

# **Systemes d'Information – projet 2023/24**

## **Master 1<sup>ère</sup> année TIIL-A & ILIADE – UBO**

Ce projet, à réaliser par groupe de 3 étudiants, compte pour la note de contrôle continu de l'UE Système d'Information ( $\frac{1}{3}$  de la note de l'UE).

Il s'agit d'implémenter une application Web de gestion d'événements d'une association en définissant plusieurs services implémentés par différentes technologies (Servlet, Spring) et s'appuyant sur plusieurs serveurs de bases de données (SQL, MongoDB).

### **Description de l'application**

L'application à réaliser gère les événements d'une association et de ses membres.

Membre : un membre d'association est caractérisé par son nom, son prénom, son âge et son adresse. Il n'existe pas deux membres ayant le même nom et le même prénom. Un membre se connecte à l'application en précisant un mot de passe.

Événement : un événement est défini par un nom, une date et une heure, une durée, un lieu et un nombre maximum de personnes y participant. Deux événements ne peuvent pas avoir lieu en même temps dans le même lieu. Le lieu choisi pour l'événement doit avoir une capacité suffisante pour accueillir tous les participants. Les membres ont la possibilité de déposer des commentaires sur un événement (message textuel).

Lieu : un lieu est défini par un nom, par une adresse et par une capacité d'accueil (nombre maximum de personnes dans la salle).

Inscription : un membre peut s'inscrire à un événement. La contrainte est que le membre ne doit pas être déjà inscrit à un événement qui se chevauche dans le temps avec le nouvel événement auquel il veut s'inscrire. On ne peut également s'inscrire à un événement que s'il n'a pas déjà eu lieu et que le nombre maximal de participants n'est pas atteint. Un membre peut se désinscrire d'un événement s'il n'a pas encore eu lieu.

Fonctionnalités attendues : on doit pouvoir visualiser l'ensemble des membres de l'association, l'ensemble des événements (tous ou ceux à venir), l'ensemble des inscriptions pour un événement donné (avec le nombre d'inscrits) et pour chaque membre, pouvoir lister les événements auxquels il est inscrit (tous ou ceux à venir). L'application doit permettre de créer, modifier et supprimer de nouveaux membres ou événements (par soucis de simplification, n'importe quel membre connecté peut créer un événement). On doit également pouvoir afficher les événements d'un lieu (tous ou ceux à venir) avec la carte d'accès au lieu.

## **Contraintes techniques**

La base de données stockant les informations sur les événements, lieux et membres sera une base de données SQL. L'accès aux données de cette base se fera via des entités JPA. Les commentaires seront stockés dans une base MongoDB.

L'interface client Web sera développée en utilisant le framework Vue.js. Elle permet d'accéder et de modifier les informations sur les membres et événements. Pour un lieu, en complément de l'adresse, une carte localisant le lieu sera intégrée dans l'interface via un appel à l'API d'OpenStreetMap.

Plusieurs API REST seront développées, elles seront définies pendant les TP.

Vous devrez implémenter deux API via Spring et deux autres API plus basiquement avec une Servlet. Le choix des API à faire en Spring ou en Servlet est le vôtre, vous faites ce qui vous semble le mieux. S'il y a plus de 4 API dans l'application, vous implémentez les autres API avec la technologie que vous souhaitez.

Pour partager le code au sein de votre trinôme, vous utiliserez Git.

## **Evaluation**

### **Livrable 1 et évaluation sur machine**

Dates d'évaluation des projets en fonction du groupe de TP (voir emploi du temp ADE) :

- TP1 : Mercredi 6 Mars à 13h30
- TP2 : Jeudi 7 Mars à 13h30
- TP3 : Vendredi 8 Mars à 13h30

Pour le Vendredi 8 Mars à 19h au plus tard, vous déposerez sur le Moodle de l'UE, le code complet de votre application (gestion des données, code serveur et client HTML/JS).

Vous joindrez dans l'archive, un fichier PDF dans lequel vous détaillerez votre schéma de BDD, la structure des POJO ainsi que la liste des méthodes d'accès aux données. Pour rappel, l'idée est qu'il y a une couche données qui sera utilisée par les Servlet et sera bien découplée de l'implémentation des API de la partie métier Web. Vous détaillerez également dans le document les API REST que vous avez définies en précisant pour chaque API avec quelle technologie elle a été implémentée.

### **Livrable 2**

Partie Spring : A DÉFINIR