Bachelier en informatique de gestion – Bloc 3

Applications informatiques – SGBD V

Projet FabricTout

# Description de l’application :

Nous avons développé une application web destinée à une usine souhaitant gérer ses machines et leurs maintenances. Nous avons utilisé JEE et ses différentes fonctionnalités pour remplir cette mission. Une API de type Restful java permettant de gérer tous les appels à notre base de données Oracle.

Concrètement, l’application va permettre :

* aux ouvriers de l’usine d’avoir leurs listes de maintenances en cours ainsi que l’historique. L’ouvrier pourra rédiger un rapport pour chacune de ses maintenances traitées. Il pourra aussi marquer les maintenances finies comme « done » afin d’avertir son responsable.
* aux responsables de maintenances de s’assurer du bon fonctionnement des machines se situant sur leurs sites. Ils peuvent faire des demandes de maintenances à un ou plusieurs ouvriers si l’état de la machine le requiert. Ils pourront également consulter les maintenances réalisées sur chacune des machines et valider ou non les maintenances marquées comme « done » par les ouvriers.
* Aux employés sur service achat accéder à l’ensemble des maintenances effectuées sur chacunes des machines que possède l’usine. Si une machine dépasse un certain nombre de maintenances, l’employé pourra rechercher dans la liste des machines des fournisseurs une machine correspondante afin de remplacer l’appareil.

## Pour l’ouvrier :

Une fois connecté, l’ouvrier arrive sur la page de consultation des maintenances :

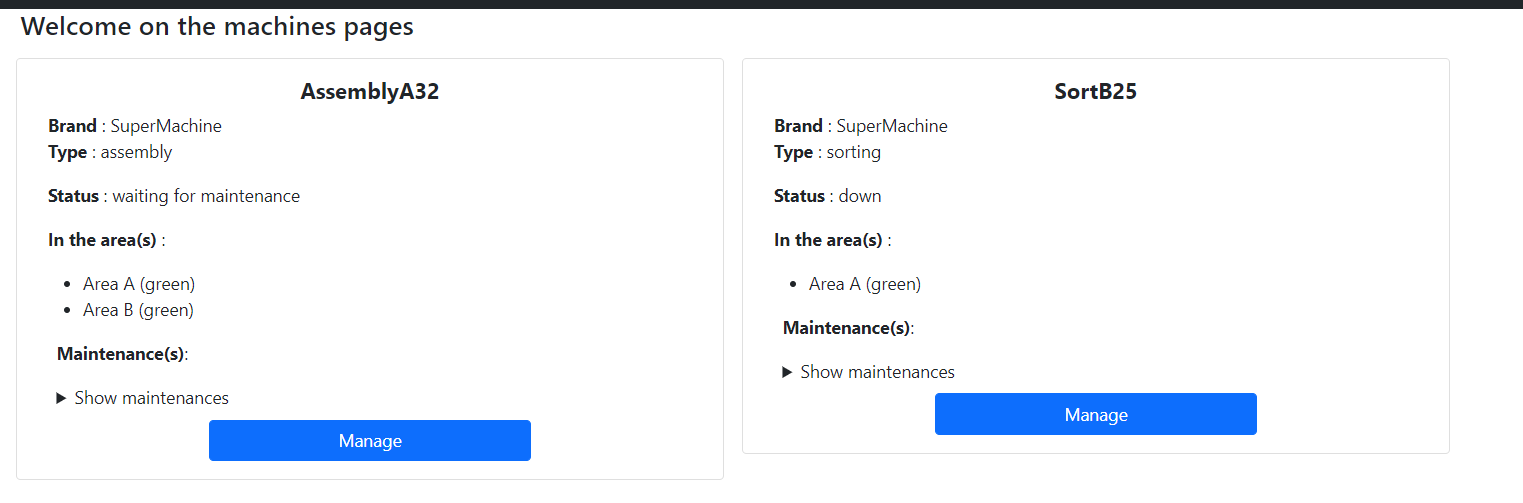
Une image contenant table

Description générée automatiquement

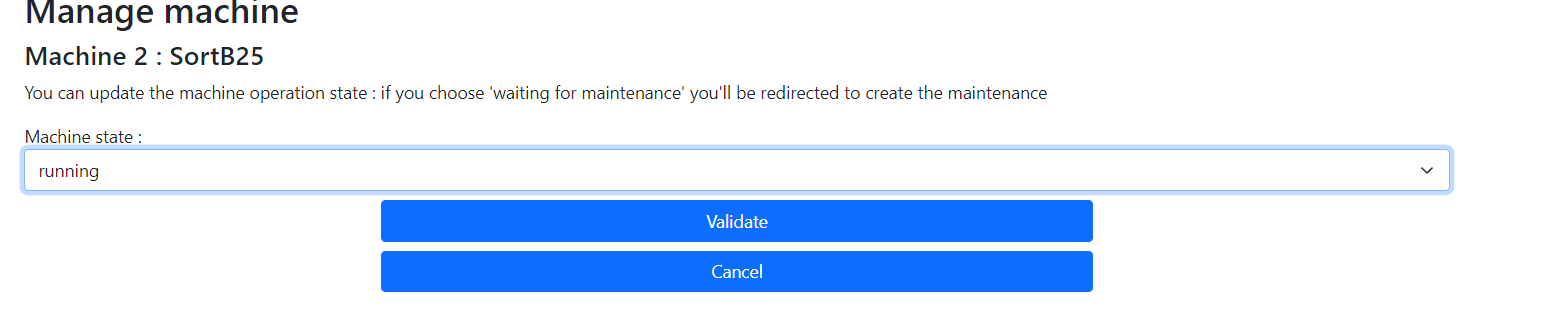
Afin de pouvoir mettre une maintenance comme « done » et/ou faire son rapport il doit appuyer sur le bouton « consult ».

## Pour le responsable :

Une fois connecté, le responsable est sur la page des machines et peut consulter leurs maintenances en cliquant sur le « > Show maintenances ».  Il peut également gérer le statut d’une machine en cliquant sur « manage ».



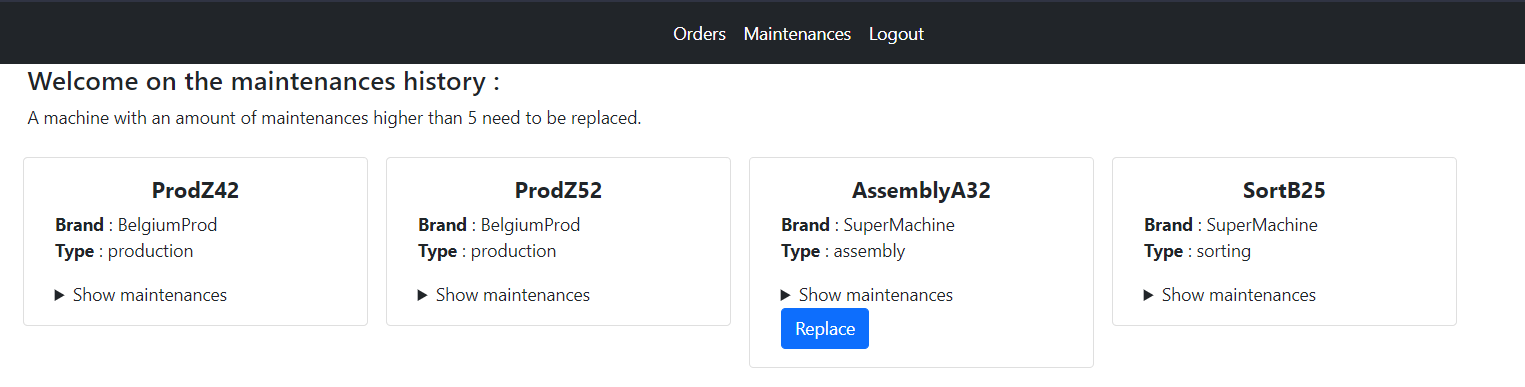
Après avoir cliqué sur « manage », il peut modifier le statut de la machine s’il le juge nécessaire :



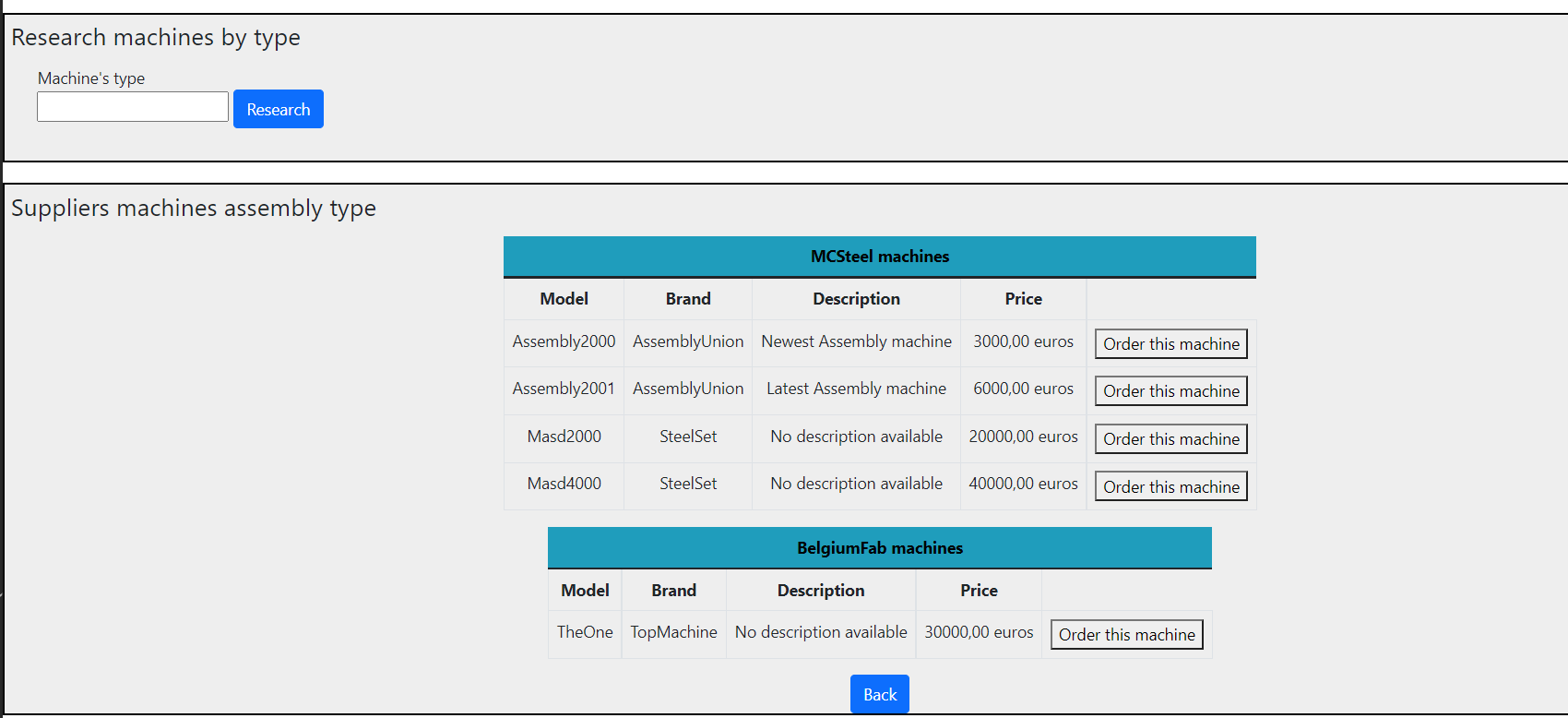
Ensuite on peut valider ou annuler.

## Pour l’employé :

Une fois connecté, l’employée arrive sur la page de l’historique des maintenances triées par machine. Si le nombre de maintenance par machine dépasse un total de 5, alors on considère que la machine est à remplacer et un bouton « Replace » sera disponible.



Après avoir appuyer sur le bouton replace, l’employé peut choisir une seule machine pour remplacer l’autre :



Il suffit de cliquer sur « order the machine » si on veut commander une machine et ensuite de confirmer.

L’employée possède une autre page « Orders » qui est dans la navbar, permettant de consulter l’historique des commandes de l’entreprise.

# Mode d’emploi installation partie JEE et API :

Pour importer la partie JEE , rendez vous dans éclipse et créons un nouveau workspace

* Placer vous dans le dossier d’Eclipse : Une image contenant table

  Description générée automatiquement
* Copier coller le dossier pour le workspace contenu dans « workspace\_project ».

Une fois arrivée sur le workspace cliquez sur :

Une image contenant texte

Description générée automatiquement

Et ensuite allons chercher le dossier du projet JEE et ensuite celui de l’API

Une image contenant texte

Description générée automatiquement

//IMAGE

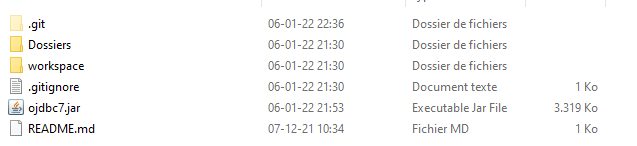
Cliquez sur finish. Refaire la même chose pour l’API au niveau du import projects et tout sera bon.

Assurez vous bien que vous avez un server TomCat déjà sélectionné sur votre workspace.

# Mode d’emploi installation partie API  :

# Avant tout :

Nous vous demandons de nous excuser pour la latence de certaines fonctionnalités, nous avons essayé d’optimiser cela au mieux mais nous devions récuperer beaucoup de données (essentiellement lors du chargement des pages concernant les machines).

Afin d’optimiser notre programme au mieux pour les appels à la D.B, nous avons utilisé ojdbc7.jar en plus de ojdbc6.jar car nous avions besoin du type Sys\_cursor pour nos appels à nos procédures stockées. Le voici à la racine du dossier : 

Nous vous demandons juste si jamais de bien vouloir faire cette manipulation sur le dossier « JEE-API » :

Une image contenant texte, moniteur, capture d’écran, écran

Description générée automatiquement



Une fois après avoir cliqué sur « Add External JARs » , sélectionnez le fichier « ojdbc7.jar » dans le dossier du projet à l’endroit où vous l’avez stocké. Ensuite « Apply And Close ».

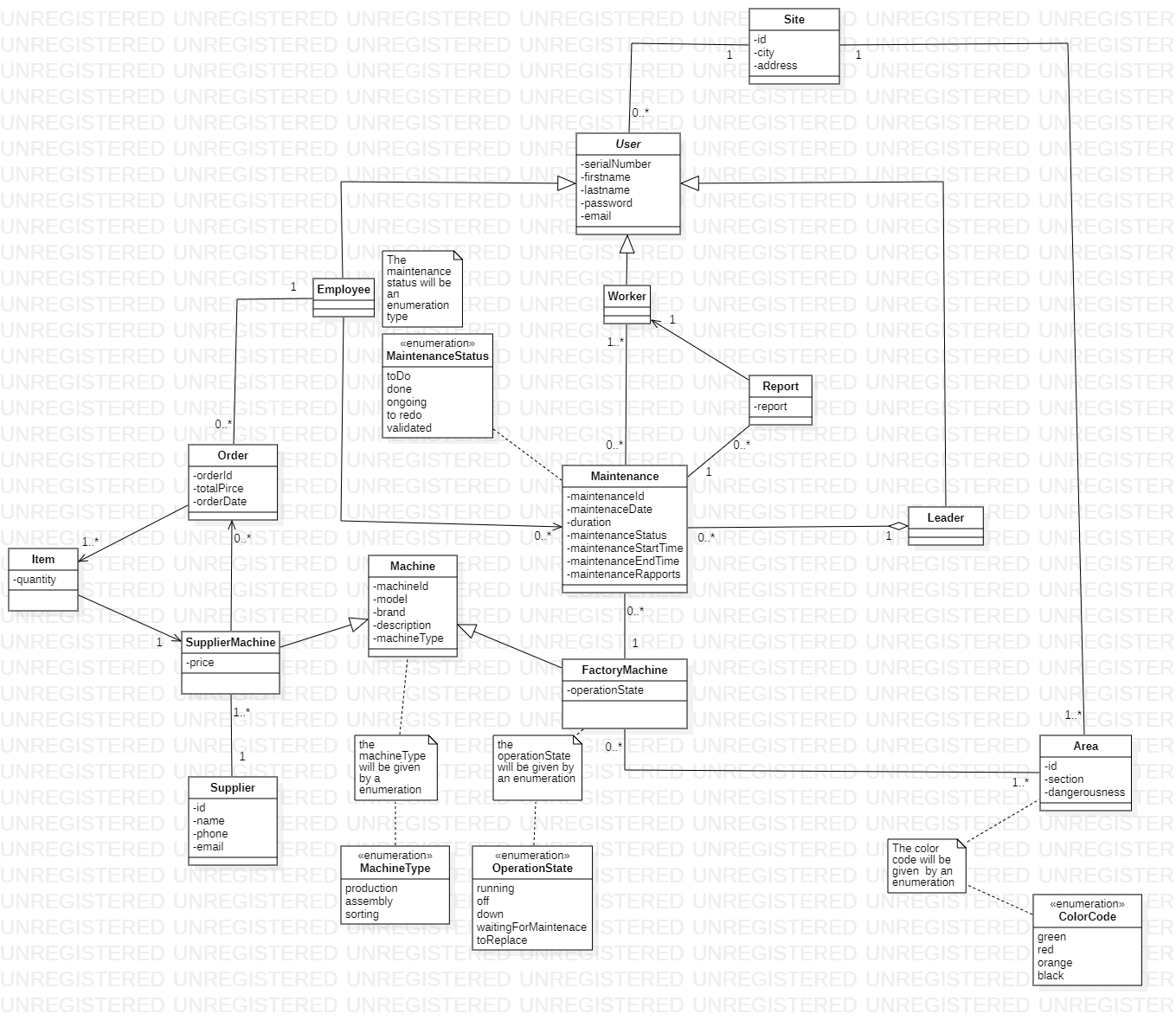
Adresse de l’application :<http://localhost:8080/JEE_Mehdi_Simon/connexion>

# Comptes opérationnels :

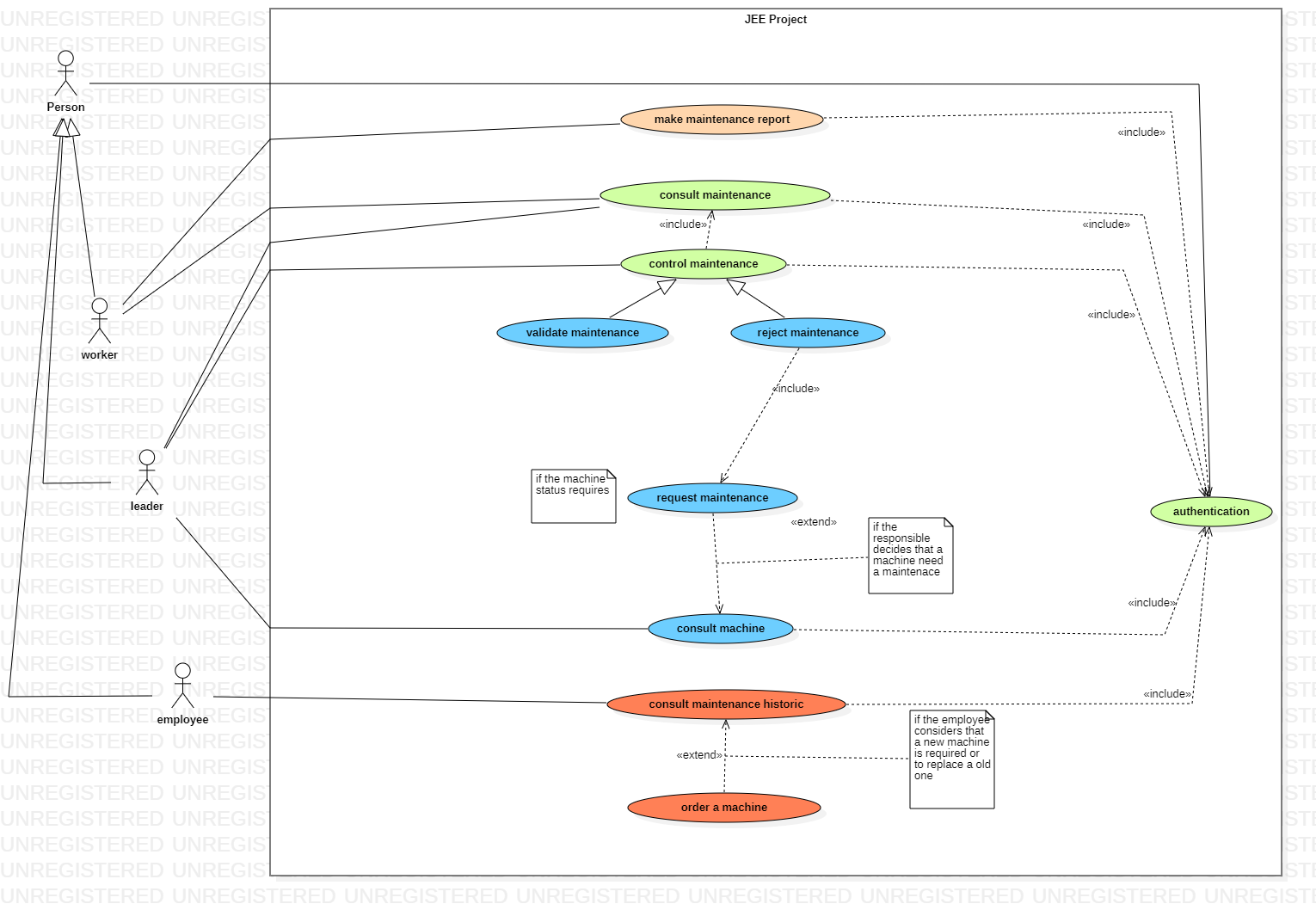
|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Type | Identifiant | Mot de passe |
|  | **Ouvriers** |  |  |
| Daniel Craig |  | 20000 | Lstbnd070 |
| Michel Platini |  | 20001 | PlFr19or |
|  | **Responsables** |  |  |
| Martin Garix |  | 30000 | Mdrtsf85 |
|  | **Employés** |  |  |
| Clif Richard |  | 40000 | Clifsq8d |
| Dan Bison |  | 40001 | B3redape |
|  |  |  |  |

Le compte de base de données utilisé est « **student03\_06** », et le mot de passe est « **root** ».

# Diagramme de classes :



# Diagramme des uses cases :



# Le schéma conceptuel /ERD :