

Gestion d'un club d'escalade

1 Introduction

Le but de cette page est de présenter le projet qui va servir de base à votre évaluation dans cette unité d'enseignement. Ce projet est à rendre en groupe de **deux étudiant(e)s** (les monômes sont exceptionnellement acceptés).

2 Le club d'escalade : Cahier des charges

L'objectif du mini projet est simple : gérer à l'aide de la technologie JEE un catalogue des sorties proposées par un club d'escalade. Plus précisément,

▶▶ Lot 1 : Les données métier.

- Un membre du club est défini par un identifiant, un nom, un prénom, une adresse électronique et un mot de passe.
- Une sortie est représentée par un ensemble d'informations : identifiant, nom, description, site WEB, date de la sortie et le membre créateur de cette sortie.
- Chaque sortie est associée à une catégorie (par exemple, Alpinisme de roche, Alpinisme de neige, Alpinisme mixte, Escalade sportive, Randonnée du vertige, Terrain d'aventure). Une catégorie a donc un nom et un identifiant.

▶▶ Lot 2 : Initialisation.

- Initialement, l'application va créer un jeu de données de test (membres, catégories et sorties).

▶▶ Lot 3 : Visiteur.

- Un visiteur (non authentifié) doit avoir accès à une liste des catégories, une liste de sorties de chaque catégorie et une vue détaillée de chaque sortie (sauf site WEB et créateur). Une fonction de recherche doit être offerte.
- L'application doit être fonctionnelle si nous avons plusieurs milliers de sorties, plusieurs dizaines de catégories et plusieurs centaines de membres (**il faut le montrer**).

▶▶ Lot 4 :

- Chaque membre doit pouvoir s'authentifier.
- Les membres authentifiés peuvent faire la même chose que les visiteurs mais ils ont accès à toutes les informations (y compris les sites WEB et les créateurs et les sorties proposées par chaque membre).
- Chaque membre authentifié peut créer, détruire et modifier **ses** sorties.

▶▶ Lot 5 :

- Il faut prévoir un mécanisme de récupération du mot de passe.
- Si vous avez besoin d'envoyer un mail pour le mécanisme de récupération d'un mot de passe, je vous conseille :
 - ▷ d'utiliser Spring Mail pour envoyer le mail,
 - ▷ d'utiliser le serveur Java Fake SMTP (aussi disponible ici) pour lancer un faux serveur SMTP qui vous permet facilement de suivre les messages envoyés,
 - ▷ de configurer Spring Mail et Fake SMTP pour utiliser la machine locale (`localhost`) et le port 10025.

3 Étape 1 : Stockage (DAO/Spring/JPA) : Lots 1 et 2

▶▶ Objectifs :

- Conception et création des *JavaBeans* permettant de représenter les données de l'application.
- Utilisation de la technologie Spring-data/JPA pour gérer la persistance des *JavaBeans*.
- Les fonctions attendues de la DAO pourraient être :
 - ▷ Obtenir la liste des catégories,
 - ▷ Obtenir une catégorie avec ses sorties,
 - ▷ Créer une catégorie,
 - ▷ Obtenir les détails d'une sortie,
 - ▷ Chercher des sorties en fonction de critères,
 - ▷ Obtenir la liste des membres,
 - ▷ Obtenir un membre particulier,
 - ▷ Obtenir un membre particulier avec ses sorties,
 - ▷ Créer, modifier et supprimer une sortie,
 - ▷ Créer et modifier un membre

i Architecture : Vous devez mettre en place une couche de service d'accès aux données (souvent appelée DAO pour *Data Access Object*), Cette couche est constituée d'une interface (**indépendante du choix de JPA**) et d'une implantation (liée à JPA).

▶▶ Contrainte à respecter :

- Vous devez **absolument** prévoir une classe de test unitaire *Junit* afin de valider les méthodes offertes par votre implantation.
- Vos composants (DAO, DataSource, tests JUnit) doivent **absolument** être connectés et paramétrés par une couche Spring-boot.

⚠ À rendre sur AMETICE (archive ZIP) pour le lundi 31 mars 2025 23h55.

4 Étape 2 : Métier (Spring)

Cette étape consiste à mettre en place une couche métier basée sur Spring. Cette couche devra prendre en charge l'accès et la modification des données métier. Elle assure

- l'authentification des membres (attendre de découvrir les fonctionnalités de Spring-Security,
- la modification des données,
- l'accès aux données.

5 Étape 3 : Présentation (JSP/SpringMVC)

Cette étape consiste à mettre en place une application WEB basée sur la technologie **Spring MVC/JSP** qui respecte les contraintes suivantes :

- Votre application WEB sera construite autour du framework **Spring MVC**.
- Les réponses seront construites par des pages JSP. Dans la mesure du possible utilisez les balises de contrôle de la JSTL (boucles, tests et affichage).
- Les aspects de sécurité seront traités par **Spring-Security**.
- Faites en sorte que votre application produise des pages HTML 5 valides accompagnées d'un framework CSS.
- **Votre application WEB doit vérifier toutes les requêtes utilisateur avant de les exécuter.**

6 Échéances

Ce projet est à rendre, par équipe de **deux personnes** (ou **une personne** exceptionnellement), pour le **23 avril 2025 à 23h55**.

- ▶▶ **Documents attendus :** une archive ZIP (de la forme `nom1-nom2.zip`) qui regroupe
- un cahier des charges (fichier `CdC.pdf` une ou deux pages),
 - un rapport (fichier `rapport.pdf` de moins de 10 pages),
 - le projet de l'application WEB (répertoire `projet`) au format maven (avec un nettoyage `mvn clean`),
 - votre application packagée sur la forme d'un fichier WAR basé sur **SpringBoot** (fichier `club.war`),

- ⚠ **Points importants :**
- **Point 1 :** Votre application doit utiliser une base de données **embarquée et stockée en mémoire** (voir exemple sur le TP Spring Boot) afin qu'elle soit facile à tester.
 - **Point 2 :** Dès la **deuxième page**, le rapport doit donner clairement des informations d'authentification afin que je puisse utiliser votre application.
 - **Point 3 :** La forme du rendu doit **ABSOLUMENT être respectée**. Les travaux sont à rendre sur la plateforme AMETICE.
 - **Point 4 :** Choisissez un créneau en modifiant en ligne le fichier du planning. Ne laissez pas de créneau vide avant le votre et ne modifiez pas les créneaux déjà choisis.