Université de Rouen

Département Informatique - Master Sécurité des Systèmes d'Informations

Rapport de projet

Projet de Langages Web

Auteurs :
Sami Babigeon
Louka Boivin

Enseignant :
M. Florent Nicart

30 décembre 2021

Table des matières

1 Introduction

Le but de ce projet est de développer une plateforme de signalement pour la maintenance des infrastructures d'une organisation. Cette plateforme permettra aux usagers de signaler une anomalie sur une ressource simplement en scannant un QR code.

1.1 Exigences fonctionnelles

On pourra distinguer trois types d'utilisateurs sur cette plateforme :
- Les utilisateurs anonymes qui peuvent signaler des anomalies en flashant le QR associé à une ressource et en décrivant l'anomalie.
- Les responsables de maintenance qui peuvent gérer les ressources et les tickets d'anomalies. Chaque responsable possèdent un ensemble de ressources dont ils sont les seuls à s'occuper.
- Enfin, un administrateur (unique) qui peut gérer les responsables de maintenance et transférer les droits sur les ressources.

Pour chaque ressource un utilisateur pourra déclarer une nouvelle anomalie ou bien séléctionner une parmi une listes d'anomalies déjà entrée par les autres utilisateurs.

On devra pouvoir imprimer sur la plateforme les étiquettes avec les QR code et les URLs correspondant aux ressources. Les étiquettes devront avoir un format de 105x42 mm une fois imprimées et l'impression devra ce faire depuis un rendu Web sans générer de PDF.

2 | Architecture

Cette section détaille l'architecture que nous avons choisi pour créer la plateforme.
Elle explique aussi les choix de technologies que nous avons fait et le déroulement du développement dans les grandes lignes.

2.1 Modèle de la base de données

La première étape de la conception était de décider du modèle de la base de données. Le schéma suivant décrit le schéma des tables que nous avons choisi :

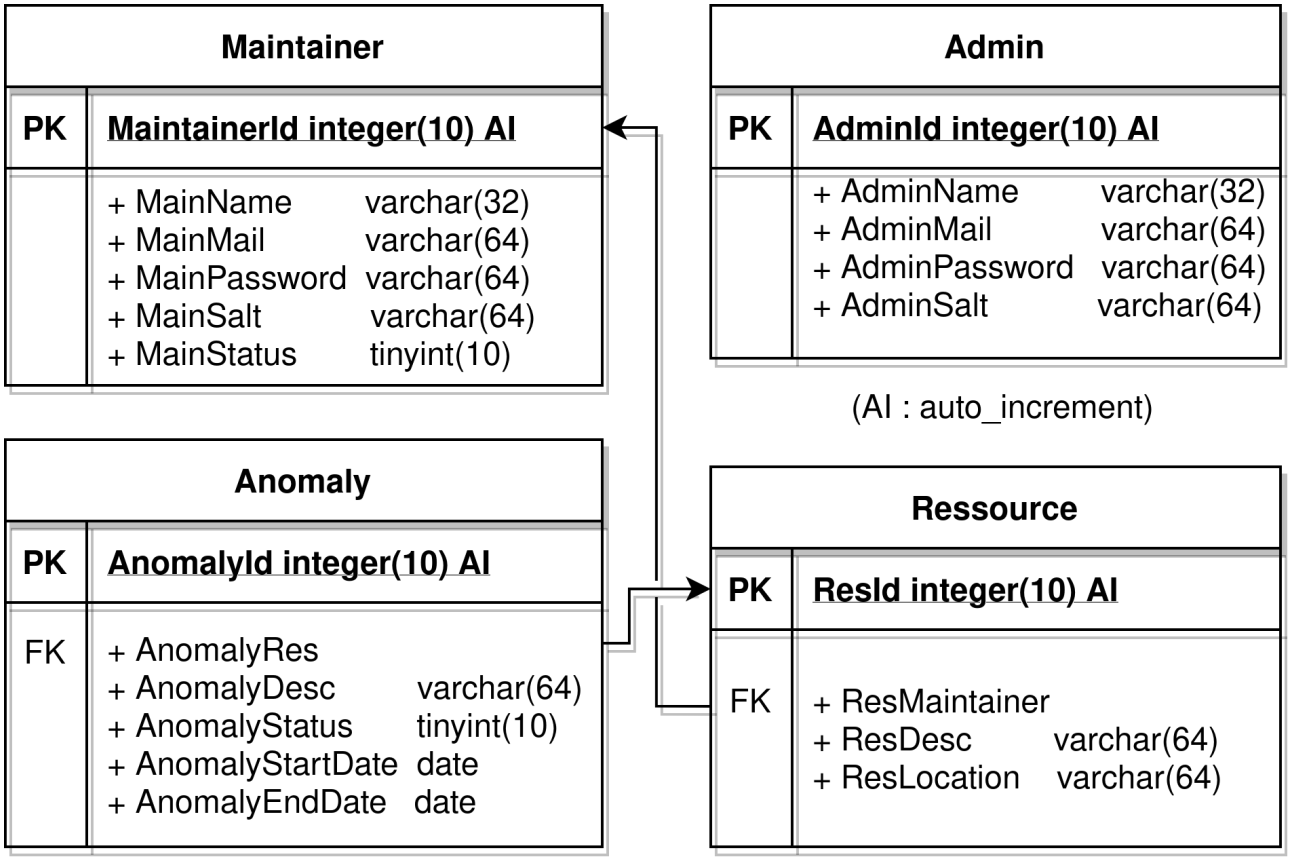

FIGURE 2.1 – Schéma des tables de la base de données

2.2 Technologies choisies

La partie vue est écrite dans les langages classiques : HTML & CSS. Nous n'avons pas utiliser de JavaScript pour la plateforme.

Nous avons choisis d'utiliser Java avec les Servlets et les JSP pour développer la partie contrôleur de l'application.

Le modèle est donc écrit en Java en utilisant la bibliothèque JDBC pour les interactions avec le SGBD (MySQL).

2.2.1 Frameworks

Pour développer la plateforme nous avons utilisés un framework pour le CSS : **Bootstrap**. Cela nous a permis de simplifier grandement le développement du front-end afin de se concentrer sur le back-end.

2.3 Modèle

L'application a été conçu en suivant le modèle MVC (Modèle-Vue-Contrôleur). C'est le modèle le plus simple et le plus utilisée et également celui que l'on a vu en cours donc c'était un choix évident.

Nous avons premièrement implémenter des classes Java décrivant simplement le modèle des types d'objets utilisés, c'est-à-dire :
- `User` : classe des utilisateurs,
- `Admin` : classe des administrateurs, héritant de `User`,
- `Maintainer` : classe des responsables de maintenance, héritant de `User`,
- `Ressource` : classe des ressources,
- `Anomaly` : classe des anomalies.

Ensuite pour chacune des classes exceptées `User` nous avons écrit une classe correspondante qui fait l'intermédiaire entre le SGBD et le modèle, selon le modèle DAO (Data Access Object).

Les DAO nous permettent donc de récupérer un élément de la base de donnée directement sous forme d'objets du modèle. Ensuite nous pouvons donc facilement manipuler ces objets à notre guise puis si besoin les renvoyer au DAO afin de mettre à jour la base de données selon l'état de cet objet.
Un exemple d'utilisation du DAO est illustré avec la figure 2.2.

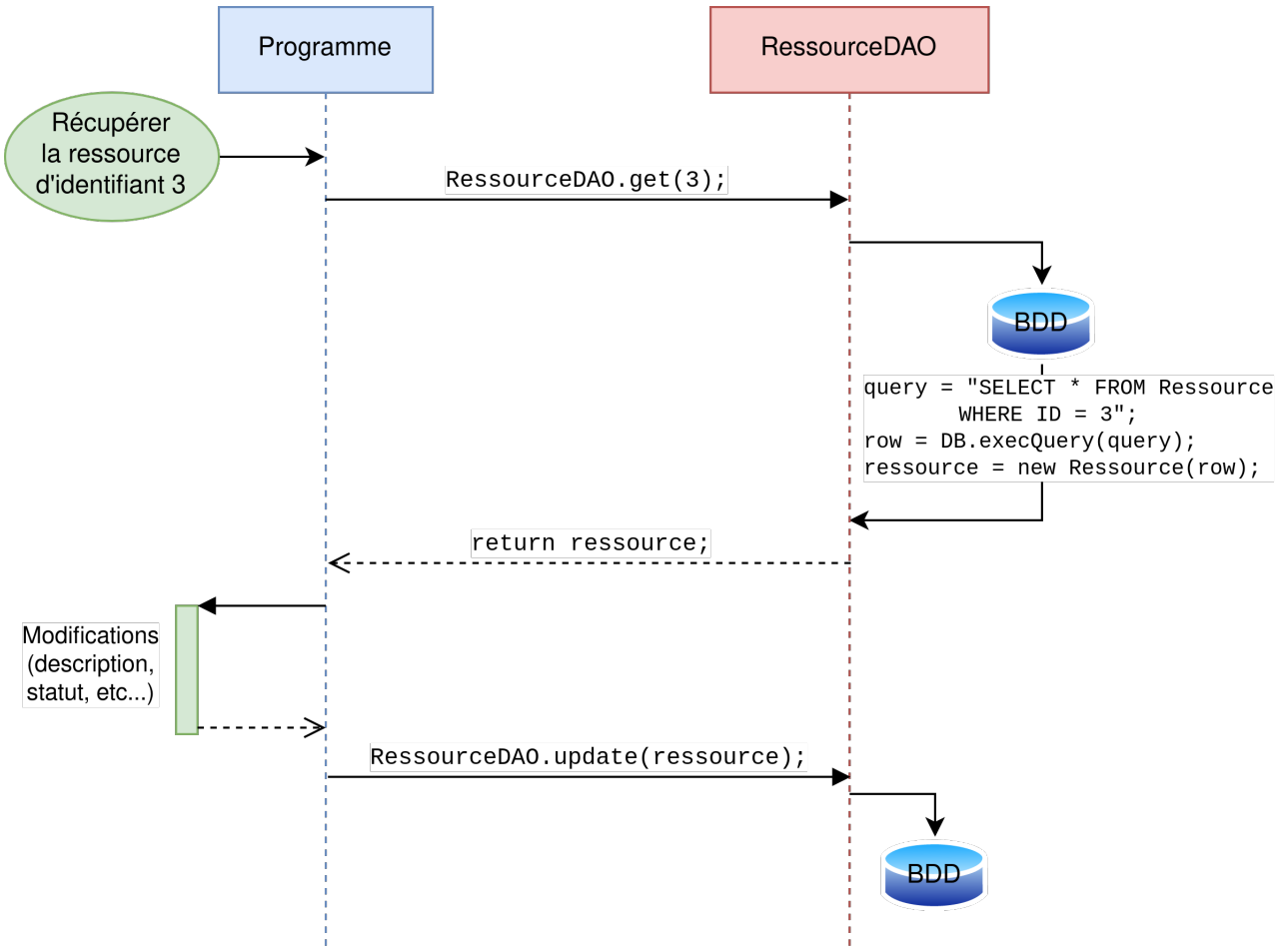

FIGURE 2.2 – Exemple d'utilisation de la classe RessourceDAO

2.4 Sécurisation

Afin de sécuriser au mieux la plateforme nous avons mis en place les sécurités suivantes :

Filtrage des urls : via un filter JEE.

Requête préparées : en utilisant la bibliothèque JDBC.

Une fois ses différentes sécuritées mises en place, nous avons testé leurs fiabilités en essayant d'effectuer différentes actions :

— Obtenir sur la plateforme des droits que nous avons pas (broken access control).
— Injections XSS.
— Injections SQL.

3 Difficultés

Durant ce projet nous avons eu des difficultés qui ont fait que celui ci nous as pris beaucoup plus de temps que prévu.

La première difficulté fut de réussir à interagir avec notre base de données en utilisant Java. Après quelques échecs et des recherches sur Internet nous avons trouvé le problème : l'ensemble de classes qui permet les interactions avec un SGBD (JDBC) n'est pas intégré à l'API Java de base. Il fallait donc trouver, télécharger et inclure au projet la librairie le permettant (`mysql-connector-java.jar`).

Après avoir écrit le modèle et la vue du projet, nous avons eu quelques difficultés au moment d'écrire le contrôleur [... à expliquer ...].

Également le fait d'utiliser Java comme langage de programmation de la plateforme nous as poussé à utiliser Eclipse comme IDE. Cela permet de gagner beaucoup de temps sur certains points (compilation, tests, auto-complétion, vérification de la syntaxe) mais aussi nous en as fait perdre énormément à causes de problèmes pour ajouter des librairies, des différentes versions et de leurs compatibilités, de bugs etc...

Aussi, le fait d'utiliser un IDE aussi puissant avec beaucoup d'outils qui permet de se passer de certaines tâches comme Eclipse fait que nous ne connaissons pas tout le fonctionnement derrière. Et donc, pour une tâche comme le déploiement du site sur la VM, nous avons eu des problème. Par exemple, les librairies externes ne se liaient pas correctement, nous avons cherché longtemps avant de comprendre qu'il fallait les copier dans un répertoire spécifique et redémarrer Tomcat (voir `install.sh`).

4 Conclusion

En conclusion, malgré les difficultés rencontrées, nous avons pu terminer le projet, et celui-ci est fonctionnel. Les différentes exigences et contraintes ont été respectées.

Ce projet nous as permis de mieux comprendre l'architecture d'une application Web, le modèle MVC et la sécurisation de celle-ci. Également il nous as fait manipuler les Servlets et JSPs qui facilite beaucoup l'implémentation d'une application en suivant le modèle MVC.

5 Annexe

5.1 Notice d'utilisation

L'installation peut se faire de deux manières différentes :

1. Premièrement, copier l'archive `install.tar.gz` sur la machine cible, ensuite l'extraire puis lancer le script `install.sh` avec les droits administrateurs.
2. Sinon l'application peut être installer directement depuis la machine cible en recupérant le script d'installation sur GitHub et en lançant le script comme precédemment :

```
wget https://raw.githubusercontent.com/loukabvn/projet-web/main/install.sh
chmod u+x install.sh
su root
./install.sh
```

Il faut ensuite renseigner les informations demandées c'est-à-dire les identifiants pour le compte MySQL pour l'accès administrateur à la base de données et les identifiants pour le compte administrateur de la plateforme.

Une fois l'installation terminée vous pouvez vous rendre sur :

`http://192.168.76.76:8080/ProjetWeb/home`

et vous connecter avec les identifiants renseignés.

5.2 Jeu de données

Un jeu de données pré-rempli est disponible pour effectuer des tests sur notre plateforme. Pour pouvoir utiliser ces données il suffit d'exécuter avec mysql le script `insertion.sql`. Également, durant l'installation, l'insertion de ce jeu de données de test vous ait proposé, que vous pouvez accepter ou refuser.