

## **Modélisation de processus**

### **Sujet de projet**

L'objectif du projet est de développer un prototype d'application workflow en utilisant Activiti, de le déployer dans le moteur de workflow, de l'exécuter et de le tester à partir de l'interface web (Activiti explorer).

### **Processus de gestion de stages dans une école d'ingénieur**

Le prototype d'application workflow a comme objectif d'automatiser le processus de gestion de stages dans une école d'ingénieurs.

Le processus contient les étapes suivantes :

- soumission de l'offre de stage par l'étudiant (informations comme sujet de stage, nom entreprise, dates, etc),
- approbation du sujet par le responsable pédagogique. Si le sujet n'est pas assez détaillé, le responsable pédagogique demande à l'étudiant d'apporter des précisions.
- vérifications des informations administratives par le service des stages. S'il manque des informations (par exemple, mail du tuteur), le service de stage demande à l'étudiant de modifier sa demande de convention.
- Après la validation pédagogique et administrative, l'étudiant peut imprimer la convention. Il la fait signer par l'entreprise et la dépose aux services de stage (qui enregistre le fait que la convention a été signée).
- l'affectation du tuteur (l'étudiant peut indiquer un tuteur, sinon le stage est proposé à tous les tuteurs)
- Le responsable administratif choisit une date de soutenance et la communique au tuteur et à l'étudiant.
- envoi du rapport de stage. Si le rapport n'est pas envoyé une semaine avant la date de soutenance, des rappels sont envoyés à l'étudiant. Si le rapport n'est pas envoyé 3 jours avant la soutenance, le tuteur est notifié. Si le rapport n'est pas envoyé avant le jour de la soutenance, le responsable pédagogique est notifié.
- Après la soutenance, la note de stage est enregistrée par le tuteur
- toutes les informations importantes liées au stage (entreprise, sujet, mots clés pour le sujet, tuteur dans l'entreprise, contact) seront enregistrées dans une base de données facilitant la recherche de stages pour les étudiants des promotions suivantes et les statistiques.

### **Travail à faire**

1. Définissez le diagramme BPMN 2.0 du processus.
2. Ecrivez le code java nécessaire pour développer une application workflow sur Activiti qui automatise ce processus. Définissez le schéma de la table (ou des tables) utilisée pour stocker les informations. Créez des utilisateurs et des groupes.
3. Déployez le processus sur Activiti-explorer et testez-le.

### **Modalités de rendu**

Le projet sera fait en binôme.

Date limite de rendu : le 25 janvier

Vous envoyez l'archive par mail à [daniela.grigori@gmail.com](mailto:daniela.grigori@gmail.com).

Votre archive devra comporter :

- un document pdf comportant :
  - une description textuelle de ce qui fait le processus métier mis en œuvre (hypothèses, limites),
  - le diagramme BPMN 2.0 réalisé,
  - les résultats obtenus (copies d'écran),
  - en annexe : le code source (java, sql, ..).
- le .zip de votre projet contenant les sources.

Les soutenances de projet seront organisées pour vous permettre de faire une démonstration de votre prototype (la semaine du 28 janvier).

## Documentation

Activiti user guide : <http://www.activiti.org/userguide/>

### Annexe – L'accès à la base de données

- Activiti utilise par défaut la base de donnée « en mémoire » H2 DB. Il est possible d'utiliser MySQL, Oracle, Postgres, en changeant le fichier de configuration.

L'accès à la base de données dans un processus peut se faire en utilisant du code SQL ou JPA (Java Persistence API).

Exemple de code qui devrait être mis dans une classe JavaDelegate :

```
//Creating a connection
Class.forName("org.h2.Driver");
Connection conn = DriverManager.getConnection("jdbc:h2:~/test", "sa",
    "");

Statement stmt = conn.createStatement();

// dropping a table
String dropTableString = "DROP TABLE IF EXISTS BOOK";
stmt.execute(dropTableString);
System.out.println("Dropped table BOOK");

//Creating a table
String createTableString = "" + "CREATE TABLE IF NOT EXISTS BOOK"
    + " (" + "ID INT, " + "TITLE VARCHAR(255), "
    + "SUBTITLE VARCHAR(255), " + "ISBN VARCHAR(255), "
    + "PRIMARY KEY(ID)" + " )";

stmt.execute(createTableString);
System.out.println("Created table BOOK");
```

Insertion et lecture des données :

```
// Inserting a row
PreparedStatement st = conn
    .prepareStatement("INSERT INTO \"BOOK\" values(?,?,?,?)");
st.setInt(1, new Random().nextInt());
st.setString(2, "Activiti in Action");
st.setString(3, "Activiti in Action");
st.setString(4, "0307269752");
st.executeUpdate();

// Reading data
Statement statement = conn.createStatement();
ResultSet resultSet = statement.executeQuery("SELECT * FROM BOOK");
while (resultSet.next()) {
    System.out.println("Book id: " + resultSet.getString(1)
        + " Book Title: " + resultSet.getString("TITLE")
        + " Book Subtitle: " + resultSet.getString("SUBTITLE")
        + " Book ISBN: " + resultSet.getString("ISBN"));
}

conn.close();
```

- La console H2 peut être lancée en utilisant le script de démarrage `h2.console.start.sh` (après avoir installé H2)

(si le script n'ouvre pas une nouvelle page dans le navigateur, allez à l'adresse <http://localhost:8082>). Cliquez sur Connect.

Attention à ne pas supprimer les tables utilisées par Activiti !