

MAI 2022



# **DUST YOUR GAMES**

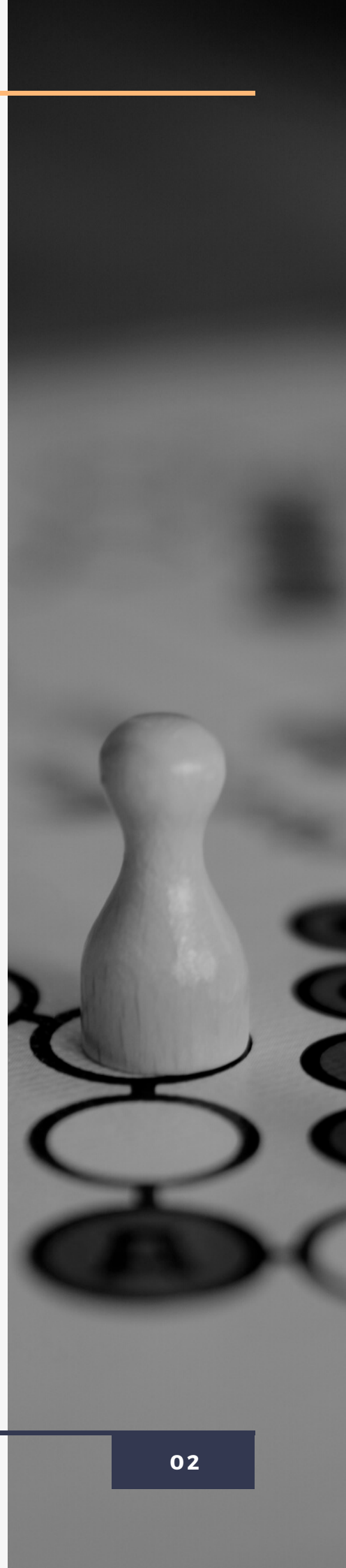
## **PRÉPARÉ PAR :**

MANUEL S  
MATTHIEU M  
MEHDI B  
MEHDY D  
ALICE C

---

# SOMMAIRE

LE PROJET	03
LA CIBLE	03
L'ÉQUIPE	04
FONCTIONNALITÉS	05
MVP	05
EVOLUTIONS À VENIR	07
TECHNOS UTILISÉES	08
DÉVELOPPEMENT FRONT-END	08
DÉVELOPPEMENT BACK-END	09
NAVIGATEURS COMPATIBLES	10
USER STORIES	11
PAGE CONNEXION/INSCRIPTION	11
PAGE TABLEAU DE BORD	11
PAGE LISTE DE JEUX	12
PAGE COMPTE	12
PAGE CONTACT	13
L'ARBORESCENCE	14
LISTE DES ROUTES	15
ROUTES FRONT-END	15
ROUTES BACK-END - API ENDPOINTS	18
MODÈLES DES DONNÉES	19
MODÈLE CONCEPTUEL DES DONNÉES	19
MODÈLE LOGIQUE DES DONNÉES	20
CHARTES GRAPHIQUES	21
CODES COULEURS	21



# LE PROJET

À quoi allons-nous jouer ce soir ?  
Qu'avons-nous sur nos étagères ?

Depuis combien de temps ce jeu prend la poussière ? Autant de questions qui ne facilitent pas la vie des joueurs de jeux de société.

C'est pourquoi aujourd'hui, nous vous proposons **Dust Your Games** : l'application qui va **révolutionner notre façon d'organiser et d'utiliser nos jeux**.

Dust your Games est une ludothèque personnelle, un moyen de répertoire virtuellement nos jeux, mais pas seulement :

Depuis notre application, nous souhaitons pouvoir trouver rapidement un jeu adapté pour toutes situations : que vous ayez 5 minutes ou 3 heures, que vous soyez jeune ou moins jeune, que vous soyez un, deux, trois, douze... !

Avec Dust Your Games, dépoussiérons nos jeux en un seul clic.

Fini les jeux laissés derrière, vous saurez quand vous aurez joué à votre jeu favori et lesquels commencent à prendre la poussière. Fini aussi les heures passées à décider quel jeu sortir.

Grâce à notre fonction "Dépoussiérage" et à la magie de notre algorithme, vous aurez plus de chance de déterrer des jeux que vous n'aurez pas vu depuis des années et qui correspondent à vos envies du moment.

Connectons nos ludothèques en ajoutant des amis, nous savons enfin qui possède tel jeu ou telle extension pour être facilement complémentaire, ou bien pour les plus généreux faire des cadeaux afin de compléter la collection de vos amis !

## LA CIBLE

Dust your Games est une application tout public qui s'adresse aux jeunes comme aux grands, aimant passer des moments conviviaux en famille ou entre amis.

Les utilisateurs de l'application aiment l'ambiance des jeux de société et le partage. Notre application est ludique, facile à utiliser et interactive pour amuser notre cible.

L'âge requis pour la création d'un compte est fixé à 13 ans.



---

# L'EQUIPE

**MATTHIEU M.**

24 ANS

PRODUCT OWNER  
LEAD-DEV FRONT  
DEVELOPPEUR REACT

JEU FAVORI : ROOT

**ALICE C.**

29 ANS

SCRUM MASTER  
DEVELOPPEUSE REACT

JEU FAVORI : BLANC MANGER COCO

**MANUEL S.**

33 ANS

RÉFÉRENT TECHNIQUE  
DEVELOPPEUR REACT

JEU FAVORI : PICTIIONARY

**MEDHY D.**

33 ANS

LEAD-DEV BACK  
DEVELOPPEUR SYMFONY

JEU FAVORI : TRIVIAL POURSUIT

**MEHDI B.**

28 ANS

RÉFÉRENT GIT  
DEVELOPPEUR SYMFONY

JEU FAVORI : DARKEST DUNGEON



---

# FONCTIONNALITÉS

## MVP

### Gestion de l'authentification

La gestion de l'authentification se fera par le biais de l'utilisation d'un formulaire d'inscription pour la création d'un compte utilisateur nécessitant un pseudo, l'année de naissance (pour vérification > 13 ans), un e-mail ainsi qu'un mot de passe. Elle se fera également via un formulaire de connexion, donnant accès à la fonctionnalité du site et au profil de l'utilisateur. Ce dernier pourra gérer son compte en modifiant ses informations ou en le supprimant.

### Tableau de bord

Il s'agit là de la page d'accueil des utilisateurs connectés. Elle permettra de naviguer rapidement vers trois des fonctionnalités, la liste de jeux, la fonctionnalité de dépoussiérage et, à terme, la liste d'amis.

De plus, on aura une section contenant divers widgets disponibles sur cette page. Le premier widget disponible au lancement (en MVP) sera un affichage de trois jeux aléatoires. Cette section sera évolutive car elle changera avec les versions du projet.

### Liste des jeux

L'utilisateur a la possibilité d'ajouter ou de supprimer une sélection de jeux fournis par une API extérieure. Cette liste est liée à l'utilisateur et est stockée dans notre base de données depuis notre API. Un affichage précis de la liste des jeux avec une image est proposé avec une différenciation visuelle du statut du jeu (vient de sortir / normal / pas sorti depuis un moment).

De plus, l'utilisateur a la possibilité d'effectuer des tris sur sa liste de jeux selon plusieurs critères (nombre de joueurs, âge minimum, type de jeu, durée, etc.).

### Description d'un jeu

Chaque jeu aura une description associée contenant les informations de celui-ci : son titre, son nombre de joueurs minimum, sa durée, l'âge minimum requis.

### Navigation

L'utilisateur doit pouvoir naviguer entre plusieurs pages : accueil, ludothèque, compte et contact. Pour cela un menu, accessible d'un clic sur le bouton correspondant, pourra être déroulé de n'importe où.

---

# FONCTIONNALITÉS

## Compte

L'utilisateur doit pouvoir gérer son compte avec une image, ses informations, son login et mot de passe mais aussi supprimer potentiellement le compte au besoin.

## Tirage au sort - Dépoussiérage

La fonctionnalité phare de Dust your Games est de pouvoir facilement tirer au sort un jeu d'une liste. Depuis le tableau de bord, aucun critère particulier est imposé, le tirage s'effectue donc directement sur la liste complète de l'utilisateur. Depuis la liste de jeux, les critères de tri sélectionnés permettent d'affiner la liste des jeux pour le tirage. De plus, si l'utilisateur le veut, il pourra manuellement sélectionner ou désélectionner des jeux avant de lancer son tirage.

Par défaut, chaque jeu a une probabilité égale d'être tiré au sort. Toutefois, si un jeu n'est pas sorti depuis un certain temps, alors son poids dans le tirage augmente et il aura ainsi plus de chance d'être tiré. De même, si un jeu vient de sortir, il aura moins de chance d'être tiré.

L'algorithme devra prévoir les cas où plusieurs listes d'utilisateurs sont sélectionnées et faire un pré-tirage parmi ces listes afin d'équilibrer les chances de chaque liste. (voir cas concret).

## Poids d'un jeu

Un jeu possède une note, une probabilité et un poids invisible qui lui est lié. Ce poids peut varier entre 0 et 10. Le poids par défaut étant 5, un jeu à zéro ne sera pas tiré au sort, de 1 à 4 il aura une faible probabilité tandis qu'il aura plus de chance d'être tiré avec un poids de 6 à 10.

### Cas concret

Axelle et Martin souhaitent sélectionner un jeu au hasard dans leur ludothèque respective. Ils appliquent le critère de sélection 'jeu d'une durée supérieur à 30 minutes' et sélectionnent leurs deux listes. Martin a 15 jeux qui remplissent ce critère, alors qu'Axelle n'en a que 6. Un pré-tirage invisible tire 6 jeux au hasard dans la liste de Martin avant d'effectuer le tirage final en regroupant les 6 d'Axelle. Ainsi, parmi les 12 jeux il y a autant de chances pour qu'un jeu d'Axelle ou de Martin soit tiré.

Le tirage final utilise les 12 jeux, prend en compte le poids des jeux et en sélectionne un. Si Axelle et Martin acceptent ce tirage, le jeu est considéré comme fraîchement sorti et ajuste donc son poids, sinon un autre tirage peut être effectué.

---

# FONCTIONNALITÉS

## Évolutions à venir

### Gestion des amis

L'utilisateur peut ajouter ou supprimer d'autres utilisateurs dans sa liste d'amis. Ainsi il aura accès à leurs listes de jeux mais pourra aussi inclure leurs listes dans les autres fonctionnalités de l'application.

### Gestion des utilisateurs — Back-office

L'ajout d'un back-office est envisageable dans le futur. Il permettra de monitorer les utilisateurs, voir leurs listes et potentiellement les supprimer.

### Notation

L'utilisateur pourra attribuer une note à ses jeux. Cette note sera disponible dans le tri des jeux et pourra, si sélectionnée, être considérée lors du tirage.

### Jeux non possédés

Des jeux pourront être ajoutés à une liste "Jeux non possédés".

### Création de liste personnalisée

L'utilisateur pourra créer des listes manuellement et y ranger divers jeux. On imagine potentiellement une "wish-list" dans laquelle il rangerait les jeux qu'il ne possède pas mais qu'il désire.

Ajout d'une page listant des jeux pouvant se jouer depuis une page web / applications :  
exemple : <https://kgames.fr/>

---

# TECHNOS UTILISÉES



## Développement Front-end

### Les bases

**Yarn :** C'est un gestionnaire de packages qui se double d'un gestionnaire de projets. Utilisé en même temps que npm (node package manager), il nous permet d'installer toutes les dépendances utiles au projet. Celles-ci seront gérées depuis le fichier package.json.

**HTML/CSS/JS :** Les trois langages d'une page web. Ils seront utilisés à travers tout le projet, parfois par des extensions syntaxiques comme JSX pour l'HTML/le JS ou alors avec une forme alternative comme le SCSS pour le CSS.

La mise en production de notre site transpilera l'intégralité de notre code dans ce trio pour qu'il soit lisible par les navigateurs.

**Create React App (CRA) :** C'est un package installé depuis npm. Il permet l'initialisation d'un projet React avec une configuration moderne testée, approuvée et maintenue.

### Les dépendances

**React :** C'est une bibliothèque JavaScript créée par Facebook, elle sera la base de notre projet pour la partie front.

**Redux :** C'est une bibliothèque JavaScript open-source pour gérer l'état de l'application. Redux est couramment utilisé avec React pour créer des interfaces utilisateur.

**React-redux :** C'est la couche officielle de liaisons de l'interface utilisateur React pour Redux. Il permet à nos composants React de lire les données d'un magasin Redux et d'envoyer des actions au magasin pour actualiser le state.

**React Router Dom :** React Router DOM est un package qui va nous permettre d'implémenter un routage dynamique dans une application Web. On utilise le processus de Routing qui permet de changer le contenu dynamiquement en fonction de l'URL depuis une page unique.

**Axios :** En quelques mots, Axios est une bibliothèque Javascript utilisée pour effectuer des requêtes HTTP à partir du navigateur qui prend également en charge l'API ES6 promise. Il nous servira surtout pour communiquer avec l'API produite par l'équipe Symfony.

**ESLint :** C'est un Linter qui nous permet d'écrire un code JS avec une syntaxe propre et suivant des conventions largement adoptées.

**Babel :** Inclus avec CRA, c'est un transpileur qui va traduire notre code dans un Javascript adapté et compréhensible par les navigateurs.

**SASS :** Nécessaire pour utiliser le SCSS, c'est un préprocesseur qui, comme Babel, traduira notre code en CSS lisible par les navigateurs.



---

# TECHNOS UTILISÉES



## Développement Back-end

### Les bases

**Symfony** : Il s'agit d'un ensemble de composants PHP ainsi qu'un Framework MVC libre écrit en PHP. Il sera la base de la partie Back de notre projet.

**Doctrine** : Cette ORM permet de faire le lien entre les objets et les éléments d'une base de données

**MariaDB** : Dérivée de MySQL, MariaDB est un système de gestion de base de données édité sous licence GPL.

### Les bundles

**JWT Token (Identification & sécurisation API)** : il offre la possibilité d'utiliser les JSON Web Tokens afin de protéger les ressources de l'API REST. Lorsque l'utilisateur s'authentifie correctement, un token lui sera retourné. Ce token sera ensuite transmis au serveur afin de permettre à l'utilisateur de lire ou d'écrire les ressources protégées.

**Bundle NelmioCorsBundle** : Le cors-bundle permet de définir les règles CORS. Ce bundle permet de définir les domaines qui auront accès à votre API REST. Dans notre cas, l'API et le front n'étant pas hébergés sur le même serveur, nous devons lui ouvrir les "portes".

**Bundle symfony/filesystem** : Le composant Symfony Filesystem fournit des méthodes d'encapsulation utiles qui facilitent l'interaction des fichiers. Pour notre projet, son utilité réside dans notre service PicturesManager qui a pour but d'ajouter ou supprimer une image et par la même occasion lors d'un remplacement d'image, le service s'occupera de supprimer l'ancienne image.

**DoctrineExtensions Beberlei bundle** : il ajoute des fonctions supplémentaires pouvant être fournies par la DB. Dans notre cas, il sera utilisé afin de récupérer la fonction RAND qui nous sert pour génération des produits aléatoires.

**Bundle Faker & Bundle Fixtures** : Ces bundles permettent de créer des données fictives rapidement afin d'avoir un contenu exploitable durant la phase de développement et de test.



---

# NAVIGATEURS COMPATIBLES

Nous allons utiliser les normes proposées par la configuration de Create-React-App.

Par défaut, le projet généré prend en charge tous les navigateurs modernes. De plus, avec cette configuration qui contrôle la façon dont Babel transpile JavaScript pour qu'il soit compatible avec les navigateurs spécifiés, Dust your Games sera compatible avec 99.8 % des navigateurs encore actifs.

Nous n'établirons rien pour rendre notre application compatible avec : Opera Mini, Internet Explorer 9, 10, et 11.

---

# USER STORIES

## Page Connexion/Inscription

En tant que...	Je veux pouvoir...	Afin de/que...
Visiteur	Accéder au formulaire d'inscription	Afficher le formulaire
Visiteur	Soumettre le formulaire d'inscription	Créer mon compte utilisateur afin d'utiliser les services du site
Visiteur	Accéder au formulaire de connexion	M'identifier et récupérer mes listes personnelles
Visiteur	Accéder à la page de contact	Contacteur l'équipe du site

## Page Tableau de bord

En tant que...	Je veux pouvoir...	Afin de/que...
Utilisateur	Accéder à ma liste de jeux	Consulter ma liste de jeux
Utilisateur	Accéder à la fonctionnalité de tirage au sort	Tirer un jeu au sort sans critère

---

# USER STORIES

## Page Liste de jeux

En tant que...	Je veux pouvoir...	Afin de/que...
Utilisateur	Ajouter un jeu à ma liste	Remplir ma liste et utiliser les autres fonctionnalités
Utilisateur	Trier mes jeux selon plusieurs critères	Affiner l'affichage et sélectionner les jeux pris en compte lors du tirage
Utilisateur	Affiner la sélection jeu par jeu	Retirer certains jeux pour le tirage
Utilisateur	Effectuer un tirage	Sélectionner un unique jeu de ma liste avec les critères actuels
Utilisateur	Accéder à la description d'un jeu	Connaitre ses caractéristiques

---

# USER STORIES

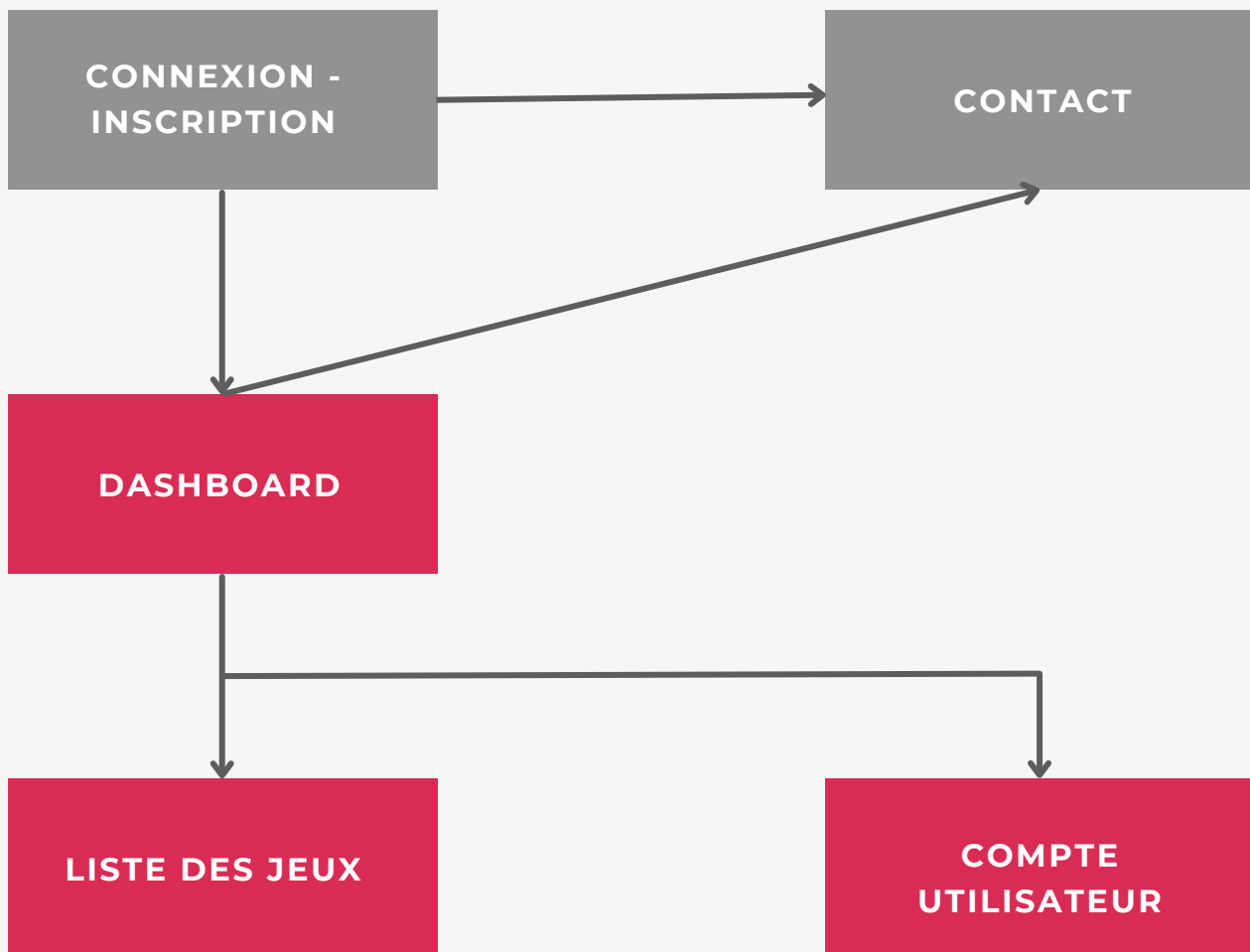
## Page Compte

En tant que...	Je veux pouvoir...	Afin de/que...
Utilisateur	Accéder aux informations de mon compte	-
Utilisateur	Modifier mes informations	-
Utilisateur	Supprimer mon compte	Retirer mon compte de la BDD

## Page Contact

En tant que...	Je veux pouvoir...	Afin de/que...
Visiteur	Soumettre le formulaire de contact	Contacteur l'équipe du site
Visiteur	Voir qui compose l'équipe	-

# L'ABORESCENCE



UTILISATEUR  
NON CONNECTÉ



UTILISATEUR  
CONNECTÉ

---

# LES ROUTES

## Routes Front-end

Page : Connexion

URL : ... /

**Accessible pour :** Utilisateur non connecté

**Contenu :** Layout en deux sections (gauche et droite sur desktop) :

- Une section présentation rapide de l'application avec illustration.
- L'autre section contenant un unique formulaire de connexion, un lien 'mot de passe oublié ?' et un bouton distinctif visant à créer un compte.
- Un footer avec le plan du site et les liens utiles

**Particularité :** Le lien pour le mot de passe oublié ainsi que le bouton de création de compte ouvre une fenêtre modale avec le formulaire correspondant.

Adresse e-mail ou numéro de tél.

Mot de passe

**Se connecter**

Mot de passe oublié ?

---

**Créer nouveau compte**

---

# LES ROUTES

## Routes Front-end

Page : Tableau de bord

URL : ... /

**Accessible pour :** Utilisateur connecté

**Contenu :** Layout vertical

- Header Global : Composé du logo et d'un bouton affichant le menu au clic (type burger)
- Une section servant de navigation direct composé de trois cartes :
- Liste des jeux – Tirage au sort – Liste d'amis
- Un autre section évolutive avec un affichage de trois jeux aléatoires
- Un footer avec le plan du site et les liens utiles

**Particularité :** La carte 'Tirage au sort' lance directement un tirage sur la liste complète de l'utilisateur. L'affichage du tirage s'effectue dans un modal avec animation et résultat.

Page : Liste des jeux

URL : ... /games

**Accessible pour :** Utilisateur connecté

**Contenu :** Layout grilles

- Header Global : Composé du logo et d'un bouton affichant le menu au clic (type burger)
- Un menu flottant permettant d'effectuer les différents filtres proposé par l'application
- Un bouton à gauche permettant l'ajout d'un jeu, et un à droite permettant le tirage
- Une liste de jeux sous forme de carte avec image et nom. Une différenciation visuelle entre les jeux fraîchement sortis et les jeux "poussiéreux"
- Un footer avec le plan du site et les liens utiles

**Particularité :** Chaque carte de jeu ouvre une modale pour l'affichage des détails. De même, le bouton d'ajout affiche aussi une modale



---

# LES ROUTES

## Routes Front-end

Page : Contact

URL : ... /contact

**Accessible pour :** Tout utilisateur

**Contenu :** Layout vertical

- Header Global : Composé du logo et d'un bouton affichant le menu au clic (type burger)
- Section de présentation de l'équipe
- Section avec formulaire de contact
- Un footer avec le plan du site et les liens utiles

**Particularité :** L'état de connexion de l'utilisateur affecte le menu du header (bouton connexion ou déconnexion)

Page : Compte

URL : ... /account

**Accessible pour :** Utilisateur connecté

**Contenu :** Layout vertical

- Header Global : Composé du logo et d'un bouton affichant le menu au clic (type burger)
- Carte de contact avec en haut la photo de profil (possibilité de la modifier au survol), pseudo, nombre de jeu et enfin l'email
- Un séparateur
- Bouton édition ouvrant une modale pour la modification des informations
- Bouton de suppression de compte
- Un footer avec le plan du site et les liens utiles
- 

**Particularité :** Demande de confirmation pour la suppression du compte

---

# LES ROUTES

## Routes Back-end - API Endpoints

### Endpoints : User

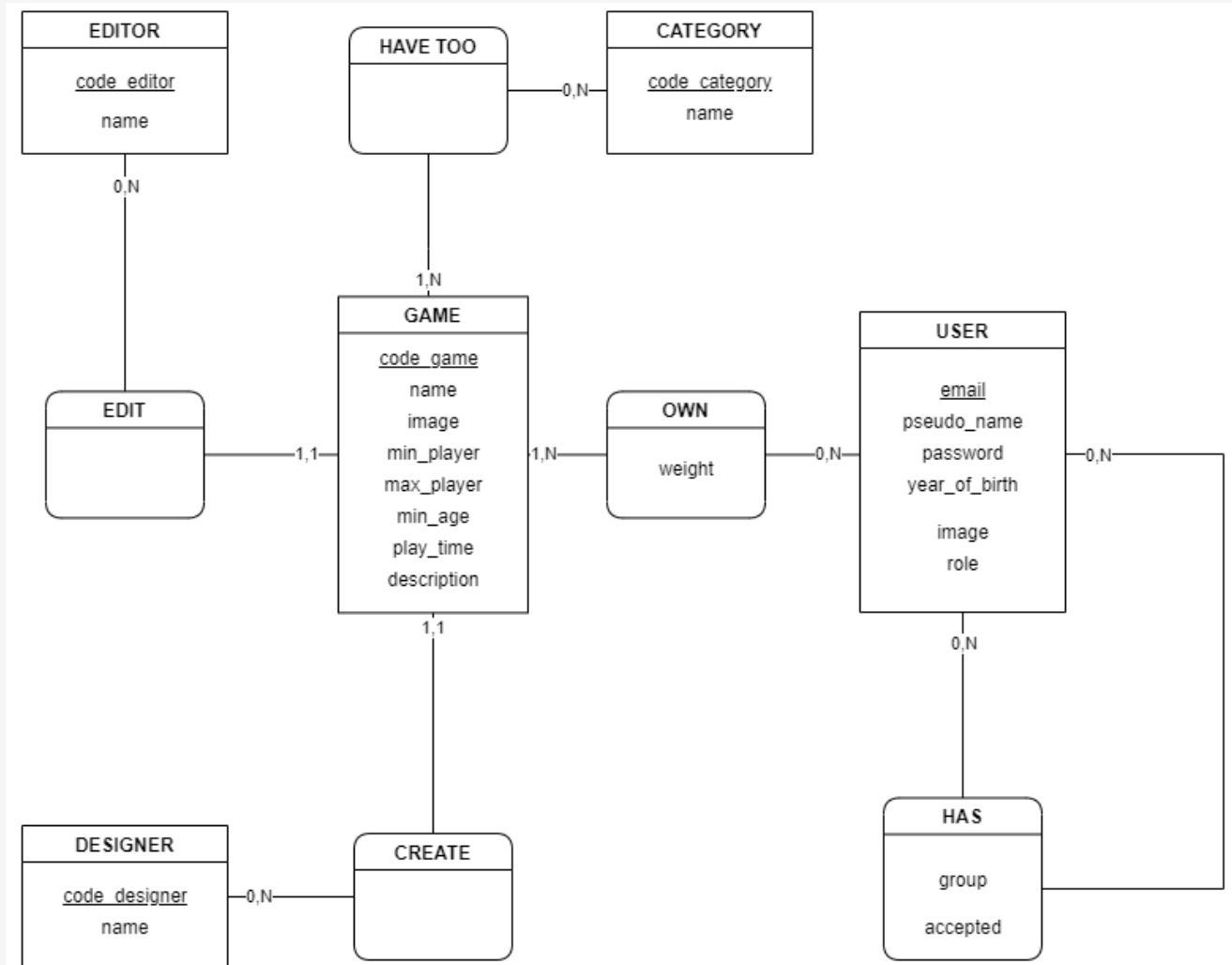
URL	Méthode http	Contrôleur	Méthode	Commentaire
/login	POST			Envoi formulaire de login
/register	POST			Envoi formulaire d'enregistrement utilisateur
/profil	GET			Affiche le profil de l'utilisateur connecté
/profil/edit	PATCH			Edition du profil de l'utilisateur connecté
/profil/delete	DELETE			Suppression du profil de l'utilisateur connecté

### Endpoints : Games list

URL	Méthode http	Contrôleur	Méthode	Commentaire
/games	GET			Récupération des jeux
/games/{id}	GET			Récupération d'un jeu
/games/categories	GET			Récupération des catégories
/games/categories/{id}	GET			Récupération d'une catégorie
/game/add	POST			Ajout d'un jeu à la liste de l'utilisateur

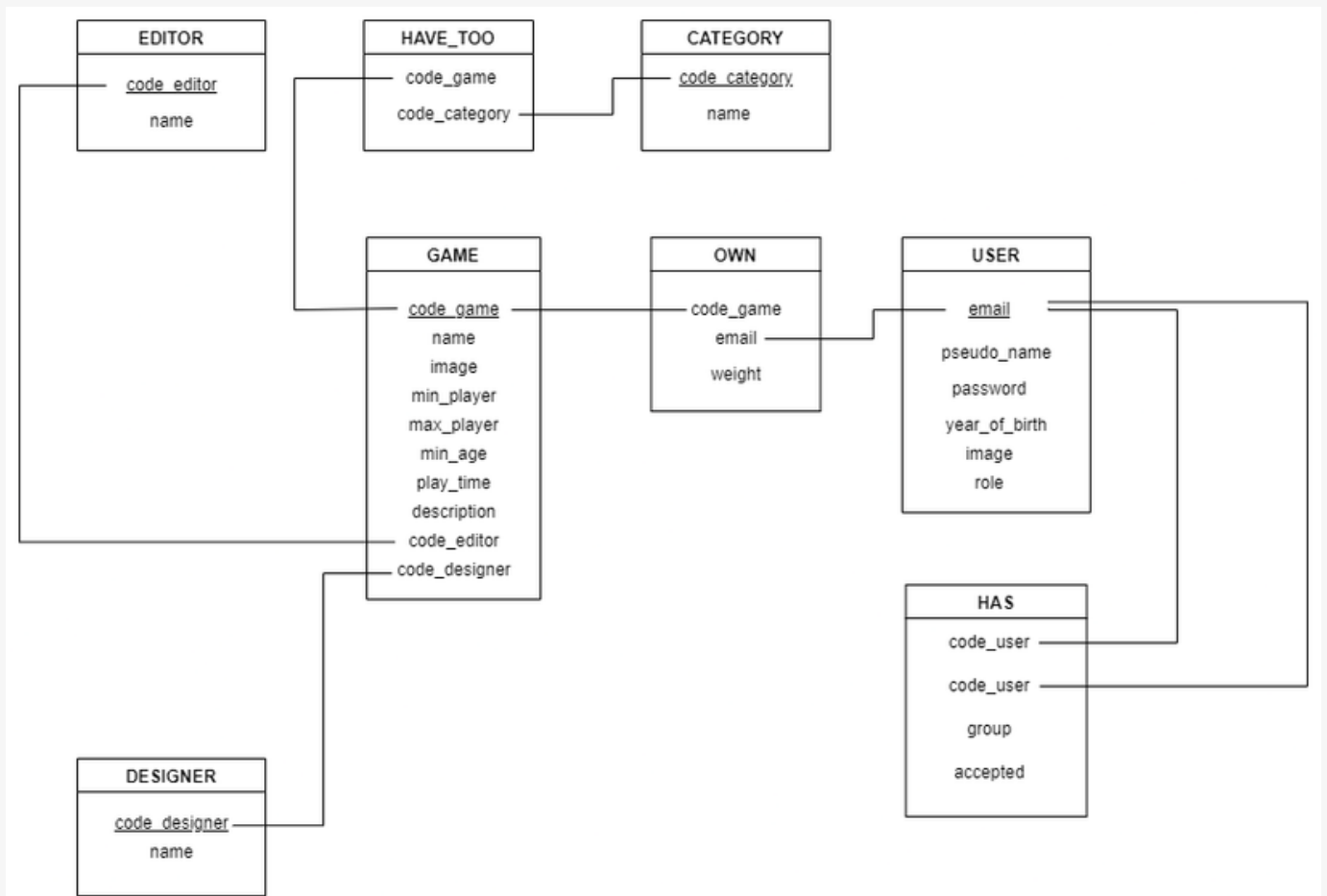
# MODÈLES DES DONNÉES

## Modèle conceptuel des données - MCD



# MODÈLES DES DONNÉES

## Modèle logique des données - MLD



---

# CHARTE GRAPHIQUE

## Les codes couleurs



#D82C55



#5D5D5D



#FFB877



#FFDFC1



#333850