

Sujet projet génie logiciel

1. Sujet

Vous développerez par groupe de 2 ou 3 une application à minimum 2-tier de type CRUD. C'est-à-dire que l'architecture de votre code sera la suivante :

- Une couche webservice « backend » qui devra répondre à des requêtes http GET/POST/PUT/DELETE qui permettront respectivement de lire, ajouter, modifier, supprimer des données. Attention dans la norme http, les requêtes PUT doivent être idempotente (=si vous faite la même requête 1, 2 ou n fois le résultat devra toujours être le même)
- Une couche donnée avec une base de données. Le choix du moteur est libre mais doit être cohérent avec votre application (SQL vs NoSQL).

Le sujet métier de votre application est libre et n'a pas besoin d'être innovant. Une gestion de bibliothèque conviendra parfaitement.

Il n'y a pas d'attente spécifique sur le langage. Par contre quel que soit le langage choisi, vous devez respecter ses normes et les bonnes pratiques de développement. Par exemple, ce n'est pas parce que python peut fonctionner sans héritage que vous devez vous en passer.

2. Attendu

En plus de cette architecture de base, votre application devra contenir :

- Un design pattern vu en cours. Cela peut être :
 - Une factory pour instancier les bonnes classes d'objet
 - Une strategy pour adapter un traitement à une condition
 - Un decorator pour ajouter du comportement à des objets
 - Un observer pour faire une action quand un objet est mis à jour
 - Une state machine pour enchaîner des traitements dans un certain ordre
- Vous êtes libre du design pattern, et il est acceptable que le design pattern ait l'air artificiel et soit uniquement là pour répondre à cette demande.
- Une base de donnée « complexe ». Si vous utilisez un modèle relationnel à minima 3 tables sans compter les tables d'association. Dans le cas d'un modèle non relationnel il faut que vos données s'y prêtent bien.
 - Des tests. Pour rester cohérent avec le cours, aucune couverture minimale est attendue, mais vos tests doivent être intelligents et utiles. Vous pouvez décider de tout tester en faisant du TDD, ou de tester la partie la plus compliquée. Attention tester les interactions avec votre base va vous demander d'utiliser des outils spécifiques.
 - De quoi lancer votre application en utilisant seulement docker. Donc à minima un dockerfile et des explications pour lancer votre application sont donc attendues. Vous pouvez choisir de proposer également un script bash qui lance tout ou un fichier docker compose pour lancer les multiples conteneurs de votre application.
 - Une explication rapide de votre application dans un fichier readme.md

Tout élément complémentaire :

- Pipeline CI/CD
- Outils d'automatisation
- Plusieurs design pattern

- Déploiement sur une plateforme de cloud computing
- Front dans le techno de votre choix
- Etc

sera valorisé.

3. Rendu

Le rendu se fera le 15 février. Sur Moodle vous aurez (remi.pepin@ensai.fr) avec si besoin la référence du commit à récupérer (branche ou tag).