Haute École Léonard de Vinci
SCIENCES ET TECHNIQUES

Projet d'application d'entreprise

Livrable 2 – Architecture & complément d'analyse

Livrable 2 : Architecture = 7% de l'évaluation

A. Implémentation d'une version qui déploie tous les composants de l'architecture.
   - Y compris les tests unitaires.
B. Réalisation d'un cas d'utilisation.
C. Ecriture des tests fonctionnels.
D. Complément d'analyse.

Délai de remise du rapport : **veille de la revue de code à 20h au plus tard, pour tous.**

Cas d'utilisation à implémenter

- Authentification d’un utilisateur.
  - **Se connecter**, avec un bouton "remember me".
    - Stockage d'un token JWT dans le session/local storage.
    - N'oubliez pas les différents cas d'erreur.
    - N'oubliez pas qu'un utilisateur doit être validé pour se connecter.
  - **Se déconnecter.**
  - **Rafraichissement.**
    - Lorsqu'on est connecté, un F5 sur la page doit garder le fait qu'on soit connecté.
    - Il faut pour cela une route pour récupérer le user connecté (*/auth/user* par ex.), sur base du token stocké dans le session/local storage, transmis dans les headers.
  - **Utilisateur identifié** :
    - Lorsqu'on est connecté, il faut afficher un signe distinctif de l'utilisateur connecté (son nom dans la navbar, un mot de bienvenue...).
- Pour vous simplifier la vie, vous pouvez préenregistrer des utilisateurs directement dans la DB.

Consignes : Implémentation

- Backend :
  - API REST qui sert du JSON, correctement structurée en ressources
  - Technologies vues en Backend-Java (Jersey, HK2...).
  - Architecture 3 couches vue en AAE, qui va évoluer au fil du temps
  - Usage de JWT et de BCrypt.
- Front-end
  - Single-Page-Application.
  - Technologies vues en Javascript.
  - Utilisation d'un framework CSS type "Bootstrap", à valider par les professeurs.
  - Pas d'utilisation d'un framework JS (React, Angular...).
  - Usage de librairies JS autorisée, mais toujours après validation par les professeurs.

Consignes : gestion des dépendances

- Injection de dépendances des objets de service.
- Fichiers *.properties, comme vu au cours d'AJ/AAE :
  - dev.properties => config de dev
  - prod.properties => config de production ?
- Implémentation avec HK2, comme vu en Backend-Java.

Consignes : tests unitaires

- Uniquement la couche business, et plus précisément, les UCC.
- Mocks pour les parties nécessaires. Surtout pas d'usage de la DB dans les tests !
- Mocks "à la main" ou usage de Mockito.
- Injection de dépendances avec HK2.

- Cfr. Document sur Moodle.

Consignes : DB

- L’application doit se connecter au serveur PostgreSQL de l’école.
  - Cfr. Guide de démarrage du projet.
- Paramètres de connexion dans le fichier properties.
- Script de création et de "seed" de la DB "init.sql" versionné dans votre repository.

Consignes : DB

- Logique des données autant que possible au niveau du Java plutôt qu'au niveau du SQL.
- En SQL, vous **pouvez** utiliser :
  - les clefs primaires et les clefs étrangères.
  - les SERIAL (= les séquences).
  - éventuellement les NOT NULL, mais ce n'est pas obligatoire.
- Vous ne **pouvez pas** utiliser :
  - les contraintes CHECK, UNIQUE et DEFAULT.
  - les procédures stoquées et les triggers.
  - les vues.
  - les enums.

Rapport à rédiger

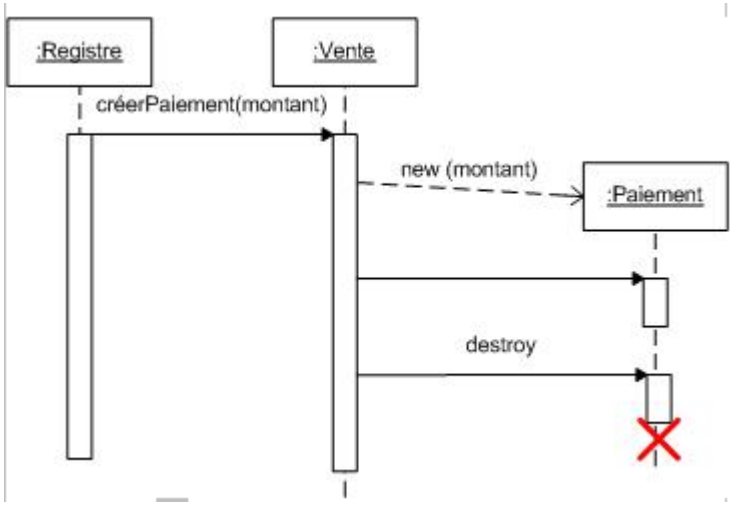
1. Une page de garde contenant au minimum un titre, la date de remise du livrable, votre numéro de groupe, sa composition, **le numéro de commit et de build jenkins**.
   - Attention ! Le commit doit être sur master, et dans le cas contraire, il faut préciser la branche.
   - Version courte du numéro du commit (0f3eccbb).
2. Une table des matières.

Rapport à rédiger

3. Rapport de métriques, avec **justification** :
   - Nombre de violations de PMD, de checkstyle et de CPD
   - Couverture de code des tests unitaires, uniquement pour la couche business.
   - Captures d'écran des listes des violations avec leur nombre par type (LineLengthCheck, MissingJavadocMethodCheck...)
   - Lien direct vers le job (exemple : https://jenkins.vinci.be/job/projet-ae-groupe-01/1112/)
   - Nombre total de merge-requests (gitlab), avec une comparaison avec les livrables précédents (tableau)
4. Si un ou plusieurs points de ce livrable n'est pas rempli, avec **justification**.
5. Spécialisation du diagramme de classes à votre architecture.

Rapport à rédiger

6. Diagramme de séquence du cas d’utilisation "authentification d’un utilisateur" (login), avec tout le flux depuis le front-end jusqu'à la DB.



Rapport à rédiger

7. Complément du livrable 1 (par groupe).
8. Un rapport d'activités : qui dans le groupe a travaillé à quoi et dans quelle proportion. Ce rapport est aussi l'occasion d'expliquer ce qui a bien ou moins bien fonctionné lors de la réalisation de ce livrable. Vous devez inclure les parties pertinentes des graphiques créés par www.coursinfo.vinci.be, ajoutez-y les heures de chacun.

Revue de code

- Parcours de votre code avec un professeur
- Il vous listera les soucis, et les remèdes à apporter

- Fiche d'évaluation sur Moodle
  - Préparez-vous !
  - Corrigez ensuite directement !