



IFAPME Tournai – Wallonie Picarde

Travail de fin d'études présenté en vue de l'obtention du titre de

Développeur Web Front-End X75

barades.com

L'application pour organiser vos sessions de jeu sur table



Année académique 2024-2025

Théophile Desmedt

Formateur: DURIBREUX GUILLAUME



IFAPME Tournai – Wallonie Picarde

Travail de fin d'études présenté en vue de l'obtention du titre de

Développeur Web Front-End X75

Barades

**L'application pour organiser vos
sessions de jeu sur table**



Année académique 2024-2025

Théophile Desmedt

Formateur: DURIBREUX GUILLAUME

Remerciements

Je remercie l'IFAPME Tournai et particulièrement Monsieur **PARFAIT Martial** ainsi que Monsieur **RACQUEZ François**, qui ont su être présents pour nous aider à approfondir les concepts avancés de JavaScript avec une grande pédagogie, essentielle pour réaliser ce projet.

Je suis reconnaissant envers Monsieur **Quentin MOUSSET**, qui m'a accepté en stage dans son agence. Je tiens également à remercier l'ensemble des formateurs: **BACHELY Vincent**, **DURIBREUX GUILLAUME**, **VAN ONACKER Michaël**, pour leurs conseils, leur patience et leur passion contagieuse pour le développement web.

Enfin, merci à ma famille et mes amis pour leur soutien essentiel.

Table des matières

Introduction.....	7
Problématique.....	7
Développement.....	8
3.1. Analyse des besoins.....	8
3.2. Gestion de projet.....	8
3.2.0 Méthodologie.....	8
3.2.1. Planning d'exécution.....	8
3.2.2. Impact Mapping.....	9
3.3. Conception.....	10
3.3.1. Arborescence du site.....	10
3.3.2. Wireframes.....	11
3.3.3. Charte graphique.....	11
3.3.4. Maquettes.....	12
3.4. Base de données.....	12
3.4.1. Modèle Conceptuel de Données.....	12
3.4.2. Modèle Logique de Données (ERD).....	12
3.4.3. Sécurité et Row Level Security.....	12
3.5. Choix technologiques.....	12
3.5.1. Front-end : Angular.....	13
3.5.2. Back-end : NestJS.....	13
3.5.3. Base de données : Supabase/PostgreSQL.....	13
3.5.4. API externes.....	14
3.6. Réalisation technique.....	14
3.6.1. Architecture de l'application.....	14
3.7. Mise en production.....	15
3.7.1. Déploiement.....	15
3.7.2. Migration de la base de données.....	15
3.7.3. Configuration.....	16
J'ai choisi ovh pour acheter le nom de domaine du site, barades.com. Le prix était très attractif, seulement 13,49 €/an.....	16
Conclusion.....	16
4.1. Évaluation du travail accompli.....	16
4.2. Difficultés rencontrées et solutions.....	16
4.3. Perspectives d'évolution.....	17
4.4. Bilan personnel.....	17
Références bibliographiques.....	18
1. Become a ninja with Angular.....	18
Annexes.....	19
Wireframes.....	20

Introduction

L'industrie mondiale du jeu de société connaît une croissance spectaculaire, avec un chiffre d'affaires estimé à 4,84 milliards de dollars d'ici 2027, soit une progression annuelle de 9,31 % entre 2022 et 2027¹. L'industrie voit paraître chaque année plus de 1200 nouveautés², et ça seulement pour la France !

Problématique

Face à ce raz de marré de possibilités, les consommateurs sont confrontés à une paralysie du choix. Les bars à jeux et ludothèques jouent un rôle crucial en permettant aux joueurs de tester des jeux avant de s'engager dans un achat coûteux.

Et pourtant, malgré l'engouement des joueurs, les lieux ludiques restent rares. Pour les non initiés, il est parfois difficile de savoir où se retrouver pour jouer, et quand.

C'est face à ce constat qu'est né le concept de Baradés, une application web ayant pour but d'aider les joueurs à se rencontrer et à découvrir de nouveaux lieux.

¹ <https://gusandco.net/2023/11/22/statistiques-jeux-de-societe-2023>

²<https://www.lefigaro.fr/conso/1200-nouveautes-chaque-annee-la-surproduction-de-jeux-de-societe-met-elle-le-secteur-en-peril-20250727>

Développement

3.1. Analyse des besoins

Pour être efficace, un projet doit répondre à des besoins client spécifiques et y apporter des solutions. Plusieurs méthodes peuvent être utilisées pour recueillir ces besoins: sondages, interviews, réunions, témoignages, etc. Ici, le choix étant libre, je suis mon propre client et connaît donc mes besoins. Dans le cas de Baradés, qui s'adresse à des joueurs itinérants (clients), l'application doit être :

- **Utile:** planification de sessions de jeux, découverte de personnes et de lieux
- **Satisfaisante:** chargement rapide, navigation intuitive (UX)
- **Captivante:** design moderne (UI), interactivité
- **Sécurisée:** respecter la vie privée des utilisateurs
- **Pérenne:** elle doit pouvoir être mise à jour et évoluer
- **Ecologique:** consommer le moins de batterie/data possible, réduire les achats inutiles en permettant d'essayer les jeux

Une fois ces besoins identifiés, il faut également prendre en compte les contraintes de temps, de budget et d'expérience pour trouver la solution la plus adaptée et réaliste à chacun de ces problèmes. Cela permet d'établir la liste précise des objectifs qui seront livrés au client, dans un délai déterminé.

Les fonctionnalités clés sont donc:

1. Un site web single page événementiel
2. Carte interactive affichant des lieux ludiques
3. Système de création de sessions de jeu avec inscription
4. Cryptage des données (bcrypt)
5. Choix de technologies permettant le support sur le long terme
6. Technologies gratuites et open source (autant que possible)

3.2. Gestion de projet

3.2.0 Méthodologie

Pour ce travail, j'ai décidé de m'aider d'agent LLM (GPT-5-Codex, Claude Sonnet 4.5, Gemini 2.5 pro) pour itérer très rapidement et implémenter des fonctionnalités au plus vite. Je suis persuadé que l'avenir est indissociable des intelligences artificielles, il faut donc faire de notre mieux pour apprendre à s'en servir dans notre processus de développement, de façon raisonnée. Afin de réduire les risques d'hallucination, j'ai placé des tests unitaires et e2e, et je vérifie manuellement les modifications réalisées. Les IA sont encore loin d'être parfaites, et il est souvent nécessaire de corriger leur travail, mais elles sont un outil précieux pour appréhender une technologie nouvelle.

3.2.1. Planning d'exécution

Dû à des contretemps personnels, le projet Barades a été réalisé sur une période de deux semaines, avec une répartition du temps selon les phases suivantes :

Phase 1 : Analyse et Conception

1. Analyse des besoins et définition d'un périmètre précis
2. Recherche de technologies adaptées
3. Création de l'impact mapping
4. Conception de l'arborescence et wireframes
5. Prototypage rapide sur gemini canvas

Phase 2 : Modélisation de la base de données

1. Conception du MCD (Modèle Conceptuel de Données)
2. Création du ERD (Entity Relationship Diagram)
3. Configuration de Supabase et PostgreSQL
4. Mise en place de la sécurité (Row Level Security)

Phase 3 : Développement Front-end

1. Configuration de l'environnement Angular
2. Développement des composants UI
3. Intégration de la charte graphique
4. Mise en place du routing et de la navigation
5. Intégration des APIs (cartes, authentification)
6. Responsive design et optimisations mobiles

Phase 4 : Développement Back-end

1. Configuration de NestJS
2. Développement des modules et services
3. Création des endpoints API
4. Intégration avec Supabase
5. Gestion de l'authentification et des sessions

Phase 5 : Tests et Débogage

1. Tests unitaires (Jest)
2. Tests d'intégration
3. Tests end-to-end (Playwright)
4. Correction des bugs identifiés
5. Tests de performance

Phase 6 : Déploiement et Mise en Production (Semaines 15-16)

1. Configuration du serveur de production
2. Migration de la base de données
3. Déploiement de l'application
4. Configuration du domaine et SSL

3.2.2. Impact Mapping

Objectif principal : Faciliter les rencontres entre joueurs de jeux de société

WHO

1. **Joueurs itinérants** : Personnes cherchant où et quand jouer
2. **Organisateurs de sessions** : Joueurs créant des événements
3. **Propriétaires** : Bars à jeux, ludothèques, associations

HOW

Pour les joueurs itinérants :

1. Découvrir des lieux ludiques près de chez eux
2. Rejoindre des sessions de jeu
3. Rejoindre des groupes de joueurs

Pour les organisateurs :

1. Créer des sessions de jeu
2. Recruter des joueurs
3. Gérer les inscriptions

Pour les propriétaires des lieux :

1. Augmenter la visibilité de leur établissement
2. Attirer des clients

Livrables (WHAT) :

1. **Carte interactive**
 - a. Affichage géolocalisé des lieux ludiques
 - b. Filtres par type de lieu
 - c. Navigation vers les établissements
2. **Système de sessions**
 - a. Crédit de sessions de jeu
 - b. Inscription/désinscription
 - c. Notifications
 - d. Gestion du nombre de places
3. **Profils utilisateurs**
 - a. Authentification sécurisée
 - b. Gestion des préférences
 - c. Historique des sessions
4. **Interface responsive**
 - a. Application mobile-first
 - b. Performance optimale
 - c. Mode hors ligne (PWA)

3.3. Conception

3.3.1. Arborescence du site

```
Racine ('')
  -- '' → HomePage
  -- sessions
    -- '' → SessionsListPage
    -- new → SessionCreateComponent (authGuard)
    -- :id → SessionDetailComponent
    -- :id/edit → SessionEditComponent (authGuard)
  -- locations → LocationsListComponent
  -- groups
    -- '' → GroupsListComponent
    -- :id → GroupDetailComponent
  -- forum → ForumPage
  -- charter → CharterPage
  -- profile → ProfilePage (authGuard)
  -- dashboard → DashboardPage (authGuard)
  -- showcase → ShowcasePage
  -- about → AboutPage
  -- careers → CareersPage
  -- contact → ContactPage
  -- help → HelpPage
```

3.3.2. Wireframes & Layouts

Voir annexes page 20 et 21

3.3.3. Charte graphique

Palette de couleurs :

- Couleur principale : Indigo profond `#6366f1` (hover `#4f46e5`, focus `#818cf8`)
- Couleurs secondaires : Fonds sombres `#0f1419`, `#1a1d2e`, `#262b3d` pour renforcer le contraste
- Couleurs de texte : `#f9fafb` (principal), `#d1d5db` (secondaire), `#9ca3af` (tertiaire)
- Couleurs d'état : Succès `#10b981`, Erreur `#ef4444`, Alerte `#f59e0b`, Info `#3b82f6`

Typographie :

- Titres : `Inter` (gras 700 pour H1/H2, medium 600 pour H3/H4)
- Corps de texte : `Inter` taille 1rem, graisse 400 avec un interlignage 1.5
- Pile de secours : `system-ui`, `-apple-system`, `BlinkMacSystemFont`, `Segoe UI`, `sans-serif`
- Hiérarchie : H1 2.5rem, H2 2rem, H3 1.5rem, H4 1.25rem, H5 1.125rem, H6 1rem

Logo et identité visuelle :



Le logo de barades est un d6 de jeu de rôle posé devant un livre de règle, un pion à ses côtés, pour faire écho aux joueurs de jeu sur table auquel le site s'addresse. Le logo a été généré avec gemini, puis retouché et vectorisé manuellement en utilisant le logiciel inkscape.

3.4. Base de données

3.4.1. Modèle Conceptuel de Données

Entités principales :

1. **User** : Utilisateurs de l'application
2. **Location** : Lieux ludiques (bars à jeux, ludothèques)
3. **Session** : Sessions de jeu organisées
4. **Group** : Groupes de joueurs
5. **Poll** : Sondages pour choisir des jeux
6. **Reservation** : Réservations de places dans les sessions

Relations :

1. Un User peut créer plusieurs Sessions (1,n)
2. Un User peut participer à plusieurs Sessions (n,n)
3. Une Session se déroule dans un Location (n,1)
4. Une Session peut avoir plusieurs Polls (1,n)
5. Un User peut rejoindre plusieurs Groups (n,n)

3.4.2. Modèle Logique de Données (ERD)

Voir Annexes page 22, fig. 6

Note: Pour maintenir la documentation à jour avec le schéma de la base de données, le projet utilise **Prisma ERD Generator**, un outil qui génère automatiquement un diagramme visuel à partir du fichier `schema.prisma`.

3.4.3. Sécurité et Row Level Security

La RLS est une fonctionnalité de PostgreSQL qui permet de définir des politiques de sécurité au niveau des lignes de données. Cela garantit que les utilisateurs ne peuvent accéder qu'aux données qui leur appartiennent ou qu'ils sont autorisés à voir.

3.5. Choix technologiques

3.5.1. Front-end : Angular

Selon l'enquête [State of JavaScript 2024](#)³, Angular est le troisième framework front-end le plus utilisé au monde, derrière React et Vue.js.

Cependant, Angular possède un atout majeur par rapport à ses concurrents : c'est un framework "opinionné". Il impose un cadre et des bonnes pratiques de développement, ce qui est idéal pour la pérennité du projet. En partant sur des bases saines, on diminue les risques de dette technique qui rendrait le projet ingérable.

Avantages d'Angular pour ce projet :

1. **Structure claire** : Architecture modulaire imposée qui facilite l'organisation du code
2. **TypeScript natif** : Typage fort qui réduit les bugs et améliore la maintenabilité
3. **Outils intégrés** : Router, HTTP client, Forms, tout est inclus
4. **CLI puissant** : Génération de code, builds, tests automatisés
5. **Écosystème mature** : Documentation complète, grande communauté
6. **Performance** : Change detection optimisée, lazy loading natif
7. **PWA** : Support intégré pour Progressive Web Apps
8. **Signals** : Nouvelle approche réactive (Angular 16+) pour une meilleure performance

Technologies complémentaires :

1. **Angular Material** : Bibliothèques de composants UI moderne
2. **RxJS** : Programmation réactive pour gérer les flux de données
3. **Nx** : Monorepo tooling pour gérer le projet full-stack

3.5.2. Back-end : NestJS

NestJS est **fortement inspiré par Angular**.

1. Il utilise la même structure en Modules, Services/Providers, Controllers.
2. Il implémente le même système d'injection de dépendances, facilitant la gestion du code et les tests.
3. Tout comme Angular, NestJS est entièrement basé sur TypeScript, un superset de Javascript, assurant une cohérence de langage et de typage sur l'ensemble de la stack (full-stack).

La structure et les bonnes pratiques imposées par NestJS et Angular sont idéales pour un projet qui vise la **pérennité** et la maintenabilité. Cela garantit que le projet reste organisé à mesure qu'il grandit. De plus, la logique similaire permet de réduire la courbe d'apprentissage. NestJS nous permet de gérer l'accès aux routes authentifiées de notre application et de créer facilement des points d'API pour alimenter notre frontend Angular.

³ https://2024.stateofjs.com/en-US/libraries/front-end-frameworks/#front_end_frameworks_work

3.5.3. Base de données : Supabase/PostgreSQL

Choix de Supabase :

Supabase est une alternative open source à Firebase, construite sur PostgreSQL.

Avantages :

1. **PostgreSQL** : Base de données relationnelle robuste et mature
2. **Open source** : Code source accessible, pas de vendor lock-in
3. **Row Level Security** : Sécurité au niveau des lignes
4. **API REST auto-générée** : Pas besoin de tout coder manuellement
5. **Realtime** : WebSockets pour les mises à jour en temps réel
6. **Storage** : Stockage de fichiers (avatars, images)
7. **Auth intégré** : Authentification avec email, OAuth, magic links
8. **Tier gratuit** : 500 Mo de stockage, 2 Go de bande passante

Alternatives considérées :

1. **Prisma + PostgreSQL** : Très bon mais nécessite plus de configuration
2. **MongoDB** : NoSQL, moins adapté pour des données relationnelles
3. **Firebase** : Propriétaire, prix élevé à grande échelle

3.5.4. API externes

OpenStreetMap & Leaflet :

Pour la carte interactive affichant les lieux ludiques :

1. **OpenStreetMap** : Données cartographiques open source
2. **Leaflet** : Bibliothèque JavaScript légère pour cartes interactives
3. **Nominatim** : Géocodage (conversion adresse ↔ coordonnées)

Alternative considérée :

1. **Google Maps API** : Payante après quota gratuit, moins de contrôle

3.6. Réalisation technique

3.6.1. Architecture de l'application

Architecture Monorepo avec Nx :

Le projet utilise Nx pour gérer un monorepo contenant le backend et le front end en un endroit centralisé, facilitant l'échange entre les deux systèmes. Le monorepo Nx inclut des outils de documentation et visualisation :

1. **Nx Graph** : Graphe interactif des dépendances entre apps et libraries
2. **Prisma ERD Generator** : Génération automatique du diagramme de base de données
3. **Mermaid CLI** : Création de diagrammes de flux et user journeys

Ces outils permettent de maintenir automatiquement une documentation technique à jour, synchronisée avec le code source.

Outils de monitoring :

1. **Nx Console** (VS Code extension) : Interface graphique pour les commandes Nx
2. **Chrome DevTools** : Debugging du frontend Angular
3. **Supabase Dashboard** : Monitoring de la base de données et des queries

3.7. Mise en production

3.7.1. Déploiement

Frontend (Vercel) :

L'application Angular est déployée sur Vercel, une plateforme optimisée pour les applications front-end.

Processus de déploiement :

1. Push sur la branche `main`
2. Vercel détecte automatiquement le push
3. Build de l'application (`nx build frontend --prod`)
4. Déploiement automatique
5. URL de production : <https://barades.vercel.app>

Backend (Render) :

L'API NestJS est déployée sur Render, une plateforme cloud pour les applications backend.

Processus de déploiement :

1. Push sur la branche `main`
2. Render détecte le changement
3. Build du backend
4. Redémarrage automatique du service
5. URL API : <https://barades-api.onrender.com>

3.7.2. Migration de la base de données

Prisma Migrations :

Les migrations de la base de données sont gérées avec Prisma, ce qui permet de faire un versionning de la base de donnée durant le développement.

Seeding de la base de données :

Pour pouvoir tester l'application avec des tests unitaires et end to end (playwright), la base de donnée est allimentée par des données factices générées via un script de seeding.

3.7.3. Configuration

Les données sensibles sont stockées dans des fichiers d'environnement qui ne sont jamais commitées dans Git.

Domaine personnalisé (optionnel) :

J'ai choisi ovh pour acheter le nom de domaine du site, barades.com. Le prix était très attractif, seulement 13,49 €/an.

Conclusion

4.1. Évaluation du travail accompli

Le projet Barades est une application web complète, fonctionnelle et utile, disposant de nombreuses fonctionnalités:

- PWA Angular offline first
- Carte interactive fonctionnelle
- Système de gestion avec dashboard pour avoir une vue d'ensemble sur les groupes et les sessions créées
- Sécurité avec JWT et la Row Level Security de Supabase
- Monorepo Nx Angular + NestJS permettant l'ajout facile de nouvelles fonctionnalités, proprement
- Application accessible publiquement sur Vercel et Render
- Test automatisés avec Jest (unitaires) et Playwright (E2E), pour garantir la qualité du code

Malgré les manquements sur le design de l'application, trop basique, je suis fier d'être arrivé à créer une application from scratch, avec plusieurs outils relativement nouveaux pour moi: nestjs, angular, nx. Je crois en ce projet et je pense qu'il vaut la peine de continuer à le développer.

4.2. Difficultés rencontrées et solutions

J'ai passé beaucoup de temps à vouloir optimiser l'application en passant l'application angular en zoneless et en utilisant les signals, mais la documentation est les exemples sont encore relativement limités. J'ai dû utiliser une combinaison de Signaux pour gérer l'état local et RxJS pour les flux asynchrones, de façon à garder le bundle javascript petit sans sacrifier en réactivité.

Comprendre et déboguer les politiques RLS dans Supabase était complexe, surtout pour les relations entre tables. L'outil de debug de Supabase pour visualiser les requêtes SQL générées m'a été très précieux.

J'ai également rencontré des problèmes avec l'API Nominatim d'OpenStreetMap, qui a des limites relativement strictes. Pour réduire les erreurs de taux dépassées, j'ai mis en place un

système de cache côté client et backend et ajouté du debouncing sur les recherches utilisateur pour réduire le nombre de requêtes.

J'ai pris du retard sur le projet en voulant implémenter trop de fonctionnalités "nice-to-have". J'aurais dû me fier plus à l'impact mapping pour prioriser les fonctionnalités essentielles.

4.3. Perspectives d'évolution

De nombreuses améliorations peuvent être apportées au projet, particulièrement sur son design. J'ai également une longue liste de fonctionnalités qui pourrait y être ajoutées:

Court terme (3-6 mois) :

1. **Notifications push** : Alertes pour nouvelles sessions, rappels avant événements
2. **Système de messagerie** : Chat entre participants d'une session
3. **Évaluation des lieux** : Système de notes et commentaires
4. **Filtres avancés** : Par type de jeu, niveau d'expérience, horaires
5. **Mode hors ligne** : Fonctionnalités basiques disponibles sans connexion (PWA)

Moyen terme (6-12 mois) :

1. **Application mobile native** : Ionic ou React Native pour iOS/Android
2. **Intégration avec BoardGameGeek API** : Base de données de jeux
3. **Système de réputation** : Badges, points, gamification
4. **Recommandations personnalisées** : ML pour suggérer sessions et jeux
5. **Gestion de ludothèque personnelle** : "Ma collection de jeux"
6. **Calendrier personnel** : Export iCal, Google Calendar

Long terme (1-2 ans) :

1. **Monétisation** : Modèle freemium avec fonctionnalités premium
2. **Partenariats** : Collaboration avec bars à jeux, éditeurs
3. **Marketplace** : Vente/échange de jeux d'occasion entre utilisateurs
4. **Tournois** : Organisation d'événements compétitifs
5. **Version internationale** : i18n, adaptation à d'autres pays

Améliorations techniques :

1. Analytics détaillés (Google Analytics, Plausible)
2. A/B testing pour optimiser l'UX

4.4. Bilan personnel

Ce projet m'a permis de continuer à apprendre Angular, m'entraîner à développer des API REST avec NestJS, et utiliser les outils de Supabase.

Références bibliographiques

Documentation officielle :

1. **Angular** Angular.io. (2024). *Angular Documentation*. Récupéré de <https://angular.io/docs>
2. **NestJS** NestJS.com. (2024). *NestJS - A progressive Node.js framework*. Récupéré de <https://docs.nestjs.com>
3. **Supabase** Supabase.com. (2024). *Supabase Documentation*. Récupéré de <https://supabase.com/docs>
4. **Prisma** Prisma.io. (2024). *Prisma ORM Documentation*. Récupéré de <https://www.prisma.io/docs>
5. **Nx** Nx.dev. (2024). *Nx: Smart Monorepos · Fast CI*. Récupéré de <https://nx.dev/getting-started/intro>
6. **Leaflet** Leafletjs.com. (2024). *Leaflet - an open-source JavaScript library for interactive maps*. Récupéré de <https://leafletjs.com>
7. **OpenStreetMap** OpenStreetMap.org. (2024). *OpenStreetMap Wiki*. Récupéré de <https://wiki.openstreetmap.org>

Articles et ressources :

1. Gus&Co. (2023). *Statistiques des jeux de société 2023 : une passion planétaire*. Récupéré de <https://gusandco.net/2023/11/22/statistiques-jeux-de-societe-2023/>
2. Le Figaro. (2025). *1200 nouveautés chaque année : la surproduction de jeux de société met-elle le secteur en péril ?* Récupéré de <https://www.lefigaro.fr/conso/surproduction-jeux-de-societe>
3. State of JS. (2024). *The State of JavaScript 2024: Front-end Frameworks*. Récupéré de <https://2024.stateofjs.com/en-US/libraries/front-end-frameworks/>

Livres et cours :

1. Become a ninja with Angular

Outils et bibliothèques :

1. **TypeScript** Typescriptlang.org. (2024). *TypeScript Documentation*. Récupéré de <https://www.typescriptlang.org/docs/>
2. **RxJS** RxJS.dev. (2024). *RxJS Documentation*. Récupéré de <https://rxjs.dev/guide/overview>
3. **Jest** Jestjs.io. (2024). *Jest - Delightful JavaScript Testing*. Récupéré de <https://jestjs.io/docs/getting-started>
4. **Playwright** Playwright.dev. (2024). *Playwright Test*. Récupéré de <https://playwright.dev>
5. **Prisma ERD Generator** Github.com. (2024). *prisma-erd-generator*. Récupéré de <https://github.com/keonik/prisma-erd-generator>
6. **Mermaid** Mermaid.js.org. (2024). *Mermaid - Diagramming and charting tool*. Récupéré de <https://mermaid.js.org>

Annexes

Wireframes

Fig. 1 Wireframe, page des lieux de barades.com

The wireframe shows a search interface for game locations. At the top, there are navigation links: Bar à Dés, Sessions, Lieux, Groupes, and two buttons: Se connecter and S'inscrire. Below this is a search form titled 'Lieux de jeu' with fields for 'Recherche' (Nom, adresse, ville...) and 'Type de lieu' (Tous les types). There are also filters for 'Équipements': WiFi, TABLES DE JEU, NOURRITURE, BOISSONS, and PARKING. A large map area is centered below the search form. To the right, a box titled 'Détails du lieu' contains a placeholder 'Selectionnez un lieu sur la carte' with a note: 'Cliquez sur un indicateur ou sur une entrée de la carte pour afficher ses informations détaillées ici.' A small info icon is inside the box.

The homepage features a main title 'Votre Prochaine Aventure Commence Ici' and a subtitle 'Trouvez des joueurs, des maîtres de jeu et des lieux pour vos parties de jeu de rôle, où que vous soyez.' Below this are two search input fields: 'Nom du jeu, univers' and 'Ville ou en ligne', followed by a 'Rechercher' button. A 'Bar à Dés' navigation bar is at the top, and 'Se connecter' and 'S'inscrire' buttons are at the top right.

This section displays three upcoming events: 'Catan' (1 place(s) disponible(s)), 'Texas Hold'em Poker' (3 place(s) disponible(s)), and 'Catan' (2 place(s) disponible(s)). Each event card includes a checkbox for date/time, player names, and level, along with 'Réserver ma place' and 'Voir détails' buttons. A 'Voir toutes les parties' link is located above the third card. Below this is a box titled 'Prêt à organiser votre propre partie ?' with a note: 'Créez une session, invitez des joueurs et vivez une aventure inoubliable.' It contains 'Créer une session' and 'Rejoindre un groupe' buttons.

The footer is divided into four columns: 'Bar à Dés' (À propos, Nous contacter, Carrières), 'Communauté' (Charte, Forum, Discord), 'Ressources' (Aide, Devenir partenaire), and 'Développeurs' (GitHub, Composants UI, API).

Fig.2 Wireframe, page d'accueil de barades.com

Layouts

Fig. 3 Layout, page d'accueil de barades.com

The screenshot shows the homepage of Barades.com. At the top, there's a search bar with fields for 'Nom du jeu, ville...' and 'Ville ou en ligne' with a placeholder 'Bar à dés'. Below the search bar is a button labeled 'Rechercher'. The main content area has a dark background with white text. It features a section titled 'Parties à la Une' with three cards: 'Edge Cases', 'Casual Board Games', and 'Brussels Adventurers Guild'. Below this is a section titled 'Prêt à organiser votre propre partie ?' with a card for creating a new session. At the bottom, there are navigation links for 'Bar à Dés', 'Sessions', 'Lieux', and 'Groupes', along with social media icons.

Fig.4 Layout, page des groupes de barades.com

This screenshot shows the 'Groupes de jeu' (Groups) page. It features a header with tabs for 'BAR à DÉS', 'Sessions', 'Lieux', and 'Groupes'. A search bar and a 'Rechercher' button are at the top right. The main content area displays three group profiles: 'Edge Cases', 'Casual Board Games', and 'Brussels Adventurers Guild'. Each profile includes a thumbnail, the group name, a brief description, and two buttons: 'Voir les détails' and 'Répondre'. Below the groups, there are four categories: 'Bar à Dés', 'Communauté', 'Ressources', and 'Développeurs', each with a small description and a link. The footer contains copyright information and social media links.

This screenshot shows the 'Lieux de jeu' (Locations) page. It features a large map of a city area with various landmarks and roads. Overlaid on the map is a search interface with a central modal window. The modal has a title 'Lieux de jeu' and a subtitle 'Trouvez l'endroit idéal pour vos sessions'. It includes a search bar with placeholder 'Nom, adresse, ville...', a dropdown menu for 'Type de lieu' set to 'Tous les types', and several filter buttons for 'WIFI', 'TABLES DE JEU', 'NOURRITURE', 'BOISSONS', and 'PARKING'. A location pin is visible on the map. To the right of the map, there's a sidebar with a button 'Liste des lieux (2 / 2)' and a section titled 'DÉTAILS DU LIEU' with the instruction 'Sélectionnez un lieu sur la carte'. The footer of the page includes a copyright notice and a link to 'Leaflet | © OpenStreetMap contributors'.

Fig.5 Layout, page des lieux de barades.com

ERD

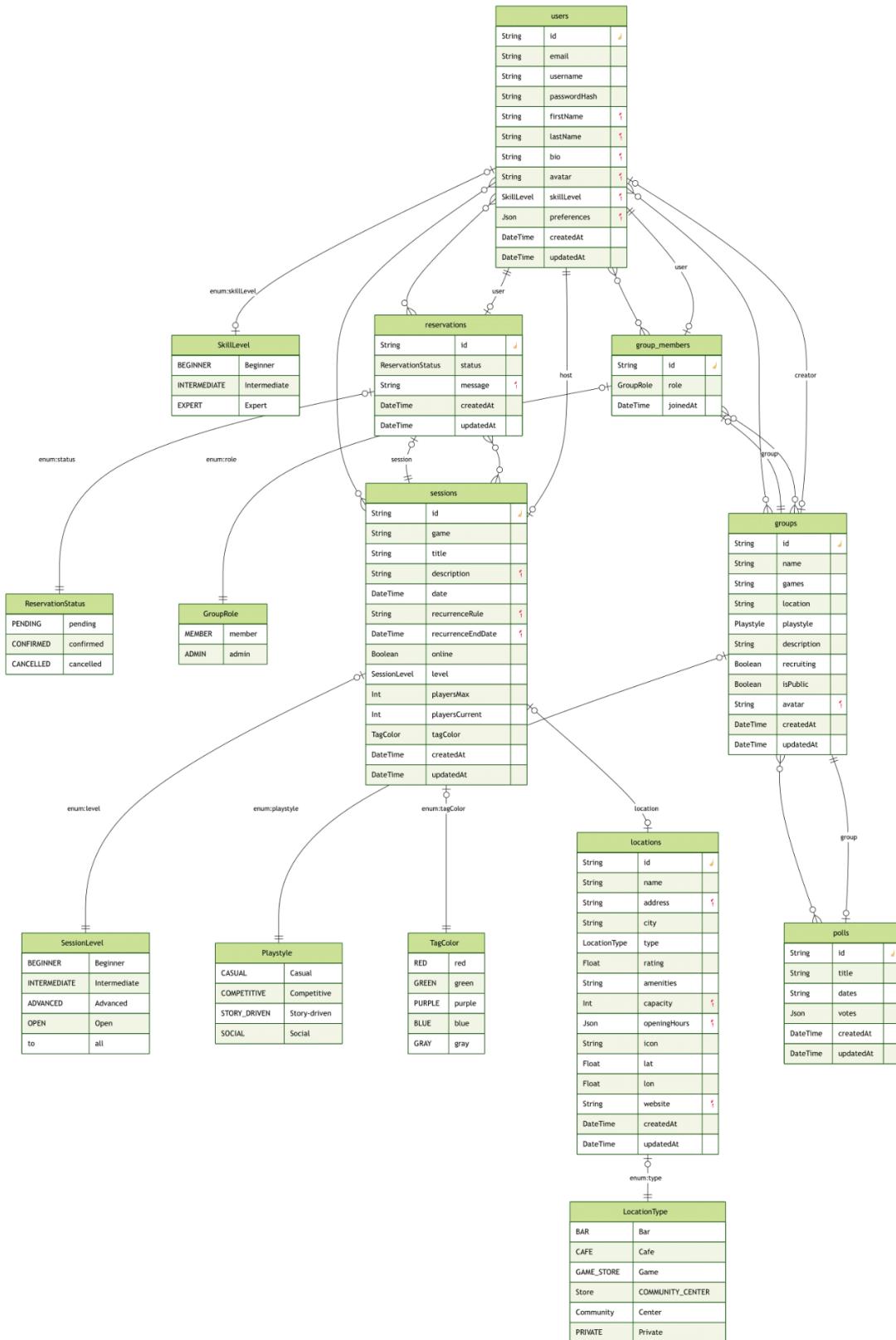


Fig. 6 ERD diagram de la base de données de barades.com