

Rapport d'itération n°5

1. Documentation utilisateur

1.1. Mode opératoire

- **Nom du livrable** : Projet JAVA modération
- **Version** : 0.5
- **Date de livraison** : 31/03/2025
- **Description** : Les propriétaires du site r3st0.fr souhaitent se doter d'une application Java qui sera installée sur les postes de travail des modérateurs, permettant de contrôler les avis déposés et de supprimer ou de bloquer la publication des avis ne respectant pas la charte d'utilisation du site ; chaque modérateur devra s'authentifier pour pouvoir utiliser l'application.

Installation :

- Pré-requis :
 - java <= 18
 - un IDE (netBeans, vs code)
 - wamp/xamp avec la BDD importée
- Procédure d'installation :
 1. Récupérer le projet depuis le dépôt gitLab (git clone <https://gitlab.com/APeixoto/modo-modo.git>)
 2. Importer la BDD fournit avec le projet dans wamp/xamp
 3. Le projet est prêt à être exécuté

Choix de la technologie d'accès aux données :

Nous avons choisi d'utiliser une **couche DAO** dans notre projet pour plusieurs raisons :

- **Simplicité et contrôle** : Le DAO nous permet de gérer directement les requêtes SQL et d'avoir un contrôle total sur l'accès aux données, sans complexité supplémentaire.
- **Architecture adaptée** : L'application étant locale et monolithique, une **API REST** serait inutilement lourde car il n'y a pas de communication entre services ou clients distants.
- **Alternative à JPA** : JPA apporte de la facilité pour des projets complexes, mais il impose un certain cadre et génère des surcharges inutiles pour un projet simple. Le DAO reste plus léger et adapté ici.

- **Bonne pratique** : Le DAO respecte la séparation des responsabilités et facilite la maintenance du code.

2. Documentation technique

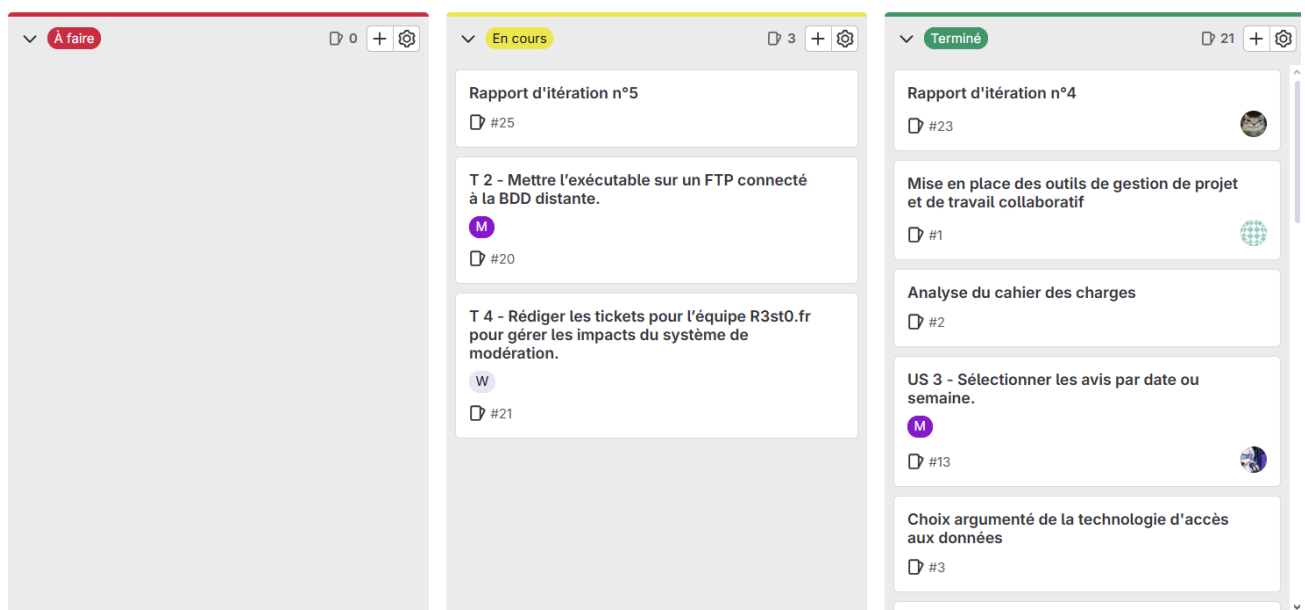
2.1. Gestion de projet

Comptes-rendus des Daily Scrums :

Date	Participants	Avancement	Problèmes rencontrés	Prochaines actions
24/02/2025	Loric Anthony Titouan Stefen	Argumentation du choix de la technologie et P.O.C effectué	Problème de synchronisation avec le fichier restoJdbc dans le dossier « src » et « build »	Maquette à terminer, déterminer les prochains objectifs du projet.
03/03/2025	Loric Anthony Titouan Stefen	Avancement sur la détermination de la priorisation des tâches	x	Commencer US 1 et T 3 .
10/03/2025	Loric Anthony Titouan Stefen	Réorganisation des tâches prioritaires	x	Commencer US 3 , US 5 , US 6 , US 4 et US 7 .
17/03/2025	Loric Anthony Titouan Stefen	Organisation des tâches prioritaires	x	Commencer US2 , T 1 , T 2 et T 4 .
24/03/2025	Loric Anthony Titouan	Organisation des tâches prioritaires	x	Terminer T 2 et T 4 .

	Stefen			
31/03/2025	Loric Anthony Titouan Stefen	Organisation des tâches prioritaires	x	tâches terminées.

Répartition des tâches :



Dépôt de code :

- Adresse du dépôt : [https://gitlab.com/APeixoto/modo-modo]
- Branche de travail de l'itération : [Main]

2.2. Code source

- **Branche Git associée :**
[https://gitlab.com/APeixoto/modo-modo/-/tree/main/ModerationResto?ref_type=heads]
- **Évolutions du code :**

- Depuis l'application, le champ permettant au modérateur d'afficher les commentaires selon une date précise a été changé par un Jcalendar, plus pratique et intuitif :


```

private void jButtonDateActionPerformed(java.awt.event.ActionEvent evt) {
    SimpleDateFormat f = new SimpleDateFormat(pattern: "yyyy-MM-dd");
    String dateString = null;

    dateString = f.format(date: jCalendar1.getDate());

    try {
        ArrayList<Critiquer> commentaires = CritiquerDAO.getCommentairesByDate(dateStr: dateString);

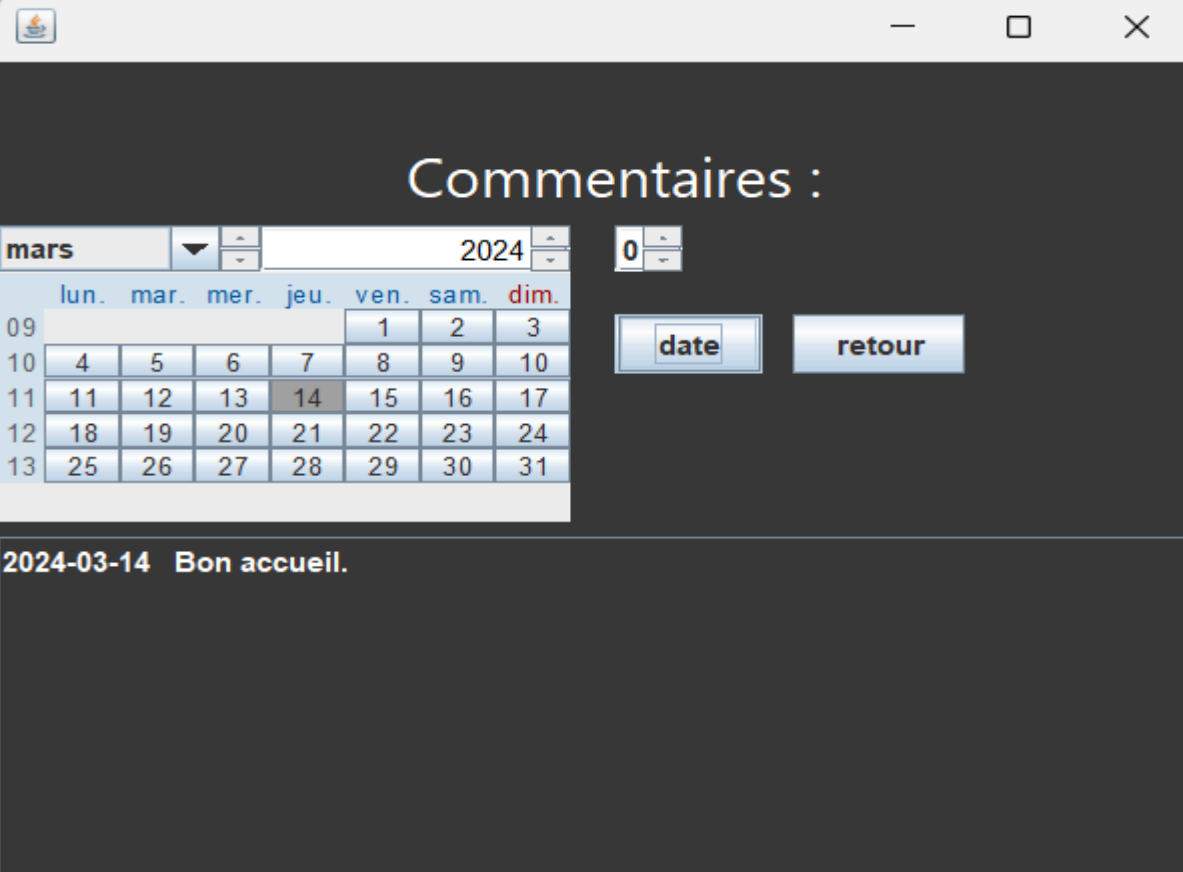
        // Création d'un modèle de liste
        DefaultListModel<String> model = new DefaultListModel<>();

        // Ajout des commentaires au modèle
        for (Critiquer commentaire : commentaires) {
            model.addElement(commentaire.getDate().toString()+" " +commentaire.getCommentaire());
        }

        // Application du modèle à la JList
        jListResto.setModel(model);
    } catch (IOException ex) {
        Logger.getLogger(name: Moderateur.class.getName()).log(level: Level.SEVERE, msg:null, thrown:ex);
    } catch (Exception ex) {
        Logger.getLogger(name: Moderateur.class.getName()).log(level: Level.SEVERE, msg:null, thrown:ex);
    }
}

```

- Résultat obtenu avec la date suivante : 14/03/2024 :



Commentaires :

mars 2024 0

	lun.	mar.	mer.	jeu.	ven.	sam.	dim.
09					1	2	3
10	4	5	6	7	8	9	10
11	11	12	13	14	15	16	17
12	18	19	20	21	22	23	24
13	25	26	27	28	29	30	31

date retour

2024-03-14 Bon accueil.

2.3. Déploiement du projet

- Avant de déployer le projet sur le serveur, nous avons modifié les paramètres de connexion pour permettre à l'application d'interagir avec la base de donnée distante et non celle en locale :

```
rest JDBC.properties X
C: > Users > loric > Documents > BTS_SIO > 2ème année > AP > modo-modo > doc > rest JDBC.properties
1 #####
2 # Paramètres de connexion JDBC
3 # Utilisation : d\commenter les paramètres spécifiques au SGBD vis\
4 # Les paramètres suivants doivent être renseignés : url, utilBD, mdpBD
5 #####
6
7 # MySQL / MariaDB local : mysql-connector-java-8.0.18.jar
8 url=jdbc:mysql://10.15.253.248:3306/2slamprj3eq4_resto2
9 utilBD=2slamprj3eq4
10 mdpBD=opii1qG01jU9yav7ez3jzw==
11 #url=jdbc:mysql://localhost:3306/resto2
12 #utilBD=resto_util
13 #mdpBD=fnw7bmWaVAGHA62TyMOyTQ==
14 key=MySuperSecretKey
15
```

NB : On a utilisé le fichier "TestCryptoUtils.java" pour pouvoir crypter le mot de passe du compte d'accès de notre groupe.

2.4. Priorisation des tâches

Estimation des complexités manquantes :

ID	Description	Criticité	Complexité (estimée)
US 1	Afficher la liste des avis par date décroissante.	M	S
US 2	Authentification avec rôles et stockage en BDD.	M	L
US 3	Sélectionner les avis par date ou semaine.	M	M

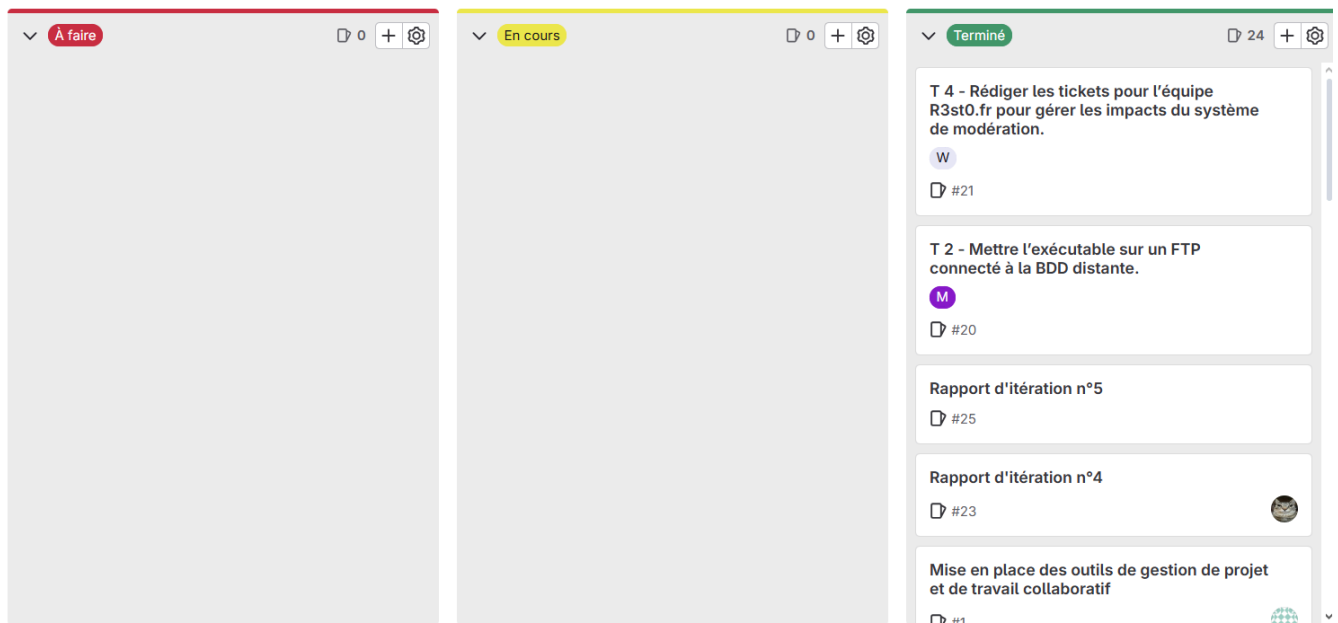
US 4	Filtrer par « avis masqués ».	W	S
US 5	Masquer un avis.	S	S
US 6	Démasquer un avis pour le republier.	S	S
US 7	Supprimer un avis inopportun.	C	S
T 1	Déployer la nouvelle BDD sur le serveur.	M	S
T 2	Mettre l'exécutable sur un FTP connecté à la BDD distante.	M	M
T 3	Sécuriser les accès BDD (config séparée, chiffrement).	S	M
T 4	Rédiger les tickets pour l'équipe R3st0.fr pour gérer les impacts du système de modération.	W	M

A faire :

- **Restant : toutes les tâches prévues ont été réalisées.**

2.5. Bilan de l'itération

- **Travail réalisé :**



- **Modification** : ajout d'un **Jcalendar** sur la vue du modérateur pour la sélection d'une date précise et changement des **paramètres de connexion** pour se connecter à la base de données serveur.

- **Déploiement** : Comme demandé dans le **T 2**, le projet a été déployé sur le serveur, l'action a été réalisée depuis FileZilla.

- **Rédaction** : Des tickets pour l'équipe de R3sto.fr ont été rédigés, requis pour le **T 4**, afin de gérer les impacts du système de modération sur leur site.

- **Travail restant à faire :**

- Projet terminé.

- **Difficultés rencontrées :**

- Retard de remise du projet dû au problème avec le certificat SSL.