Projet POEC JAVA

m2iformation.fr

Cahier des charges détaillé pour le projet de groupe - Application de gestion de tournois de jeux vidéo avec Spring MVC

1. Introduction

L'application "Tournois Gamers" est une plateforme web dédiée à l'organisation et à la gestion de tournois de jeux vidéo. Elle est conçue pour permettre aux joueurs de participer à des compétitions, aux administrateurs de gérer les événements et de suivre les performances des participants. Ce projet, réalisé dans le cadre d'une POEC Java, vise à mettre en pratique les compétences des participants en développement web avec Java et Spring MVC.

2. Objectifs du projet

- **Objectif principal** : Développer une application robuste et évolutive permettant de gérer efficacement des tournois de jeux vidéo.
- **Objectifs secondaires** :
  - Fournir une interface utilisateur intuitive et agréable.
  - Assurer une sécurité des données et une gestion fiable des utilisateurs.
  - Implémenter une architecture en trois couches pour une maintenance et une évolutivité facilitées.

3. Contexte et justification

Avec l'essor des compétitions de jeux vidéo et des communautés de gamers, il est crucial de disposer d'outils performants pour l'organisation de tournois. Cette application répond à ce besoin en offrant une solution centralisée pour la gestion des événements, des joueurs et des résultats, tout en garantissant une expérience utilisateur de qualité.

4. Description de l'application

4.1. Fonctionnalités de la couche de présentation

1. **Interface utilisateur** :
   - Accueil avec les tournois en vedette, les prochains événements et les actualités.
   - Pages de tournois avec détails, calendrier des matchs et inscriptions.
   - Dashboard utilisateur pour suivre ses participations, ses résultats, et gérer son profil.
2. **Gestion des utilisateurs** :
   - Formulaires d'inscription et de connexion.
   - Gestion des profils (informations personnelles, avatar, préférences).
   - Système de récupération de mot de passe.

3. **Interaction et communication** :
    - Notifications pour les rappels de match, les mises à jour de tournoi, etc.
    - Messagerie interne pour les communications entre les participants et les administrateurs.

4.2. Fonctionnalités de la couche de logique métier

1. **Gestion des tournois** :
    - Création de tournois par les administrateurs avec configuration des détails (nom, jeu, format, règles, etc.).
    - Gestion des inscriptions (limite de joueurs, vérification des critères d'éligibilité).
    - Génération automatique des brackets (arbre de tournoi) selon le format choisi (simple ou double élimination, round-robin).
2. **Gestion des matchs et résultats** :
    - Planification des matchs avec date et heure.
    - Saisie et validation des résultats par les administrateurs.
    - Mise à jour automatique des brackets et des classements en fonction des résultats.
3. **Suivi des performances** :
    - Calcul et affichage des statistiques des joueurs (victoires, défaites, ratios).
    - Historique des participations et des performances dans les tournois.

4.3. Fonctionnalités de la couche d'accès aux données

1. **Gestion des données utilisateurs** :
   - Stockage des informations personnelles, des rôles et des préférences.
   - Sécurisation des données sensibles (mots de passe, emails).
2. **Gestion des données de tournois** :
   - Base de données des tournois avec les détails, les participants et les résultats.
   - Historique complet des tournois passés et des statistiques associées.
3. **Logs et monitoring** :
   - Enregistrement des actions clés pour audit et suivi des erreurs.
   - Monitoring de la performance de l'application et des requêtes.

5. Technologies utilisées

- **Langage de programmation** : Java 8 ou supérieur
- **Frameworks** :
  - Spring (core, MVC pour la présentation, Data JPA pour l'accès aux données)
  - Thymeleaf pour les templates de vues
- **Base de données** : MySQL ou PostgreSQL, selon les préférences
- **Outils de build et de gestion de dépendances** : Maven
- **Système de contrôle de version** : Git
- **Sécurité** : Spring Security pour la gestion des authentifications et autorisations

6. Exigences fonctionnelles

1. **Utilisateurs** :
   - Les utilisateurs doivent pouvoir s'inscrire, se connecter, et gérer leur profil.
   - Les administrateurs doivent pouvoir créer et gérer les tournois, y compris la validation des résultats.
2. **Tournois** :
   - L'application doit permettre la création de différents formats de tournoi (simple élimination, double élimination, etc.).

3. **Notifications et communications** :
    - Les utilisateurs doivent recevoir des notifications importantes (début de match, résultats, etc.).
    - Un système de messagerie interne doit être disponible pour faciliter la communication.

7. Exigences non fonctionnelles

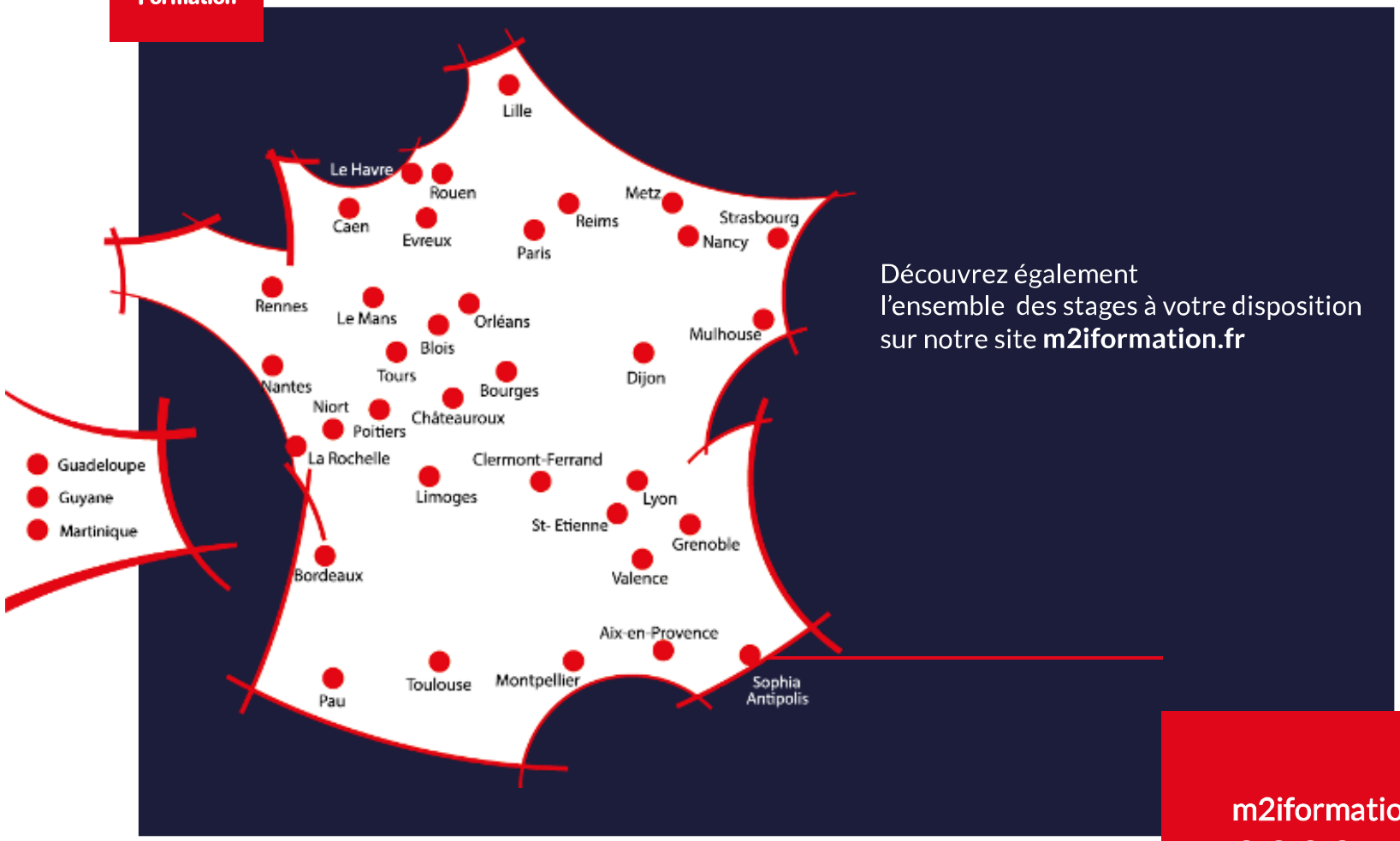
1. **Sécurité** :
    - Chiffrement des données sensibles.
    - Protection contre les attaques courantes (injections SQL, XSS, CSRF).
2. **Accessibilité** :
    - Interface responsive et adaptée aux différents supports (ordinateurs, tablettes, smartphones).
    - Respect des normes d'accessibilité pour les utilisateurs en situation de handicap.

8. Livrables

- **Code source** : Hébergé sur un dépôt Git, documenté et structuré.
- **Documentation** :
  - Technique : architecture, API, configuration du déploiement.
  - Utilisateur : guide de l'utilisateur, FAQ, guide d'administration.
- **Tests** :
  - Tests unitaires et d'intégration pour assurer la qualité du code.
  - Rapport de test et plan de correction des bugs identifiés.

9. Gestion de projet

- **Équipe** :
  - Développeurs Java (3 personnes)



Découvrez également
l'ensemble des stages à votre disposition
sur notre site **m2iformation.fr**

m2iformation.fr