

Gestion d'un système de dépôt de vente

Projet - AWI

Christophe Fiorio

octobre 2024

1 Résumé du projet

Le projet consiste à développer un système de *gestion de dépôt-vente* de jeux de société, à utiliser lors d'événements tels que des festivals de jeux. Le but principal est de permettre aux vendeurs de déposer leurs jeux et de les vendre à des acheteurs potentiels, tout en gérant les transactions financières et les stocks de manière efficace.

Vous devez concevoir une application web complète comprenant à la fois un front-end et un back-end. Cette application devra permettre plusieurs actions clés : le *dépôt* de jeux par les vendeurs, la *vente* de ces jeux aux clients, la *gestion des stocks* restants, ainsi que la *gestion des bilans financiers pour chaque vendeur*. Le système devra également être *robuste* et capable de gérer un afflux important de transactions simultanées, notamment lors de périodes de forte affluence. L'*ergonomie* et l'*efficacité* seront donc des critères importants.

Le projet est conçu pour évaluer votre la *capacité à analyser* un besoin client, à concevoir une *architecture logicielle évolutive* et à développer une application capable de fonctionner dans un environnement réel. L'application devra être *facilement déployable* sur des plateformes cloud, et les étudiants seront responsables de fournir une documentation complète expliquant le processus de déploiement et les choix technologiques effectués. La maintenance devra se faire en environnement déployé. Pour rappel, vous avez accès aux serveurs du département et vous pouvez interroger Lucas Cimini à ce sujet.

Le projet inclut également des éléments d'évolutivité et d'innovation, vous êtes encouragés à proposer des fonctionnalités supplémentaires ou à améliorer celles qui sont demandées.

2 Fonctionnalités attendues

2.1 Dépôt d'un jeu

Le site doit permettre aux *vendeurs* de déposer des jeux qui seront mis en vente. Chaque dépôt engendrera des *frais* calculés par jeu déposé, que le vendeur devra régler. Il doit être possible de déposer plusieurs jeux à la fois, avec un paiement global pour tous les frais. Une remise peut être appliquée aux frais de dépôt.

Un *identifiant unique* doit être généré pour chaque jeu sous forme d'étiquette, afin de *suivre son dépôt et son propriétaire*.

2.2 Vente d'un jeu

Lorsque des jeux sont vendus, il faut :

Enregistrer la transaction , y compris le ou les jeux vendus.

Encaisser la somme et l'intégrer dans le bilan comptable.

Mettre à jour le stock des jeux vendus et créditer le vendeur du montant de la vente (moins les commissions).

Pouvoir consulter les informations financières à tout moment concernant les ventes et de vérifier le montant dû aux vendeurs.

2.3 Gestion des stocks

Les gestionnaires doivent pouvoir :

- Consulter les jeux encore en vente ou déjà vendus.
- Consulter, pour chaque vendeur, la liste des jeux encore en stock et vendus, avec les informations financières associées.

2.4 Transactions et bilans financiers

Les bilans financiers doivent inclure :

Un bilan général : Trésorerie totale, sommes dues aux vendeurs, frais de dépôt encaissés, commissions prélevées, etc.

Un bilan particulier : pour chaque vendeur, avec ses gains, ses sommes dues, et les jeux déposés ou vendus.

2.5 Les vendeurs

Un vendeur doit être enregistré dans le système avec a minima un *nom*, un *email* et un *téléphone*. À son compte doit être associé son stock, ses ventes et l'argent gagné. À tout moment, il faut pouvoir indiquer à un vendeur quels jeux ont été vendus, quelle somme lui est due, et il doit pouvoir **récupérer ses gains ou ses jeux invendus**.

2.6 Les acheteurs et les ventes

2.6.1 Enregistrement des acheteurs

À leur demande, les acheteurs doivent pouvoir demander une facture. **Dans ce cas, l'acheteur est enregistré dans le système avec ses coordonnées (*nom*, *email*, *téléphone*, et éventuellement *adresse*).**

2.6.2 Consultation des ventes

Toutes les ventes doivent être enregistrées et consultables avec les détails sur les jeux vendus et les sommes encaissées. **Si un acheteur demande une facture, il doit être possible de la générer.**

2.7 Les jeux déposés

Chaque jeu déposé doit avoir un *nom* et un *éditeur*¹. Il doit être possible de retrouver facilement un jeu en fonction de son nom et/ou de son éditeur. Une base de données des jeux existants pourra être constituée pour faciliter la saisie d'un dépôt ou d'une vente, des données au format **csv** pourront être fournies à la demande. Toutes les opérations **CRUD** (Create, Read, Update, Delete) doivent être disponibles pour la gestion des jeux.

3 Exemple d'interaction : Processus de vente d'un jeu

1. **Dépôt :** Un vendeur dépose plusieurs jeux via l'interface de dépôt. Pour chaque jeu, une étiquette avec un identifiant unique est générée. Les frais de dépôt sont calculés et payés.
2. **Affichage des jeux :** Les jeux déposés sont **immédiatement visibles** par les acheteurs via l'interface de vente. Les acheteurs peuvent filtrer les jeux en fonction du nom, de l'éditeur ou du prix.
3. **Achat :** Un acheteur sélectionne un jeu et procède au paiement. Le système enregistre la transaction, met à jour le stock du vendeur et crédite son compte, après déduction de la commission.
4. **Suivi des ventes :** Le vendeur peut à tout moment consulter l'état de ses jeux (vendus ou non) et les sommes qui lui sont dues.

1. Un nom de jeu n'étant pas unique, on préfère avoir en plus le nom de l'éditeur pour mieux l'identifier

5. **Récupération des invendus** : À la fin de l'événement, le vendeur peut récupérer ses jeux invendus via une **interface qui affiche ses stocks restants**.

4 Rôles

4.1 Administrateur

L'administrateur du site pourra :

- Gérer les comptes utilisateurs et les rôles.
- Définir les **frais de dépôt** (fixes ou en pourcentage) ainsi que les **commissions**.
- Créer une **session de dépôt-vente**, qui correspond à une période de vente définie (seule une session active à la fois). Les dépôts peuvent se poursuivre pendant une session de vente, mais **aucun dépôt ou vente n'est possible en dehors d'une session**.

5 Autres demandes

1. La **sécurité** et la **robustesse** du site sont primordiales. Des **tests** (unitaires et d'intégration) pourront être réalisés pour garantir la qualité du produit fini.
2. **Statistiques** : Le système devra permettre de consulter des statistiques sur les sessions de dépôt-vente en cours et passées. Toute initiative ou fonctionnalité supplémentaire sera appréciée et pourra être notée en **bonus** (ex. suivi des statistiques détaillées, tableau de bord, etc.).
3. **Gestion des erreurs** : Le système doit gérer les cas d'erreurs tels que la tentative de dépôt de jeux en dehors d'une session active, des erreurs de paiement, ou des jeux en stock non disponibles à la vente.
4. **Initiatives et évolutions** : Les étudiants sont encouragés à proposer des améliorations ou des fonctionnalités supplémentaires, telles que :
 - Un tableau de bord statistique pour les administrateurs.
 - Une interface permettant aux vendeurs de suivre leurs stocks à distance en temps réel.
 - **Intégration de notifications par e-mail après une vente.**

6 Livrables intermédiaires

Afin de vous aider à structurer votre projet et de permettre une progression fluide, deux livrables intermédiaires seront à fournir avant la soutenance finale. Ces livrables vous permettront d'obtenir des retours réguliers et d'ajuster votre travail en fonction des suggestions reçues. Le calendrier exact des rendus sera précisé sous peu sur Moodle.

6.1 Livrable intermédiaire 1 : Plan d'architecture et conception initiale

Date de remise : environ 2 semaines après le lancement du projet.

Ce livrable vise à valider l'architecture technique et l'ergonomie de votre projet avant d'entamer le développement détaillé.

Contenu attendu :

- Un *plan d'architecture initial* comprenant :
 - Un diagramme d'architecture globale.
 - Le schéma de base de données préliminaire.
 - Une justification des choix technologiques.
- Des *wireframes* ou maquettes des principales interfaces utilisateur.
- Une définition préliminaire des *endpoints API*.

Objectif : Valider la structure de votre projet, détecter des erreurs potentielles de conception, et obtenir des retours sur l'ergonomie avant de commencer l'implémentation complète.

6.2 Livrable intermédiaire 2 : Prototype fonctionnel et plan de tests

Date de remise : environ 4 semaines après le début du projet.

Ce livrable servira à démontrer une première version fonctionnelle de certaines parties du projet ainsi qu'à valider votre plan de tests.

Contenu attendu :

- Un *prototype fonctionnel* incluant :
 - Une interface utilisateur basique avec navigation et certaines fonctionnalités opérationnelles (par exemple, le dépôt des jeux).
 - Quelques *endpoints API fonctionnels*, même avec des données simulées.
- Un *plan de tests préliminaire* définissant les types de tests à implémenter et les outils choisis, ainsi que quelques premiers tests unitaires.

Objectif : Vérifier la cohérence de l'implémentation et des APIs, valider l'ergonomie des interfaces, et s'assurer que le projet est sur la bonne voie.

7 Procédure de déploiement

Les étudiants devront configurer le déploiement sur des plateformes cloud comme **Heroku** ou **Vercel**. Ils sont également encouragés à utiliser **Docker** pour standardiser les environnements de déploiement. La gestion des **variables d'environnement** doit être correctement documentée et sécurisée.

8 Points d'évaluation clés

- **Analyse des besoins et clarification** : Les étudiants devront poser des questions pertinentes pour clarifier les fonctionnalités.
- **Conception architecturale** : Évolutivité, modularité, et séparation claire entre front-end et back-end.
- **Déploiement** : Qualité de la procédure et fluidité du déploiement.
- **Tests** : Mise en place de tests pour garantir la robustesse de l'application.

9 Points d'évaluation clés

Analyse des besoins et clarification : Les étudiants devront poser des questions pertinentes pour clarifier les fonctionnalités.

Conception architecturale : Évolutivité, modularité, et séparation claire entre front-end et back-end. Ce point sera évalué dès le livrable intermédiaire 1.

Prototype fonctionnel : Présentation d'une première version fonctionnelle avec quelques fonctionnalités implémentées lors du livrable intermédiaire 2.

Déploiement : Qualité de la procédure et fluidité du déploiement.

Tests : Si possible, mise en place de tests pour garantir la robustesse de l'application.