

Créez un site communautaire autour de l'escalade

Mis à jour le 09/05/2018

- 150 heures

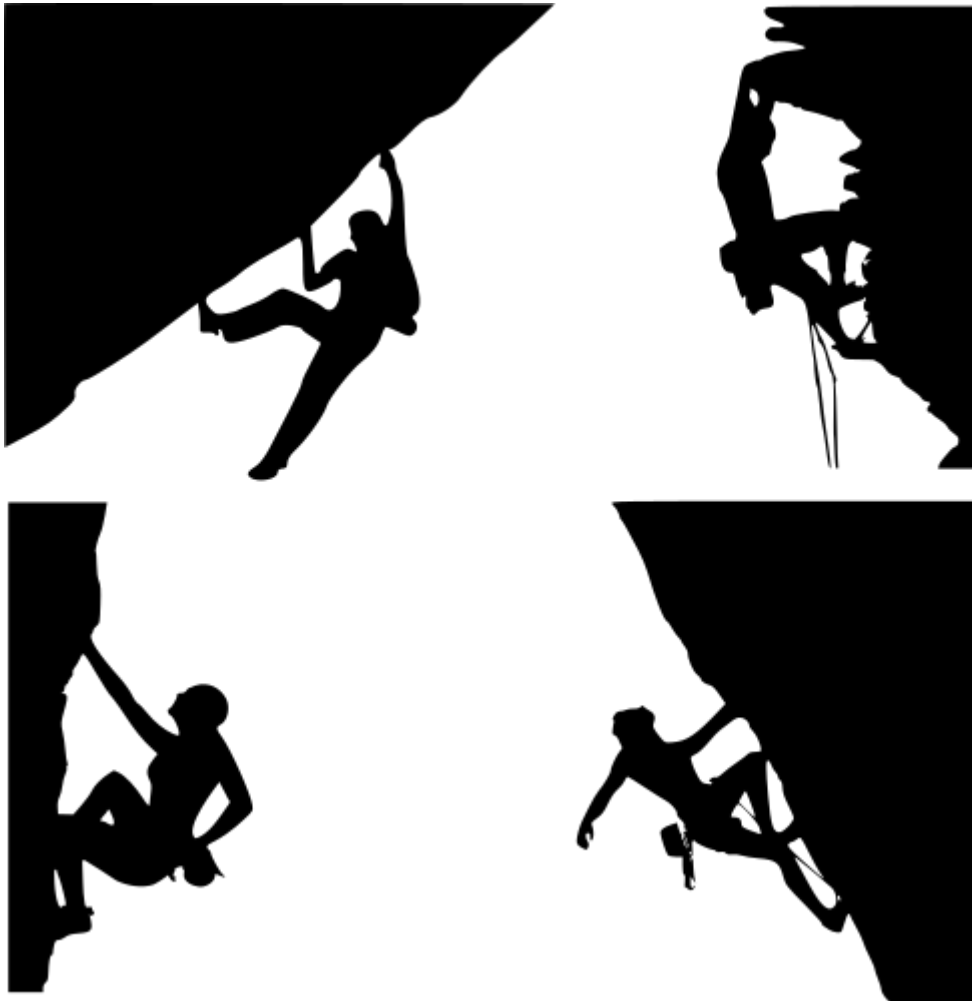
Énoncé

Ce projet doit être réalisé avec le suivi de votre mentor. Prenez contact avec lui pour en savoir plus.

Pré-requis

- Être capable de coder une petite application simple en Java

Contexte



Votre ambition :
créer un site communautaire autour de l'escalade

Vous faites partie d'un club d'escalade et comme tout bon grimpeur vous passez des heures à discuter avec les autres grimpeurs, des voies sur les sites de la région, des lieux insolites pour grimper...

Et puis un jour, vous vous dites : « Ce serait sympa de regrouper un peu toutes ces informations, et de les partager avec le plus grand nombre ! ». Ça tombe bien, vous êtes en pleine formation de développeur d'application. Vous faites un petit sondage autour de vous et vous vous lancez dans la création d'une application web permettant :

- de partager des informations sur les sites, leurs secteurs et les voies de chaque secteur (hauteur, cotation, nombre de points...)
- de faire une recherche multi-critères pour trouver votre prochain site de grimpe
- de laisser des commentaires
- de présenter les topo qui existent et les sites/secteurs qu'ils couvrent
- d'avoir un espace de prêt de topo (les propriétaires de topo peuvent proposer le prêt de leur topo et les gens intéressés peuvent voir les topo disponibles et les réserver pour une période)

Vocabulaire

Pour les non-initiés, voici quelques éléments du vocabulaire du *grimpeur* :

- Un *site* ou *spot*, c'est un lieu où il est possible de grimper
- Les *sites* peuvent être découpés en plusieurs *secteurs* qui regroupent un ensemble de *voies*
- « Lao Tzeu l'a dit : il faut trouver la *voie* » : c'est le chemin à emprunter par le grimpeur pour arriver à destination : le haut de la voie.
- si la voie est découpée en plusieurs « parties » à grimper les unes après les autres, ces parties s'appellent des *longueurs* et on trouve un *relai* en haut de chaque *longueur*
- Quand le *grimpeur* arrive en haut de la *longueur*, c'est à ce *relai* qu'il se *vache*, c'est à dire qu'il s'y accroche, avec à l'aide de sa *vache* ou *longe* (corde nouée sur le *baudrier* du grimpeur et équipée d'un *mousqueton à verrouillage*)
- Les *points* ou *spits* sont des ancrages fixes que l'on trouve dans les *voies* dites « équipées » et qui permettent au grimpeur de s'assurer au fur et à mesure de sa progression, à l'aide de *dégaines*
- La *cotation* d'une *longueur* ou d'une *voie*, représente sa difficulté. En France, le système de cotation va, en gros, par ordre croissant de difficulté, de 3 à 9c. Le chiffre est en quelque sorte, l'unité principale et la lettre une sous-unité (de « A » à « C »)
- Un *topo* est un recueil contenant toutes les informations utiles sur les sites d'escalade d'une région (les secteurs, les voies, leur hauteur, leur cotation, le nombre de points...). Une bible quoi.

Mais le meilleur des perles du vocabulaire des grimpeurs et sans aucun doute le nom des voies, dont l'honneur du baptême revient à celui qui l'a ouverte. Et autant vous le dire, les ouvriers ne manquent pas d'imagination !!

Aperçu d'un topo

Voici un petit aperçu de ce que l'on peut trouver dans un topo d'escalade :

[Colima'100 - topo provisoire v1](#) , le PDF est aussi [ici](#) (source : Franck Bernard, [site web](#) du Comité Régional FFME Réunion, 9 juillet 2014).

Attention, si vous êtes réellement un grimpeur et que ce site vous tente, il s'agit d'un topo provisoire d'un nouveau spot, ne vous y fiez pas aveuglément. Renseignez-vous auprès de la FFME et respectez les recommandations de sécurité liées à la pratique de l'escalade.

Vous êtes seul responsable des conséquences de l'utilisation de ce topo et de votre pratique en générale !

Travail demandé

Vous réaliserez une application web en Java/JEE (JDK 8) avec les fonctionnalités décrites ci-dessus. L'application sera à déployer sur un serveur *Apache Tomcat 9* et utilisera une base de données *PostgreSQL 9.x*.

L'application web sera packagée (WAR) avec *Apache Maven*.

Le code sera géré avec *Git* et ce, dès le début du projet. Vous aurez donc un historique assez important sur votre dépôt Git.

Votre mentor et le mentor de soutenance pourront consulter cet historique dans le but, non pas de relever vos erreurs, mais plutôt de voir votre démarche de développement et votre utilisation de Git.

N'ayez donc pas peur de faire des commits ! Au contraire, un historique contenant seulement quelques "gros" commits révèle souvent une mauvaise approche de développement et une utilisation peu efficace de Git.

Livrables attendus

Vous livrerez, sur [GitHub](#) ou [GitLab](#) (dans un seul ou deux dépôts Git) :

- le code source de l'application
- les scripts SQL de création de la base de données et d'un jeu de données de démo
- une documentation succincte (un fichier `README.md` suffit) expliquant comment déployer l'application (base de données, configuration sur serveur Tomcat, fichiers de configuration...)
- les fichiers de configuration exemple

Vous donnerez un accès en lecture à votre/vos dépôt(s) Git à votre mentor et au mentor qui vous fera passer la soutenance

Soutenance

Avant la soutenance

Quelques jours avant la date de la soutenance (en général 3-4 jours avant), vous transmettez tous vos livrables au mentor qui vous fera passer celle-ci, afin qu'il puisse en prendre connaissance en amont et éventuellement tester votre application.

Déroulement de la soutenance

Il vous est demandé de vous mettre en situation réelle : en effet, il s'agit d'un **rendez-vous professionnel**.

Vous vous adresserez à un *développeur expérimenté* rencontré dans un salon des métiers de

l'informatique. Son entreprise recrute et il est intéressé par votre profil. Vous avez fixé un rendez-vous avec lui afin de lui présenter une de vos réalisations.

La soutenance, d'une durée de 25-30 minutes, se déroulera en deux parties :

- Partie 1 – 20-25 minutes : **Simulation d'une réunion professionnelle.**
Le mentor jouera le rôle d'un développeur expérimenté
 - [~ 5 minutes] Vous ferez une démonstration rapide de votre application web
 - [~ 5 minutes] Vous lui présenterez votre solution technique et les points particuliers de votre développement (à vous d'être pertinent sur les points à aborder)
 - [~ 10 minutes] Vous échangerez sur votre travail et vos choix techniques
- Partie 2 – 5-10 minutes : **Retour sur la soutenance**
 - Le mentor pourra vous demander d'approfondir certains aspects ou vous questionner sur vos livrables
 - Le mentor vous fera un debrief sur votre prestation en soutenance

Étapes de réalisation proposées

La conception

1. Lister les cas d'utilisation que vous voulez développer
2. Modéliser le domaine fonctionnel (diagramme de classe UML) et le schéma de base de données

Monter l'environnement de développement

1. Créer les scripts de création de la base de données et le jeu de données de démo
2. Initialiser la base de données PostgreSQL

Créer l'application web

1. Configurer l'EDI (Eclipse/IntelliJ/...)
2. Préparer le squelette de l'application web (configuration Maven, Spring IOC, fichiers de configuration...)
3. Configurer le serveur Apache Tomcat dans votre EDI pour pouvoir le lancer en mode debug
4. Implémenter l'application :
 1. Créer la structure de la webapp (configuration, images, CSS...)
 2. Créer et configurer les filters/listeners
 3. Créer les actions
 4. Implémenter les fonctionnalités des actions (et des couches inférieures)
5. Packager l'application web avec Maven

Compétences à valider

- Respecter les bonnes pratiques de développement en vigueur
- Stocker et récupérer les informations dans la base de données en langage SQL
- Sélectionner les langages de programmation adaptés pour le développement de l'application
- Développer une application proposant les fonctionnalités attendues par le client
- Gérer ses données avec une base de données
- Gérer les dépendances et packager une application avec Apache Maven