IUT Sénart-Fonatinebleau
département informatique
Route forestière Hurtault
77300 Fontainebleau

Ubistock

Projet de stage

Cahier des charges

Année scolaire 2019-2020

Tutrice de projet : Régine Laleau

Par :
- Louka Doz
- Thibault Meynier

Table des matières

Synthèse

L'objectif est de réaliser une application web qui permette de gérer les stocks d'une entreprise à la manière d'un système de fichiers. L'équipe est composée de 2 membres. Le projet doit être terminé pour la semaine du 22 juin. L'enseignante Régine LALEAU sera notre tutrice. Le code source, sera disponible sur le serveur Git du département informatique de l'IUT : https://dwarves.iut-fbleau.fr/git/meynier/Ubistock.

1. Présentation du projet et de l'équipe

1.1 Projet

Le projet est une application web permettant à une entreprise de trier, organiser et gérer tous types de stocks.

1.1.1 Systèmes similaires

L'application se présente sous la forme d'un système de fichiers. Cela signifie que l'on peut créer, supprimer, renommer ou effacer par exemple, de la même manière que pour les fichiers et dossier sur un ordinateur. Les stockages sont comparables à des dossiers et les ressources sont comparables à des fichiers.

1.2 Equipe

L'équipe se compose de deux étudiants en IUT informatique : Louka DOZ et Thibault MEYNIER

2. Spécifications fonctionnelles

Les fonctionnalités sont ordonnées par importance avec la notation MoSCoW (Must,Should,Could et Would, du plus au moins important).

2.1 API PHP

La première étape du projet est de créer une partie serveur de l'application pour interragir avec la base de données. Les modifications se feront par l'intermédiaire de requêtes GET sur les différentes pages PHP du serveur.

Cette API devra permettre de :

M Créer un stock ou une ressource ;

M Supprimer un stock ou une ressource ;

M Augmenter ou diminuer la quantité d'une ressource ;

M Déplacer un stockage ou une ressource ;

S Renommer un stockage ou une ressource.

L'API est accessible via le lien suivant : http ://dwarves.iut-fbleau.fr/ meynier/stockAPI/

2.2 Interface graphique

Pour faciliter la prise en main et améliorer l'ergonomie, il faut également créer une interface graphique qui soit intuitive. Cette interface devra permettre d'effectuer toutes les actions de l'API ainsi que les actions suivantes si les contraintes de temps le permettent :

M Afficher tous les stockages, leurs noms, leurs ressources et la quantité de ces ressources ;

C Rechercher une ressource en masquant les stockages qui n'en ont pas ou qui n'ont aucun stockage enfant en possédant ;

C Lister toutes les ressources d'un stockage et de ses stockages enfants ;

S Définir des minimums aux ressources afin de pouvoir générer la liste des ressources manquantes pour satisfaire les minimums ;

C Générer des tableaux au format PDF téléchargeables par l'utilisateur pour les deux fonctionnalités précédentes ;

W Drag and Drop pour le déplacement de resources et stockages.

3. Spécifications techniques

3.1 Langage et Framework

Côté serveur, en ce qui concerne les scripts qui gèrent les requêtes, le langage utilisé est le PHP. Pour traiter les données, on utilisera également l'API SQLite ainsi que mySQL. Côté client, les requêtes vers le serveur sont effectuées avec l'API Javascript AJAX et les réponses sont traitées avec du JavaScript. L'interface homme-machine est en HTML et CSS avec l'utilisation du Framework Bootstrap, pour faciliter la création de l'interface, et de Riot.js pour la mise à jour des informations de la page. Le framework jsPDF (lien github : https://github.com/MrRio/jsPDF) sera également utilisé pour générer des fichiers au format PDF que l'utilisateur pourra télécharger.

3.2 Modèle de données

Les données sont stockées côté serveur à l'aide de SQLite ou mySQL dans une base de données.

Cf le diagramme UML disponible en Annexe, page 12

3.3 Architecture

L'application est codée avec PHP Storm et Web storm.

Le code source est sur GitHub via le lien : http ://dwarves.iut-fbleau.fr/ meynier/Ubistock

L'équipe projet tentera de maintenir une version à jour sur le serveur de l'IUT : http://dwarves.iut-fbleau.fr/~meynier/Ubistock

L'organisation des dossiers est la suivante :

Côté serveur :

- Les actions sont divisées en dossiers (un dossier pour chaque action), chaque dossier contient un script PHP nommé index.php ;
- Le dossier "includes" contient d'autres scripts PHP contenant diverses fonctions ;

– Le dossier “db” contient la base de données et les scripts SQLite et mySQL.

Coté client :

– Le site ne possède qu’une seule page nommé “index.html” ;
– Le dossier “css” contient les fichiers CSS ;
– Le dossier “js” contient les fichiers JavaScript ;
– Le dossier “tags” contient les fichiers Riot.js.

3.4 Convention

Le code et les commentaires sont en anglais.

3.5 Outils

PHP Storm comme IDE pour développer le projet ;

Discord pour la communication et l’organisation entre les membres de l’équipe ;

GitHub pour stocker le projet.

4. Charte Graphique

4.1 Couleurs

Pour le projet, il a été décidé d'utiliser les couleurs suivantes :

- Primaire : rouge (#c80000) ;
- Secondaire : gris foncé (#444).

4.2 Polices

Les polices utilisées sont les polices par défaut du Framework Bootstrap :

- Segoe UI ;
- Roboto ;
- Helvetica Neue, Arial, sans-serif ;
- Apple Color Emoji, Segoe UI Emoji, Segoe UI Symbol.

4.3 Icônes

Les icônes utilisées sont les icônes du Framework Bootstrap disponibles ici : https://fontawesome.bootstrapcheatsheets.com/

5. Planning et organisation

5.1 Planning

Il est prévu que l'API soit terminée pour le mois de juin.

Le reste du projet devra être terminé dans les deux semaines suivantes.

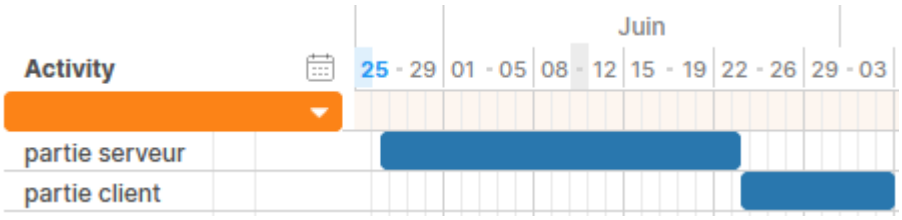

FIGURE 5.1 – Diagramme de Gantt du projet

5.2 Organisation

Les membres de l'équipe s'organiseront chaque jour sur les tâches faites et à faire via Discord.

Le projet étant découpé en beaucoup de fichiers, les membres du projet utiliseront une méthode de développement agile afin de développer chaque fonctionnalité indépendamment l'une de l'autre. Les membres de l'équipe ayant les mêmes compétences techniques, chacun pourra donc commencer le développement d'une nouvelle fonctionnalité lorsqu'il en aura terminé une.

6. Gestion des risques

Problème	Solution
Perte de code via une mauvaise manipulation	Logiciel de versionning, copie locale chez chaque membre de l'équipe
Membre du groupe dans l'incapacité de travailler	Les membres de l'équipe possèdent les mêmes connaissances. Quelques fonctionnalités peuvent tout de même être abandonnées car l'avancement du projet en sera ralenti.
GitHub indisponible	Copie locale chez chaque membre de l'équipe
Discord indisponible	Autres moyens de communication : téléphone, mails

Annexes

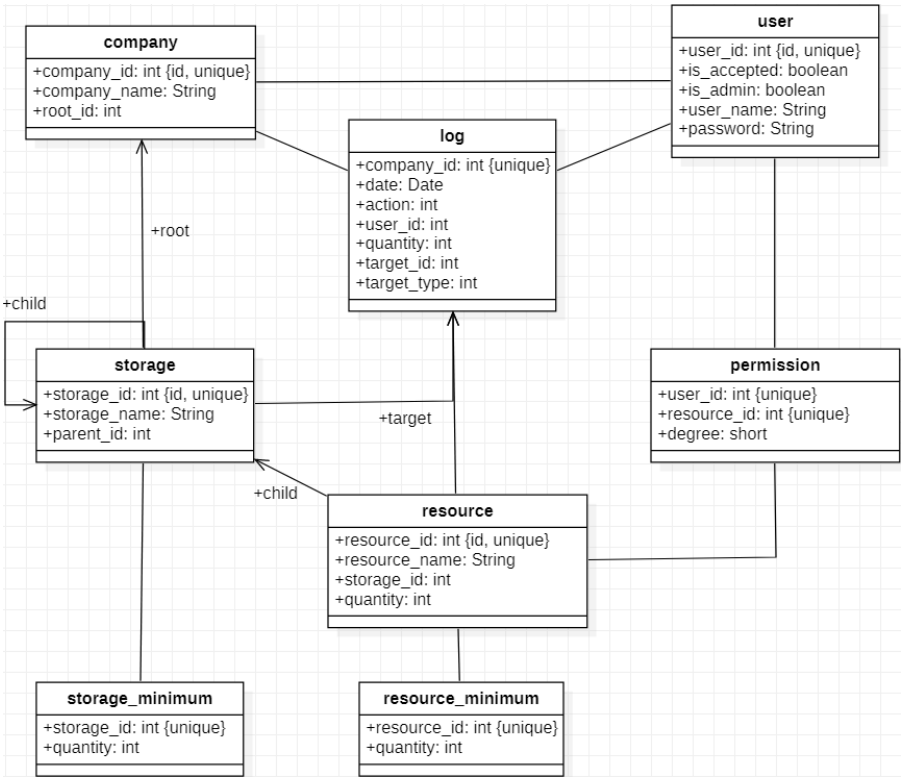

FIGURE 6.1 – Diagramme UML de la base de données