

## **Cahier des Charges :**

### *Contexte général*

L'association Betton Ludique cherche à mettre en place une ludothèque virtuelle et en ligne afin de faciliter l'échange de jeux entre ses adhérents.

Votre mission dans le cadre de ce PPE consiste à proposer une telle application, évolutive. Vous devez implémenter une solution de bout en bout. Cette implémentation implique une réflexion sur les outils déjà disponibles dans la communauté, les patrons de conceptions adaptés et se termine par une étude des solutions éventuelles d'hébergement.

Cette application doit être pensée pour intégrer facilement d'autres produits d'autres associations (musique, livres, matériel de jardinage...).

### *Livrables :*

Comme pour n'importe quel projet, vous aurez un certain nombre de livrables à produire :

- Un diagramme de Gantt permettant de planifier votre travail.
- Un dossier de *specs* , qui contient une description fonctionnelle de votre application. Y figureront en particulier un diagramme de classes UML et la description de votre base de données. Ce dossier sera rédigé pour une équipe de développement.
- Un manuel d'utilisateur qui est rédigé pour l'association, en distinguant les différents rôles prévus dans vos diagrammes de cas d'utilisation UML.
- Le code de votre application et un manuel d'installation qui est rédigé pour l'association.

### *Étapes de votre projet*

Vous devrez prioriser les fonctionnalités que vous mettrez en oeuvre parmi celles qui sont proposées et celles que vous souhaitez implémenter. Vous découperez votre travail en trois parties :

- code livré pour mi-mai : le noyau de votre code qui contient un certain nombre de fonctionnalités basiques liées à votre application permettant en particulier de créer des compte utilisateurs et d'ajouter/supprimer/modifier/lire une partie de la base de données.
- code livré pour mi-juin : une version finale de votre code qui contiendra une bonne partie des fonctionnalités importantes de votre application.
- une version future de votre code proposant des fonctionnalités qui peuvent être intégrées à votre application avec un coût raisonnable mais qui n'auront pas été implémentées.

### *Fonctionnalités possibles :*

- fiche de jeux (ajout, modification, suppression)
  - déclaration de ses propres jeux (ajout d'une fiche, d'un nombre d'exemplaires, de la disponibilité au prêt...)
  - gestion des comptes personnels (création, suppression, modifications, droits...)
  - gestion des emprunts (demande, validation, clôture, évaluation)
  - disponibilités (recherche de jeux, recherche de propriétaire)
  - description du contenu des boîtes (fiche descriptive d'un jeu à l'état neuf, contenu effectif d'un exemplaire)
  - recherche selon la mécanique du jeu, le nombre de joueurs, la durée, le thème, l'âge minimal, le nombre d'exemplaire, la popularité, les derniers jeux...
  - récupération de liens sur le net pour générer facilement de nouvelles fiches de jeux
- Outre les fonctionnalités présentées ci-dessus, vous devez également réfléchir à des fonctionnalités qui seront propres à votre outil de ludothèque collaborative.

### *Technologies à utiliser*

Vous utiliserez l'IDE de votre choix, par exemple Eclipse ou Netbeans.

Il vous est proposé d'utiliser php-MySQL, en mutualisant *a priori* votre travail *via* github.

L'utilisation de symfony serait un plus.

Il est recommandé d'adopter une charte graphique pour l'ensemble de votre application.

Il est recommandé de fournir le script de création et de peuplement des bases de données avec quelques données permettant des tests.

Il est recommandé de respecter les patrons de conception MVC et DAO.

*Liens :*

[Les bonnes pratique en programmation php, y compris pour la BD \(lien github\)](#)

[Cours sur PDO, outils de connexion à MySQL par Php \(developpez.com\)](#)

[Introduction sur les sessions \(apprendre-php\)](#)

[Gestion des sessions des utilisateurs, bonnes pratiques \(openclassrooms\)](#)

[Des patrons de conception en php, dont singleton \(openclassrooms\)](#)