**Rapport d’audit**

**Revue de code**

Equipe auditée (n°3) :

* Romain Bouissou
* Thibault Clément
* Yanis Croguennoc
* Antoine Delagnes
* P.O. : Hervé Leblanc

Equipe auditrice (n°4) :

* Simon Armand
* Titouan Brando
* Christian Gikapa
* Sofia Gribanova
* Alex Pelletier
* P.O. : Pascal Fernandez



**Users Stories :**

La charte de programmation utilisée est la convention Java.

Aucun commentaire / Aucune documentation (seule documentation dans dao.java)

Il y a 10 Users Stories :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User Story | Lecture  Du code | | Bugs détectés | Catégorie des bugs |
| Login | Facile | Liste arbitre n’est pas utilisée | | Mineur  (Code mort) |
| Gestion tournoi | Facile | Possibilité de lancer un tournoi avec seulement une équipe (gérer un minimum d’équipe) | | Critique |
| Ajouter un tournoi | Compliqué | Appel au dao dans la vue ajouterTournoi | | Mineur |
| Aucune gestion des dates passées | | Majeur |
| Gestion équipe | Facile | Aucun bug détecté | |  |
| Ajouter équipe | Facile | Aucun bug détecté | |  |
| Gestion arbitre | Facile | Il reste un Todo dans gestion arbitre et il manque donc la méthode pour supprimer les arbitres d’un tournoi | | Majeur |
| Ajout arbitre | Facile | L’erreur pour l’ajout de 2 fois le même arbitre n’est pas personnalisé | | Mineur |
| Palmarès | Incompréhensible | Le code de la fonction OnDataChanged n’est pas compréhensible | | Mineur |
| Historique | Facile | Appel au dao dans la vue ajouterTournoi | | Mineur |
| Gestion matche | Compliqué | Chaque équipe a un bouton appelé boutonLose sur lequel il faut cliquer pour la déclarer vainqueur | | Mineur |
| Triage de listes d’équipe avec un stream au lieu de faire une requête SQL avec un ‘order by’ | | Mineur |
| Mauvaise vérification du cas où les 4 premières équipes sont exæquo | | Critique |

**Code Smell :**

|  |  |
| --- | --- |
| Problèmes | Type |
| Architecture DAO :  Ne contient que des getAll(), les instances sont gérées avec des streams hors des DAO | Refactoring |
| Architecture DAO :  Une seule DAO générale avec toutes les requêtes possibles, les paramètres changent selon la table appelée.  Permet trop de liberté, donc trop de chance de se tromper ou de casser la BD | Refactoring |
| Responsabilité DAO dans contrôleur :  Suppression de chaque attribut directement dans le contrôleur au lieu de faire une suppression en cascade dans la DAO | Refactoring |
| Classe session :  Devrait être un singleton, car n’appelle que des méthodes static | Refactoring |
| Valeurs en dur :  Année en dur dans le code, en particulier dans Palmarès | Refactoring |
| Interfaces fonctionnelles DAO :  Un seul fichier contient plusieurs interfaces et/ou énumérations | Refactoring |
| Main :  Le main aurait besoin d’un refactoring complet (gère l’initialisation de la base de données, code illisible) | Refactoring |
| Attributs classe métier :  Les liens entre les classes métiers ne sont pas à double sens (exemple : équipe contient joueurs mais joueur ne contient pas d’équipe) | Fonctionnalité manquante |
| EsporterJPanel :  Cette classe implémente Navigable, mais ne fait que stocker un navigable sans changer une seule méthode | Mauvaise utilisation de l’implémentation |
| Import never used (Toute la vue) | Code mort |
| Prepare Statement :  Pas de prepare statement, les requêtes sont des concaténations de chaîne de caractère  Permet des injections sql | Danger |
| Constructeurs singleton :  Les constructeurs de certains singletons ne sont pas en privés et sont utilisés dans d’autre classe (par exemple : ControleurLogin)  Permet deux connexions simultanées à la base de données | Danger |