

# Cahier des charges - Application web de gestion de collection de jeux de société

HEIG-VD – PDG 2023

BASTIAN CHOLLET, KEVIN JORAND, KEVIN FERATI, FLAVIO SOVILLA

## Table des matières

Table des matières .....	1
1 Introduction .....	2
2 Fonctionnalités principales .....	2
2.1 Recherche de jeux .....	2
2.2 Connexions et inscriptions .....	2
2.3 Gestion de la collection personnelle .....	2
2.4 Création de jeux .....	2
2.5 Création de soirées .....	3
2.6 Invitations et votes .....	3
3 Fonctionnalités additionnelles .....	3
3.1 Liste de contact .....	3
3.2 Votation en glissant .....	3
3.3 Interactions sociales avancées .....	3
4 Exigences techniques .....	4
4.1 Plateforme et technologie .....	4
4.2 Données des jeux .....	4
4.3 Base de données .....	4
4.4 Sécurité .....	4
4.5 Hébergement .....	4
5 Contraintes de développement .....	4
5.1 Stratégie de tests .....	4
5.2 Intégration et déploiement continu .....	4
6 Livrables .....	5

# 1 Introduction

L'objectif de ce cahier des charges est de définir une application web permettant la gestion de collection de jeux de société par des utilisateurs enregistrés ou non. Les utilisateurs devront également pouvoir organiser des soirées en invitant d'autres utilisateurs à voter pour une liste de jeux proposés pour ladite soirée. Il s'agira ici d'énoncer les diverses fonctionnalités souhaitées ainsi que les exigences techniques demandées, qu'elles soient de type architecturales, DevOps ou stratégies de test. Nous évoquerons également les livrables attendus en fin de projet.

Ce projet est réalisé dans le cadre du cours « Projet de Groupe » dispensé à la HEIG-VD.

## 2 Fonctionnalités principales

Pour des raisons de simplification d'écriture, nous raccourcirons le termes « jeu de société » par « jeu ».

### 2.1 Recherche de jeux

- Les utilisateurs, qu'ils soient connectés ou non doivent pouvoir chercher un jeu au travers d'une barre de recherche accessible en tout temps offrant des propositions sur la base de ce que saisi l'utilisateur. Par exemple « Loup » proposerait à l'utilisateur « Loups-garous de Thiercelieux » et « Loup-garou pour une nuit ».
  - Si l'utilisateur sélectionne une proposition auto-complétée, l'application redirige sur une page détaillée du jeu contenant les informations clés le concernant (Nombre de joueurs, durée d'une partie, descriptif du jeu, image du jeu, etc...).
  - Si l'utilisateur valide sa recherche sans sélectionner une proposition auto-complétée, une page avec divers résultats correspondant à l'entrée de l'utilisateur s'affiche. Choisir un élément de la liste affiche le détail précédemment décrit.

### 2.2 Connexions et inscriptions

- Les utilisateurs doivent pouvoir créer un compte ou se connecter à l'application.
- Les informations nécessaires pour l'inscription doivent inclure un nom d'utilisateur unique, une adresse e-mail valide et un mot de passe sécurisé.
- L'application doit prendre en charge la récupération du mot de passe en cas d'oubli.
- Les utilisateurs inscrits doivent valider leur inscription au travers d'un lien unique reçu par mail.

### 2.3 Gestion de la collection personnelle

- Les utilisateurs connectés doivent pouvoir ajouter des jeux de société à leur collection personnelle.
- Les jeux ajoutés à la collection doivent être sauvegardés avec leurs informations complètes et les utilisateurs doivent pouvoir les consulter ultérieurement.

### 2.4 Création de jeux

- Les utilisateurs connectés doivent avoir la possibilité de créer des nouveaux jeux en fournissant des informations le concernant. Les informations obligatoires et minimales requises sont :
  - Le titre du jeu
  - Le nombre de joueur min et max
  - L'âge minimal requis

- Tout jeu créé par un utilisateur s'ajoute à sa collection personnelle. **Ils ne doivent pas être retournés par la recherche.**

### 2.5 *Création de soirées*

- Les utilisateurs connectés doivent pouvoir créer des soirées où ils pourront proposer des jeux à d'autres utilisateurs.
- Les soirées doivent inclure les détails suivants :
  - Titre
  - Date
  - Heure
  - L'emplacement peut être indiqué ou non sous forme d'un champs texte libre.
- Par défaut, seuls les jeux de la collection du créateur de la soirée peuvent être proposés.
- Lors de la création de la soirée, il est possible d'accorder le droit aux invités enregistrés de proposer les jeux de leur propre collection.
- Il doit être possible de modifier une soirée en tout temps ainsi que de la supprimer
- Une soirée est supprimée automatiquement 30 jours après sa réalisation.

### 2.6 *Invitations et votes*

- Les utilisateurs connectés doivent pouvoir inviter d'autres utilisateurs à participer à leurs soirées.
- Les invitations peuvent être envoyées directement aux utilisateurs connectés ou générer un lien de partage à envoyer directement aux personnes invitées.
- **Seul le créateur de la soirée pourra inviter d'autres utilisateurs ou générer un lien de partage.**

## 3 *Fonctionnalités additionnelles*

Si le temps le permet, voici quelques fonctionnalités supplémentaires qui pourrait être ajoutées

### 3.1 *Liste de contact*

- Les utilisateurs enregistrés pourront rechercher d'autres utilisateurs, voir leur collection de jeu et faire une demande d'ajout à une liste de contact
- Un utilisateur connecté aura accès à un overlay lui permettant de voir l'ensemble des demandes de contact qu'il a en suspens. Il pourra accepter ou refuser les demandes.
- Lorsqu'un utilisateur connecté accède au détail d'un jeu, il verra qui de ses contacts disposent déjà de ce jeu dans sa collection.

### 3.2 *Votation en slidant / swipant*

- À la manière de Tinder, les utilisateurs pourront voter plus facilement aux jeux proposés en proposant une interface visuelle sous forme de carte qu'il sera possible de slider / swiper d'un côté ou l'autre de l'écran pour indiquer si on vote pour ou contre ce jeu.

### 3.3 *Interactions sociales avancées*

- Lors des soirées, les utilisateurs pourront en plus des votes, commenter sous les jeux proposés.

## 4 Exigences techniques

### 4.1 Plateforme et technologie

- L'application doit être une application web accessible depuis n'importe quel navigateur courant.
- Le développement doit se faire en utilisant les dernières technologie web (HTML5, CSS3, JavaScript)
- L'application devra être décomposée en deux : une solution frontend, et une solution backend.
- La solution frontend sera développée au moyen du framework JavaScript Angular
- La solution backend sera développée au moyen de C# (Framework ASP.NET Core).

### 4.2 Données des jeux

- Afin de ne pas devoir gérer une collection gigantesque de données concernant les jeux, il sera préférable de faire appel à une API externe fournissant les données pour la recherche et les informations détaillées.
- Pour pallier une éventuelle panne de l'API, l'application devra stocker en local les jeux enregistrés dans des collections afin de garantir les utilisations pour les soirées. Au final, seul l'outil de recherche devrait être en panne en cas de défaillance de l'API.

### 4.3 Base de données

- L'application doit utiliser une base de données pour stocker les informations concernant les utilisateurs, les soirées, les collections personnelles et les votes.
- Une base de données relationnelles sera utilisée pour stoker ces informations.
- Si une liste de contact est implémentée, elle sera stockée sous forme de base de données graphe.

### 4.4 Sécurité

- L'application doit mettre en œuvre des mesures de sécurité pour protéger les informations des utilisateurs tel qu'un mot de passe chiffré ainsi que la protection contre les injections SQL.

### 4.5 Hébergement

- L'application doit être mise à disposition sur une solution cloud permettant une scalabilité horizontale élastique afin d'encaisser des trafics variables sur l'application.
- L'application doit être tenue à jour en exploitant au mieux les technologies concernant l'intégration et le déploiement continu (CI / CD)

## 5 Contraintes de développement

### 5.1 Stratégie de tests

- L'application doit être testée rigoureusement pour assurer sa stabilité.
- Les bugs connus devront être listés et mis à jour régulièrement. Les bugs majeurs (empêchant une utilisation normale de l'application) doivent être résolus avant de déployer en production.

### 5.2 Intégration et déploiement continu

- Puisque ces technologies sont à inclure dans le processus de développement, les nouvelles fonctionnalités à tester devront être incluses dans des branches différentes du développement afin d'éviter un déploiement non testé (et donc instable) sur l'environnement de production.

## 6 Livrables

- L'application web complète répondant à l'ensemble des fonctionnalités principales mentionnées ci-dessus.
- Une documentation suffisamment détaillée sur l'utilisation de l'application. L'UX devrait être développé au maximum afin de limiter le niveau de détail nécessaire à inscrire dans la documentation.
- Le code source de l'application.