

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.

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

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

La gestion de session de jeu dans les jeux de rôle (JDR) peut relever de différent défi que ce soit dans la préparation ou le déroulement de ce dernier. A 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 être confronté à différentes problématique parfois réflexive ou procédurale là où l'informatique peut s'offrir des MJ et de créer une application web moderne utilisant des technologies nouvelles qui permettra de prendre en charge certains aspect de la gestion d'un JDR.

Table des matières

Déclaration	i
Remerciements	ii
Résumé.....	iii
Table des matières.....	iv
Liste des figures	vi
Glossaire des termes.....	1
1. Introduction	2
1.1 Le jeu de rôle	2
1.2 Maître du jeu	2
1.3 Fiche de personnage	2
2. Etat de l'art.....	3
2.1 Noobliees Chronique	3
2.2 CO-DRS	3
2.3 Dyslexic Character Sheets.....	3
3. Technologie	4
3.1 Frontend.....	4
3.2 Backend.....	4
3.2.1 Server Side Rendering	4
3.2.2 React Server Component	5
3.3 Base de données.....	5
3.4 Déploiement.....	5
3.5 Versioning.....	5
3.5.1 Github.....	6
4. Fonctionnalités.....	7
4.1 Authentification	7
4.1.1 OAuth	7
4.2 Scénario	7
4.2.1 Création.....	7
4.2.2 Suppression	8
4.3 Personnage.....	8
4.3.1 Création.....	8
4.3.2 Modification	8
4.3.3 Importation	8
4.3.4 Suppression	9
4.4 Arme	9

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

4.4.1	Création.....	9
4.4.2	Suppression	9
4.5	Attribut d'arme/de personnage	9
4.5.1	Création.....	9
4.5.2	Suggestion	10
5.	Intelligence artificielle	11
6.	Sécurité	13
6.1	Authentification	13
6.2	API.....	13
6.2.1	Interface REST	14
6.2.2	Server Action	14
6.2.3	Inconvénients.....	15
6.2.4	Poblématique	15
6.2.5	Solutions	16
7.	Avenir de l'application	17
7.1	Spécialisation des fiches	17
7.2	Software as a Service	17
7.3	Interface graphique (shared room)	17
8.	Conclusion.....	18
	Bibliographie.....	19
	Annexe 1 : MindMap	20

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

1. Introduction

1.1 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.

1.2 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.

1.3 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 varier 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.

2. Etat de l'art

Aujourd'hui (1^{er} 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.

2.1 Noobliees Chronique

« Noobliees chronique » est un site français qui possède 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 fait et donc nous n'obtenons rien de réutilisable.

2.2 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 est également possible de les partager à d'autres utilisateurs, comme des joueurs, afin qu'ils puissent également suivre l'avancée de leur personnage. Le point le plus freinant est qu'il n'est possible d'avoir uniquement 5 fiches actives pour des raisons de stockage. Cette limitation peut être frustrante lorsqu'on a un JDR avec beaucoup de joueurs ou un nombre important de personnages non-joueurs (PNJ).

2.3 Dyslexic Character Sheets

« Dyslexic Character Sheets » offre des fiches de personnage extrêmement complètes, personnalisables et faciles 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 modifiées.

3. Technologie

3.1 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.

3.2 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 ».

3.2.1 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

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

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.

3.2.2 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.

3.3 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 relationnel (SQL) semble être le choix le plus approprié.

Afin de s'adapter au divers service existant 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.

3.4 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.

3.5 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 développer différentes fonctionnalités sans en entraver d'autre encore en développement. Cette façon de

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

procédé étant plus approprié dans un projet qui engage plusieurs développeurs, elle peut néanmoins avoir certains aspect pratique.

3.5.1 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.

4. Fonctionnalités

4.1 Authentification

Les utilisateurs de cette application seront dans l'obligation de se connecter. Après étude de la meilleure façon d'attribuer de façon privée les différentes données créées par l'utilisateur, 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'implémenterais la fonctionnalité de Single-Sign on (SSO), déjà commune dans beaucoup d'applications : OpenIDConnect. Cette méthode permet 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.

4.1.1 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)

4.2 Scénario

Les scénarios seront les objets racines de la bibliothèque de données de l'utilisateur. 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 différents JDR.

4.2.1 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 accédés plus en détail pour la création de personnage au moyen d'un clic.

4.2.2 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.

4.3 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 créés de façon manuelle, en remplissant tous les champs à la main, de façon semi-automatique, en laissant l'aléatoire décidé de certains paramètres, ou complètement aléatoire avec l'aide de l'intelligence artificielle.

4.3.1 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.

4.3.2 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 modifier en tout temps.

4.3.3 Importation

Comme mentionné précédemment, après des aventures, les MJs pourraient avoir envie de réintégrer un personnage dans un 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 ne sont pas de copie conforme. Il s'agit de la même fiche, ce qui veut dire que tout changement apporté à cette dernière se verra répercuté autant pour un scénario qu'un autre.

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

4.3.4 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.

4.4 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.

4.4.1 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.

4.4.2 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.

4.5 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.

4.5.1 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.

4.5.2 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.

5. 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 model, à ce jour, le plus avancé de l'entreprise et un des plus performant du monde. Elle sera intégrée 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 donnée correspondant à celui utilisé pour créer, mettre à jour et lire les fiches de personnages.

L'utilisateur pourra de son côté, lors de la création de personnage, demandé 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.

A 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 prévenir d'éventuel malfrat qui chercherait à détourner l'IA de son objectif, un prompt système est implémenté afin que le résultat de la requête soit toujours satisfaisant.

Figure 1 Prompt

```
Context:
You are a Project-Genesis AI, an IA that create character object in JS/TS.
You have a lot of imagination in order to create great character for scenarios.

Goal:
Generate a character that match the schema given to you.

Criteria:
* The object that you generate MUST match the schema given to you.
* You never use someone reel name on purpose.
* You never add properties that is not in the schema.
* You exclusively create JS/TS object.
* You never write text or explanation of the generated object.
* You never generate the same object twice.
* You always write properties in the same language as the user prompt.
* If the user ask you something else than generating a character, you reply with an empty object.
Response format:
You reply with the generated object that match the schema.
```

(Enzo Jolidon, 2024)

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

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.

6. Sécurité

6.1 Authentification

En ce qui concerne la sécurité de l'application, il y a 2 points important 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 donnée n'est faite par l'application auprès de l'utilisateur. Les seules informations à disposition sont celles retournée par les services utilisés pour s'authentifier. Ces informations 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 effacé de l'application.

6.2 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, 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 serveur interviennent.

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

6.2.1 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 nous 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 plutôt 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, nous pouvons 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 Component 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)

6.2.2 Server Action

Une des dernières fonctionnalités stables sortie par React.js est les « Server Action ». A ce jour, Next.js est le seul Framework qui la 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

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

associer cela à un raccourci pour les développeurs dans la manière dont on écrit du code dans notre 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 où l'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)

6.2.3 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 en ajouter, PUT/PATCH sert à modifier, DELETE sert à supprimer. Toutefois, rien n'empêche d'utiliser des servers action pour supprimer modifier ou même récupérer des données. Comme il s'agit de fonction totalement libre 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)

6.2.4 Poblématique

Avec tous les éléments actuels à notre disposition, on comprend bien l'enjeu entre de Next.js. D'un côté, nous avons une version traditionnelle en établissant nos propres routes pour interagir avec le serveur et de l'autre côté, nous avons 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 qu'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

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

puissent être atteinte que par les utilisateurs étant authentifiés par l'application et qui ont accès à la page dédié. 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.

Aussi, 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'information, il n'en va pas de même. C'est la seule méthode qui oblige qu'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 à une alternative pour cette méthode également ? S'agit-il d'une faute de responsabilité de les utiliser de cette façon ?

6.2.5 Solutions

Pour avoir une application qui se prémunit des attaques externes, j'implémenterais 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 donnée pour valider les types reçus ainsi que le nombre d'argument passé. De plus, elle ouvre la possibilité d'ajouter des middlewares, ce qui permettra d'effectuer des opérations de validation server tel 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 tout 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)

7. 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. A 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.

7.1 Spécialisation des fiches

A 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 univers 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 soit plus ergonomique et qu'elles puissent se rendre plus utile lors d'une session de jeu. Néanmoins, ce processus peut être long ne serait-ce que pour un univers.

7.2 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 certaine limitation (nombre de personnage, scénario, etc.). Ainsi que d'autre plan plus avantageux qui repousserait ces limitations et donnerais d'autre avantage (utilisation de l'IA).

7.3 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 quelque clique 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.

8. 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 leur accessibilité à été pensée de façon à que les MJ puissent avoir ce qu'il leur faut directement en seulement quelques actions. Cette solution se veut global afin qu'elle puisse être utilisée par un large panel d'utilisateur pour qu'ils puissent l'utiliser dans un grand nombre d'univers. Le tout avec des technologies moderne, tourné vers l'avenir, performante, sécurisé et facilement maintenable.

Ce travail m'a permis de découvrir dans les 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 idéal pour les utilisateurs.

J'ai énormément apprécié travailler sur 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 du projet. De plus, pour une première application officielle destiné à des utilisateurs, l'expérience passé 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 fournie 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 moi-même.

J'espère que ce travail pourra transmettre la même passion que j'ai eu durant tout le projet pour motiver des étudiants qui s'intéresse 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].

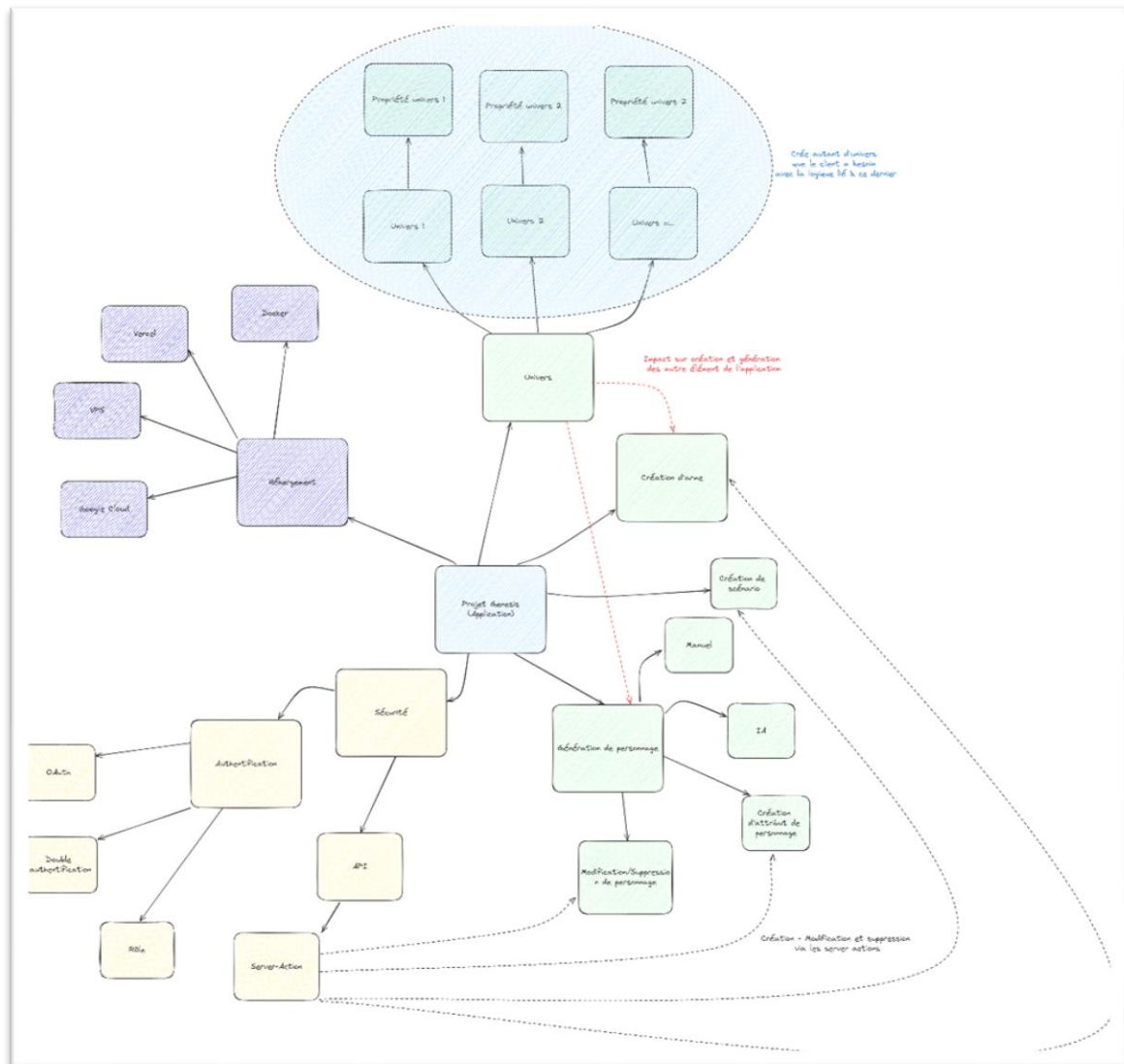
VERCEL, 2024a. Data Fetching: Fetching, Caching, and Revalidating | Next.js. [en ligne]. 23 avril 2024. Disponible à l'adresse : <https://nextjs.org/docs/app/building-your-application/data-fetching/fetching-caching-and-revalidating> [consulté le 23 avril 2024].

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].

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

Annexe 1 : MindMap



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