vendredi 9 juin 2017



Application web de gestion de matériel

LAMPIN DENOVAN, Julien Pruvost

lICENSE 3 informatique

Table des matières

[Introduction : 2](#_Toc484693860)

[Répartition des tâches 2](#_Toc484693861)

[Analyse 2](#_Toc484693862)

[Réalisation du projet 2](#_Toc484693863)

[Conclusion 2](#_Toc484693864)

[Annexes 2](#_Toc484693865)

# Introduction :

 Introduction : le sujet du projet (l'objectif) où vous décrivez ce qu'il y avait à faire, le contexte si besoin (si travail pour une association, ...), le(s) tuteur(s), le planning.

Le projet consisté au développement d’une application WEB permettant la gestion de matériels. Cette application devait permettre aux utilisateurs de consulter la liste des objets et leur quantité mais aussi d’emprunter ou de rendre des objets.

Un administrateur devait être capable d’inscrire un nouvel objet dans la base de de données et d’en supprimer.

Pour ce projet, certaines technologies nous étaient imposées :

* Java EE ;
* Utilisation des Servlets et des .JSP ;

Il nous était donc demandé de créer en plus de l’application, une base de données.

Il s’agit d’un projet qui s’étend sur la totalité du semestre 6 (Licence 3 informatique) dont notre tuteur est Monsieur Mikael Desertot.

# Répartition des tâches

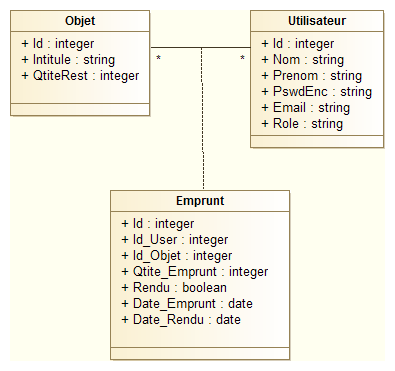
# Analyse

 L’analyse (avec diagramme de classes si dvpt objet et/ou MCD si il y a une base de données et/ou structure du site si dvpt web, etc...) : basez-vous sur les techniques apprises en génie logiciel. Décrivez et argumentez aussi dans cette partie les choix techniques faits.

Comme dit précédemment une base de données a dû être créée. Nous avons choisis la base de données Hyper SQL. En ce qui concerne la structure, cette base de données se compose de trois tables :

* Une table User, permettant de stocker les informations de l’utilisateur ;
* Une table Objet, permettant de stocker les différents objets ;
* Une table Emprunt, permettant de stocker toutes informations liées à un emprunt.

Voici le schéma de cette base de données.



Nous pouvons voir ici en détails la structure de la base de données ou nous avons dans la table utilisateur un identifiant, un nom, un prénom, un mot de passe encodé, un email (qui servira de login de connexion) ainsi qu’un rôle qui est soit « user » soit « admin ».

La table objet ne contient qu’un identifiant, un intitule (le nom de l’objet) ainsi que la quantité qu’il reste.

Enfin, nous avons la table Emprunt qui est une table associative entre utilisateur et objet ou nous retrouvons les identifiants de l’objet et de l’utilisateur (id\_user et id\_objet). L’emprunt est aussi indexé à l’aide de son id. Nous pouvons également observer la quantité empruntée par l’utilisateur ainsi que les dates de l’emprunt et de rendu. Nous avons pour finir le champ Rendu qui permet de savoir si un objet est rendu ou non. *Bien que depuis que la date de rendu est enregistrée ce champ n’est plus nécessaire.*

# Réalisation du projet

la réalisation : ce qui a été développé, avec les problèmes techniques rencontrés et solutions apportées. quelques "morceaux" de code sont acceptés dans cette partie pour illustrer ce que vous décrivez mais en aucun cas des pages entières de code. Il est aussi possible de faire référence à du code fourni en Annexe, à la fin du rapport.

Nous avons tous deux commencé à analyser le problème ainsi que deux réfléchir à une structure possible de l’application.

Ensuite chacun de notre côté nous avons réalisé un tutoriel sur le WEB (lien [ici](https://openclassrooms.com/courses/creez-votre-application-web-avec-java-ee)), nous permettant de connaître les bases du JEE.

Une première application a été créée qui nous a permis simplement de nous connecter, d’emprunter et de rendre un objet. A ce moment, il n’a avait aucune gestion des erreurs c’est-à-dire que lorsque qu’une erreur survenez, l’utilisateur connecté n’en avait aucune information. Ensuite cette application ne permettez pas l’inscription d’un nouvel utilisateur.

# Conclusion

conclusion : bilan sur ce qui a été fait, reste à faire, améliorations possibles, ce que ça vous a apporté.

Pour conclure, notre application est fonctionnelle ; elle permet à un administrateur de créer ou de supprimer un objet dans la base de données. Elle permet également à un utilisateur de réserver un nouvel objet avec la quantité souhaité à la condition qu’il reste suffisamment d’objet.

S’il y a une erreur dans la manipulation, l’application nous indique sa nature.

Pour ce qui est des améliorations possibles :

Nous pourrions à l’aide de struts2, gérer plus facilement les dépendances.

# Annexes