ont accès à la page dédiée. De plus, il faudrait que les éléments passés aux actions soient vérifiés aussi bien dans la quantité que dans la qualité. Chaque paramètre doit être du type attendu et la liste dans son ensemble ne devrait pas excéder le nombre d’arguments attendu par la fonction.

De plus, une autre question se pose quant à l’utilisation de cette fonctionnalité. Doit-elle être utilisé pour récupérer des données ? En effet, elle s’agrémente très bien pour l’ajout, la modification et la suppression. Néanmoins, en ce qui concerne la récupération d’informations, il n’en va pas de même. C’est la seule méthode qui oblige que l’on obtienne un retour du serveur et actuellement, le moyen le plus propice à la récupération de données est le « Server component ». Est-ce que dès lors les actions peuvent être considérées comme une alternative à cette méthode également ? S’agit-il d’une faute de responsabilité de les utiliser de cette façon ?

### Solutions

Pour avoir une application qui se prémunit des attaques externes, j’implémente une librairie qui va régler, d’une part, les problèmes d’authentification et d’autre part gérer le format de données.

Cette librairie s’appelle « Next-safe-action ». Elle permet de définir un schéma de données pour valider les types reçus ainsi que le nombre d’arguments passés. De plus, elle ouvre la possibilité d’ajouter des middlewares, ce qui permettra d’effectuer des opérations de validation serveur telles que l’authentification, la vérification du rôle et bien plus encore.(@TheEdoRan 2024)

En ce qui concerne le rôle des « Server Action » dans mon application, ils serviront pour toutes les actions d’ajout, de modification et de suppression. Aucune action ne serait utilisée pour récupérer les données de l’application comme le mentionne Vercel dans sa documentation :

« For most cases, where you don't need real-time data (e.g. polling), you can fetch data on the server with Server Components. » (Vercel 2024c)

# Avenir de l’application

Après tout ce travail, il reste à savoir ce qu’il adviendra de cette application. Comme il s’agit de la première version de l’application, toutes les possibilités sont ouvertes. À ce jour, rien d’officiel n’a été discuté avec les mandants. Cependant, les suggestions suivantes pourraient constituer des pistes intéressantes à explorer pour l’avenir.

## Spécialisation des fiches

À ce jour, les fiches sont globales afin d’être « passe-partout ». Afin d’élargir le champ des possibles, il serait plus qu’intéressant de mettre en place tout l’écosystème d’un système de règles de JDR. Cette approche n’avait pas été de mise dans les prémices de l’application pour des raisons de non-flexibilité. Ce serait une évolution très intéressante pour que les fiches soient plus ergonomiques et qu’elles puissent se rendre plus utiles lors d’une session de jeu. Néanmoins, adapter l’application à un système de règles spécifique nécessite un travail important de récolte de donnée et de modification de la fiche de personnage, tout en rendant l’application moins flexible lors de l’utilisation de règles différentes.

## Software as a Service

Si l’on se penche du côté de la commercialisation de la solution, un passage vers un SaaS pourrait être une opportunité intéressante. L’application pourrait offrir un plan gratuit qui comporte certaines limites (nombre de personnage, scénario, etc.). Ainsi que d’autres plans plus avantageux qui repousserait ces limites et donneraient d’autres avantages (utilisation de l’IA).

## Interface graphique (shared room)

Un grand pas dans l’évolution de l’application, toujours afin de soutenir les MJ dans leur gestion de jeu, serait la possibilité de créer des salons de jeu auxquels les joueurs pourraient se connecter. Ce salon pourra donner accès en quelques clics aux fiches, aux statistiques et aux spécificités tout en donnant la possibilité d’offrir un espace de jeu interactif et visuel pour les groupes jouant à distance.

# Conclusion

Dès lors, cette solution permet à de nombreux MJ d’organiser leur JDR de façon plus ergonomique. La création de fiche de personnages ainsi que leurs accessibilités ont été pensées de façon que les MJ puissent avoir ce qu’il leur faut directement en seulement quelques actions. Cette solution se veut globale afin qu’elle puisse être utilisée par un large panel d’utilisateurs pour qu’ils puissent l’utiliser dans un grand nombre d’univers et systèmes de règles différents. Le tout avec des technologies modernes, tournées vers l’avenir, performantes, sécurisées et facilement maintenables.

Ce travail m’a permis de découvrir en détails, le monde du jeu de rôle qui m’était inconnu ainsi que les problématiques que les MJ peuvent rencontrer. Tout au long du projet, j’ai pu m’aventurer à la recherche de l’implémentation idéale pour les utilisateurs.

J’ai énormément apprécié travailler à ce sujet tout au long de ce travail de Bachelor. Le fait d’avoir des mandants directement concernés par le sujet et donc d’avoir le public cible comme première personne de contact a été très motivant tout au long de mon travail. De plus, pour une première application officielle destinée à des utilisateurs, l’expérience passée en termes de recherche sur les technologies, les méthodes de développement, la résolution de problème, les enjeux de déploiement et la réflexion sur le code fourni m’a renforcé dans l’idée de faire cette pratique mon futur métier. Je me rends compte que les possibilités sont infinies dans ce monde qu’est l’informatique et que la seule limite à ce qu’on peut produire est l’humain.

J’espère que ce travail pourra transmettre la même passion que j’ai eu durant tout le projet pour motiver des étudiants qui s’intéressent au monde de l’informatique et donner envie aux MJ d’essayer l’expérience utilisateur que je propose avec cette solution.

# Bibliographie

AMAZON, 2024. Qu’est-ce qu’une application Web ? – Les applications Web expliquées – AWS. *Amazon Web Services, Inc.* [en ligne]. 1 mai 2024. Disponible à l’adresse : https://aws.amazon.com/fr/what-is/web-application/ [consulté le 2 mai 2024].

ANICAS, Mitchell, 2021. An Introduction to OAuth 2 | DigitalOcean. [en ligne]. 28 juillet 2021. Disponible à l’adresse : https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2 [consulté le 18 août 2024].

API REST : définition, 2024 [en ligne]. Disponible à l’adresse : https://www.redhat.com/fr/topics/api/what-is-a-rest-api [consulté le 3 mai 2024].

EZE, Peter Ekene, 2023. Diving into Server Actions in Next.js 14. *LogRocket Blog* [en ligne]. 21 décembre 2023. Disponible à l’adresse : https://blog.logrocket.com/diving-into-server-actions-next-js-14/ [consulté le 26 avril 2024].

MELVYNX • APPRENDRE À CODER, 2023. *TUTO Server Actions en NextJS : La nouvelle feature React* [en ligne]. 9 novembre 2023. Disponible à l’adresse : https://www.youtube.com/watch?v=7AvrO4fL10A [consulté le 2 mai 2024].

META - THE REACT TEAM, 2024. React 19 Beta – React. [en ligne]. 25 avril 2024. Disponible à l’adresse : https://react.dev/blog/2024/04/25/react-19 [consulté le 3 mai 2024].

OVHCLOUD, 2024. Qu’est-ce qu’une application web ? | OVHcloud | OVHcloud France. [en ligne]. 3 mai 2024. Disponible à l’adresse : https://www.ovhcloud.com/fr/learn/what-is-web-application/ [consulté le 3 mai 2024].

@THEEDORAN, 2024. Getting started | next-safe-action. *Next-safe-action* [en ligne]. 18 août 2024. Disponible à l’adresse : https://next-safe-action.dev/docs/getting-started [consulté le 18 août 2024].

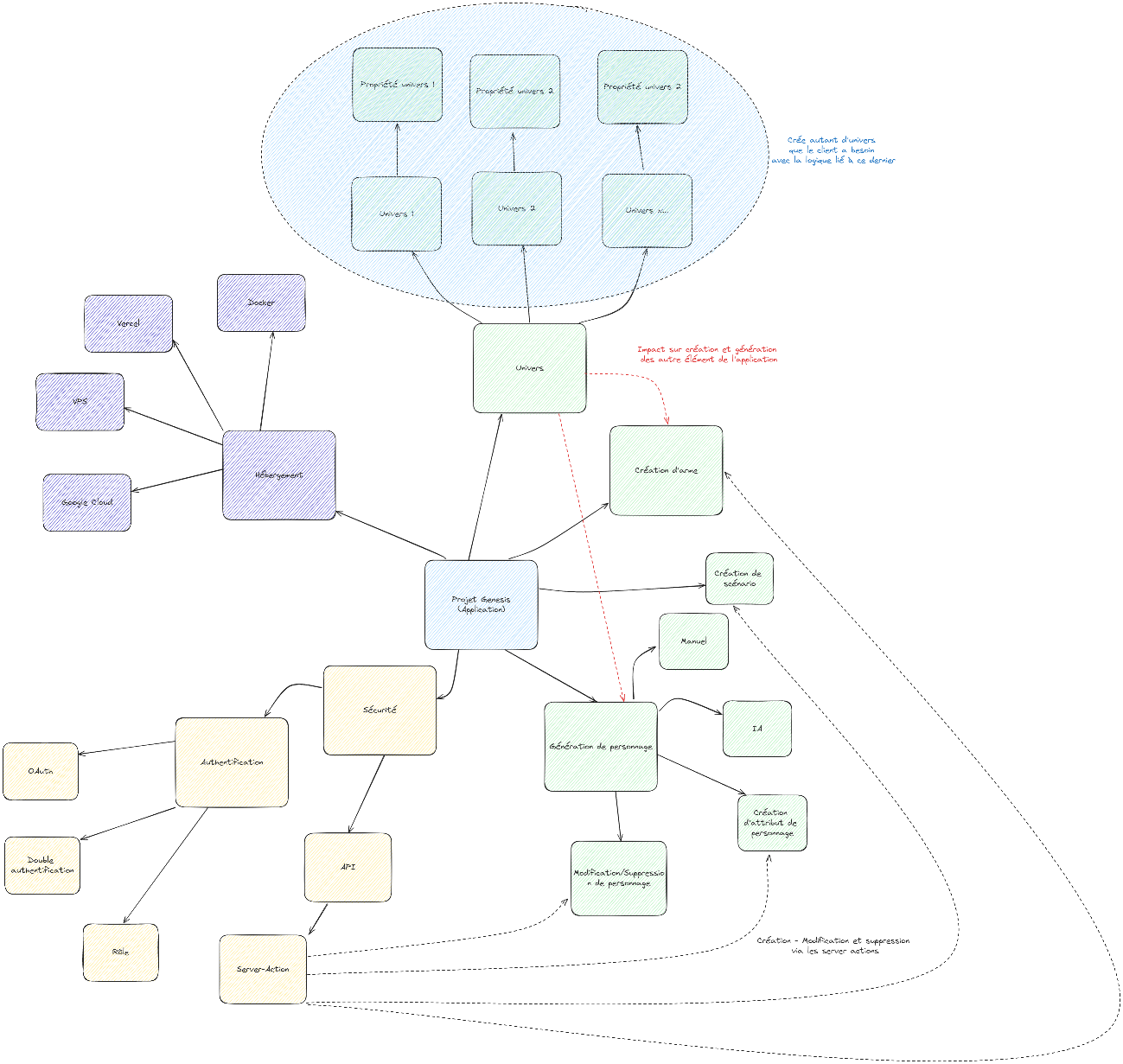
VERCEL, 2024a. Data Fetching: Fetching, Caching, and Revalidating | Next.js. [en ligne]. 23 avril 2024. Disponible à l’adresse : https://nextjs.org/docs/app/building-your-application/data-fetching/fetching-caching-and-revalidating [consulté le 23 avril 2024].

VERCEL, 2024b. Data Fetching: Server Actions and Mutations | Next.js. *Next.js* [en ligne]. 23 avril 2024. Disponible à l’adresse : https://nextjs.org/docs/app/building-your-application/data-fetching/server-actions-and-mutations [consulté le 23 avril 2024].

VERCEL, 2024c. Data Fetching: Server Actions and Mutations | Next.js. *Next.js* [en ligne]. 23 avril 2024. Disponible à l’adresse : https://nextjs.org/docs/app/building-your-application/data-fetching/server-actions-and-mutations [consulté le 23 avril 2024].

WIKIPÉDIA, 2024. Jeu de rôle sur table. *Wikipédia* [en ligne]. Disponible à l’adresse : https://fr.wikipedia.org/w/index.php?title=Jeu\_de\_r%C3%B4le\_sur\_table&oldid=217207627 [consulté le 23 août 2024]. Page Version ID: 217207627

Annexe 1 : MindMap



Annexe 2 : Modèle de base de données

Une image contenant texte, capture d’écran, conception

Description générée automatiquement