



Grimoires Guardians

Nom de l'Étudiant: Lehenerz Geoffrey

Date: 23/04/2024

Dernière modification: 23/04/2024

Cours/Institution: Projet Web, EAFC Fleron Charlemagne

Cahier des Charges - Grimoires Guardians

- Grimoires Guardians
- Cahier des Charges - Grimoires Guardians
 - 1. Définition des Parties
 - 2. Contexte du Projet
 - Objectifs Principaux
 - Objectifs Secondaires
 - 3. Présentation de l'Application
 - Accessibilité
 - Fonctionnalités
 - Pour la première phase (projet):
 - Pour les joueurs :
 - Pour le MJ :
 - Pour la seconde phase (projet) :
 - 4. Technologies Utilisées
 - Front-end
 - Back-end
 - Base de Données
 - 5. Déploiement et Maintenance
 - 7. Prix
 - 8. Mentions Légales

1. Définition des Parties

Client :

Le client pour ce projet est l'université qui supervise le travail de fin d'études. Le représentant désigné pour ce projet est le professeur chargé du cours de développement web, qui évaluera les aspects techniques et la réalisation du projet.

Développeur :

Lehenerz Geoffrey, étudiant en développement web, responsable de la conception, du développement, de l'implémentation, et de la maintenance de l'application web "Grimoires Guardians". Ce projet constitue une partie significative de son évaluation finale.

2. Contexte du Projet

"Grimoires Guardians" est un projet ambitieux visant à créer un site de jeu de rôle interactif qui permettra aux utilisateurs de s'immerger dans des univers fantastiques à travers des jeux de rôle en ligne. Ce projet émerge de la passion pour les jeux de rôle et de l'observation d'un besoin de plateformes intégrées permettant une gestion fluide et complète des éléments clés de ces jeux. Notre but est de développer une plateforme qui non seulement héberge des sessions de jeu captivantes mais facilite également la gestion des campagnes par les maîtres de jeu (MJ) et l'engagement des joueurs.

Le projet utilisera des technologies modernes telles que AdonisJS pour le backend et React pour le frontend, choisies pour leur robustesse, leur flexibilité, et leur large communauté de support. Ces technologies permettront de mettre en place une architecture solide, adaptative et capable de supporter des extensions futures, comme l'ajout de nouveaux systèmes de jeux.

Objectifs Principaux

1. **Création d'une Plateforme Multi-utilisateurs** : Permettre à plusieurs groupes de joueurs de créer, gérer et participer à des jeux de rôle. Les utilisateurs pourront s'inscrire en tant que MJ ou joueurs, avec des niveaux d'accès et des fonctionnalités spécifiques à chaque rôle.
2. **Gestion Intégrée des Parties** : Offrir des outils pour la création et la maintenance de fiches de personnage, de jets de dés automatisés, et de gestion des éléments de jeu tels que les cartes, les sortilèges, les objets et les créatures.
3. **Interface Interactive et Immersive** : Développer une interface utilisateur conviviale et esthétiquement agréable qui engage les utilisateurs et enrichit leur expérience de jeu.
4. **Tchat en Direct pour les Joueurs** : Implémenter un système de tchat en direct où les joueurs peuvent non seulement effectuer des jets de dés visibles par tous mais aussi écrire et communiquer librement pendant le jeu, renforçant ainsi l'immersion et l'interaction.

Objectifs Secondaires

1. **Scalabilité** : Construire une base solide qui permette l'intégration de nouveaux jeux et systèmes à l'avenir sans nécessiter de refonte majeure.

3. Présentation de l'Application

Accessibilité

L'application sera accessible via un navigateur web, sans nécessité de télécharger ou installer de logiciel additionnel. Optimisée pour les ordinateurs de bureau, elle garantira une accessibilité maximale pour tous les utilisateurs intéressés par les jeux de rôle.

Fonctionnalités

Pour la première phase (projet):

- **Page d'accueil interactive** : Les utilisateurs seront accueillis par une interface utilisateur dynamique qui affichera les options de compte, les parties auxquelles on est enregistré en tant que joueur et/ou maître du jeu (MJ).
- **Barre de navigation** : Facilitera la navigation entre différentes sections telles que le chat, l'accès à sa fiche de personnage, les objets ou les sorts.

Pour les joueurs :

- Accès au chat.
- Possibilité de jeter des dés.
- Accès à leur fiche de personnage.
- Accès à la liste des sorts.
- Accès à la liste des objets.
- Accès à la liste des capacités.
- Accès à la liste des métiers (classe, sous-classe).

Pour le MJ :

- Accès au chat.
- Possibilité de jeter des dés.
- Accès et création à toutes les fiches de personnages, de PNJ (personnage non jouable), et des créatures (profil simplifié ?).
- Accès et création de sorts.
- Accès et création d'objets.
- Accès et création des capacités.
- Accès et création des métiers (classe, sous-classe).

Pour la seconde phase (projet) :

- Un système de "drag and drop" des objets vers les fiches pour les lier.
- Système de carte où l'on pourra déplacer nos tokens (symbolisant les créatures et différents personnages) pour imager les phases de combat.

4. Technologies Utilisées

Front-end

- **React** : Pour construire une interface utilisateur dynamique avec des composants réactifs.
- **Redux ou Context API** : Pour la gestion de l'état global de l'application, facilitant la communication entre les composants.

- **React Router** : Pour gérer la navigation dans l'application.
- **Axios** : Pour faire des requêtes HTTP au serveur back-end, facilitant l'intégration avec AdonisJS.
- **Socket.IO Client** : Pour la communication en temps réel, nécessaire pour le chat et la mise à jour des positions des tokens.
- **Material-UI ou Bootstrap React** : Pour utiliser des composants UI prêts à l'emploi, accélérant ainsi le développement de l'interface.

Back-end

- **AdonisJS** : Framework backend robuste qui fournit une architecture MVC, facilitant l'organisation et la maintenance du code.
- **Lucid ORM** : Inclus dans AdonisJS, permettant de manipuler facilement les données relationnelles.
- **AdonisJS Auth** : Pour une implémentation facile de l'authentification, incluant les sessions et les tokens.
- **Edge Template Engine** : Utilisé pour générer des vues côté serveur, si nécessaire.
- **Socket.IO** : Intégré pour la gestion des communications en temps réel avec les clients.
- **Envoy** : Outil de AdonisJS utilisé pour la gestion des tâches d'administration système et de déploiement.

Base de Données

- **MySQL** : Base de données relationnelle utilisée pour stocker toutes les données structurées, telles que les informations des utilisateurs, les fiches de personnage, les détails des parties, etc.
- **Migrations et Seeds** : Utilisez les outils de migration d'AdonisJS pour définir et maintenir la structure de la base de données, facilitant les mises à jour et les déploiements.

5. Déploiement et Maintenance

- **GitHub** : Il a été décidé d'utiliser GitHub pour le déploiement ainsi que la maintenance de l'application.

7. Prix

- Les services proposés par l'application sont fournis gratuitement. La main-d'œuvre et le coût de l'investissement moral du prestataire ne sont pas facturés au client.

8. Mentions Légales

- **Droits et responsabilités** : Tous les utilisateurs auront accès aux fonctionnalités de base sans restriction, mais seront soumis aux termes d'utilisation qui incluent la non-responsabilité en cas de perte de données ou d'interruption de service.