

A large, light blue abstract shape, resembling a stylized 'C' or a partial circle, is positioned on the left side of the page, extending from the top to the bottom.

DEVELOPPEMENT WEB 2

DOKS

Dan TSOUMBOU MOUTIMBA
Oussama TRABELSI
Solofo RABONARIJAONA
Kévin VOCANSON

INTRODUCTION

Nous avons choisi de créer un éditeur de code (HTML/CSS) en ligne, c'est à dire une plateforme permettant à ses utilisateurs de développer des pages HTML et CSS. Notre application fonctionnera sur un principe de projet, c'est-à-dire qu'un utilisateur pourra créer des projets, qui contiendront des fichiers HTML et CSS, ainsi qu'un chat dédié au projet. Un projet pourra être privé ou public, seuls les projets publics pourront être vu (et non modifiés) par tout utilisateur. Le créateur du projet en sera le chef, seul lui pourra inviter des collaborateurs, créer des nouveaux fichiers, supprimer des fichiers.

Notre éditeur de code possèdera la plupart des fonctionnalités classiques d'un éditeur de code, comme par exemple :

- Système d'arbre de fichiers/répertoires ;
- Système de d'auto indentation du code ;
- Système de visualisation du résultat en direct lors d'une modification simultanée par plusieurs utilisateurs ;
- Complétion automatique de balises ;
- Colorisation syntaxique ;

UTILISATION DES DIFFERENTES TECHNOLOGIES

- Swing, pour la réalisation de l'interface "lourde" ;
- Jsp ;
- Servlet,
- XML, utilisé pour le transfert de fichiers HTML et CSS;
- Sockets, utilisées pour faire communiquer le client lourd et le serveur ;

ATTRIBUTION DES ROLES

Dan :

- Réalisation de l'interface lourde avec Swing
- Réalisation de la base de données ;

Oussama :

- Système de connexion d'utilisateurs ;

- Base de données utilisateurs ;

Solofo :

- Interface web ;
- Parseur HTML,CSS / XML et inversement ;

Kévin :

- Système de sockets ;
- Système de merge ;

CONCEPTION DU PROJET

- Diagramme de cas d'utilisation

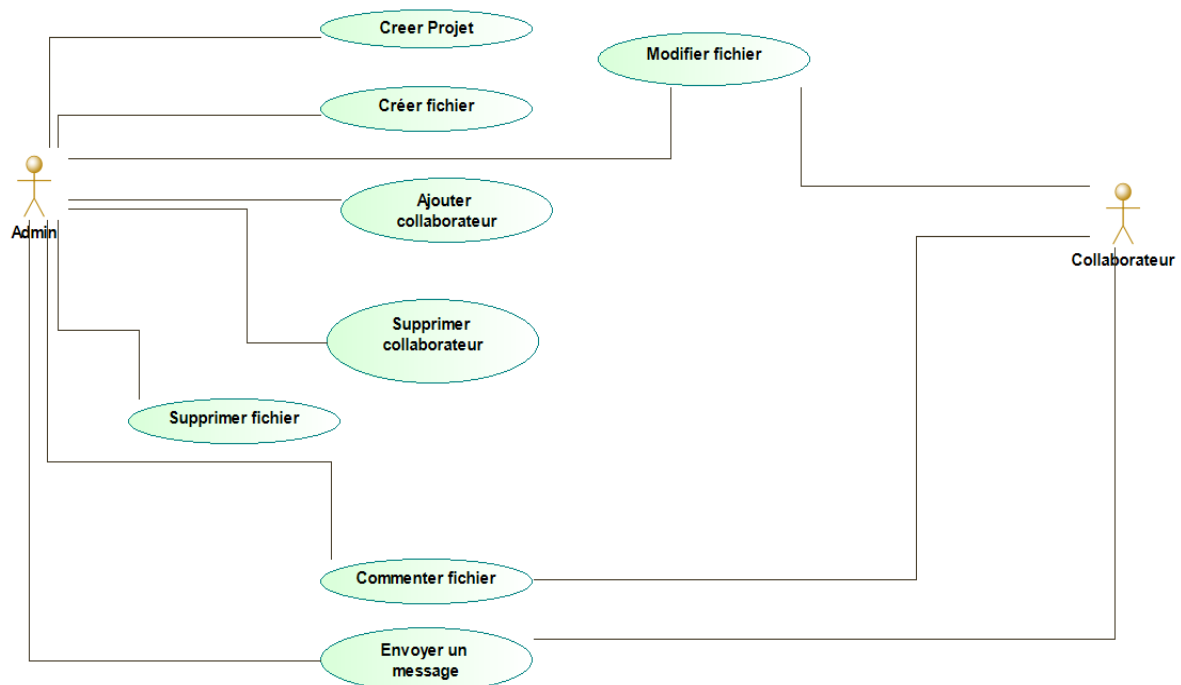


Fig 1 : Diagramme de cas d'utilisation

Ce diagramme de cas d'utilisation présente de façon simplifiée les principales fonctions qui seront implémentées lors de la réalisation du projet. Il est possible qu'il soit modifié d'ici la rédaction du rapport final.

Il n'a pas été modifié depuis son élaboration, car il résume suffisamment les interactions possibles entre un utilisateur et le système.

DIAGRAMME DE CLASSES

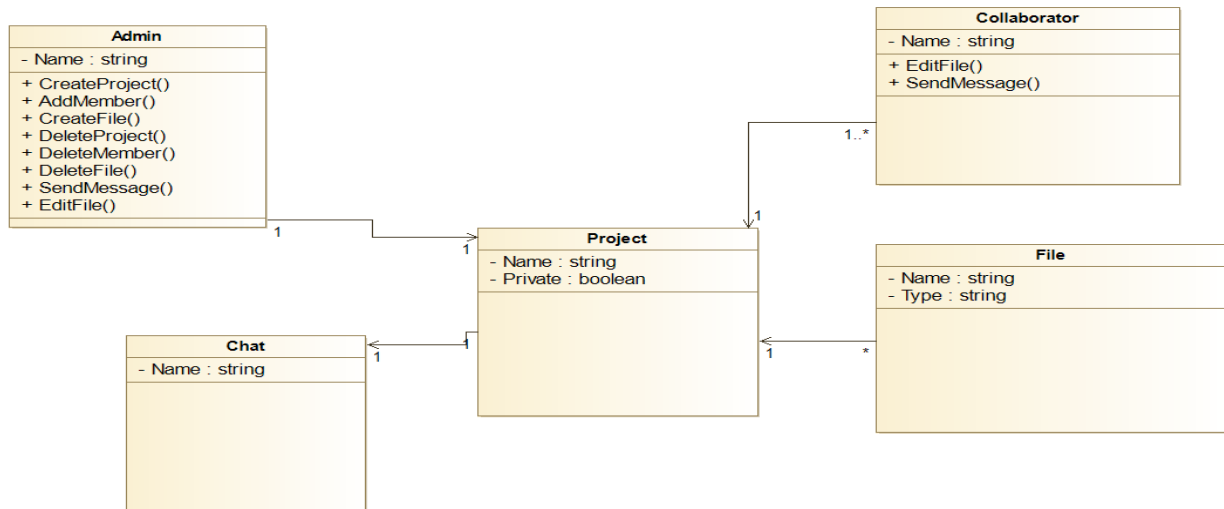


Fig 2 : Diagramme de classes

Ce diagramme de classes présente notre système dans son ensemble, et les relations entre ses différentes composantes.

Il y aura plusieurs types d'utilisateurs :

- **Administrateur** : l'administrateur a la main sur l'ensemble du système. Lorsqu'il crée un fichier, il peut autoriser un ou plusieurs autres utilisateurs à accéder à son fichier, mais étant le créateur du fichier, seul lui a le droit de le supprimer. Tout utilisateur extérieur au projet a par défaut le droit d'accéder au fichier en lecture, mais seul l'administrateur peut l'autoriser à le modifier, ou lui retirer son droit d'accès en lecture en privatisant son projet.
- **Collaborateur** : le collaborateur d'un projet n'a le droit qu'aux actions que lui offrent l'administrateur du projet. Par défaut il peut accéder en lecture à tous les projets, sauf ceux qui sont privés. Il peut aussi demander à l'administrateur d'un projet de l'y inclure.

Un administrateur a le choix entre créer un projet public et un projet privé. Il pourra toujours modifier l'accès à son projet ultérieurement, quel que soit son choix lors de la création de son projet.

Lors de la création d'un projet, un chat incluant tous les membres du projet (l'administrateur et ses collaborateurs) est créé. Cette fonction permet aux utilisateurs de communiquer et de modifier un fichier de façon simultanée.

DEVELOPPEMENT CLIENT LOURD

Le développement du client lourd s'est fait via l'utilisation d'un plug-in Eclipse appelé Window Builder. Cet outil permet un développement d'interface plus rapide dans la mesure où il met à la disposition de l'utilisateur une interface graphique permettant de créer de façon plus concrète sa fenêtre, en y posant des éléments via un *drag and drop*. De cette manière, un code basique de l'élément est automatiquement généré, que l'on peut modifier à notre façon pour obtenir le résultat souhaité.

Ainsi, nous avons développé les interfaces pour la connexion, la création de compte, et l'affichage de l'interface principale une fois un utilisateur connecté.

Les fonctionnalités de bases ont pu être implémentées, telles que la création d'un compte, et la connexion sur un compte créé. En utilisant les sockets, nous avons pu faire en sorte de créer des fichiers avec l'interface swing de l'application. Il est possible de créer, modifier et supprimer un fichier.

Malheureusement, par manque de temps nous n'avons pas pu implémenter le chat pour cette application via *Swing*.

CLIENT LEGER

ARCHITECTURE :

- **MVC**

FONCTIONNALITES :

- **Inscription**
- **Connexion**
- **Déconnexion**
- **Détection existence d'utilisateur**
- **Récupération mot de passe oubliée**

Les technologies utilisées : Ajax, JS, CSS

Pour le client léger, nous avons au début une interface pour se connecter au site ou faire l'inscription avec des servlet qui détecte si un utilisateur est déjà enregistré et s'il a oublié son mot de passe c'est possible de le récupérer grâce à la jsp ForgotPass et sa servlet Forgot_serv qui envoie un email à l'utilisateur avec son mot de passe.

- **Editeur de Texte orienté WEB**
 - **Multi-utilisateur**
L'écriture sur le fichier via l'éditeur de texte est synchronisée par un petit système d'OBSERVER implémenter en **Ajax / JS**
 - **Auto-complétions : codes HTML**
 - **Coloration syntaxique : codes HTML**
Nous avons implémenté un petit système d'automate pour faire l'auto-complétions et coloration en temps réel du code saisis (Malgré la limite du temps nous avons seulement cette fonctionnalité que pour la syntaxe HTML) avec **JS / CSS**
 - **Raccourci clavier : BOOTSTRAP**
En exploitant les événements des éléments INPUT et manipulation de l'objet DOM avec

JS / CSS

- **Numérotation lignes**
En exploitant les événements et manipulation de l'objet DOM avec **JS / CSS**

- **Aperçu en temps réel**
Nous avons utilisé l'élément iframe et les événements pour générer automatiquement un aperçu en temps réel du code HTML/JS/CSS saisis par l'utilisateur via l'éditeur de texte avec **JS / CSS**

- **Accès rapide à tous les fichiers de l'arborescence du projet**

- **Notification action des autres utilisateurs collaborateur**
Encore en implémentant un système d'observer qui nous permet de faire une notification sur tout les clients en cas de changement sur le serveur avec **Ajax / JS**

- **Gestion des projets et fichiers**
 - **Création**
 - **Suppression**
 - **Téléchargement**
 - **Importation (En cours)**

Les dossiers et fichiers du projet sont enregistrer sur le serveur et tracé dans la base de données pour faciliter leurs gestions. Nous avons privilégié l'utilisation de **Ajax** pour envoyer les requêtes aux Servlet.

Une fois l'utilisateur est connecté au site il trouve un Menu ou il peut ouvrir ses anciens projets ou créer un nouveau et utiliser le TextEditor.

- **Gestion invitations**

L'utilisateur trouve aussi une gestion des invitations pour accepter ou refuser une invitation d'un autre utilisateur pour collaborer sur un projet.

Les technologies utilisées : Ajax, JS, CSS

Ici on utilise l'ajax pour afficher la liste des invitations reçus

- **Gestion permissions**

Aussi il peut changer les permissions pour un utilisateur donné comme l'écriture ou la lecture...

Les technologies utilisées : Ajax, JS, CSS

Ici nous utilisons ajax pour remplir la liste déroulante avec les projets de l'utilisateur ou il est admin et on change de cette liste ça nous remplissons une deuxième liste déroulante avec les membres de projet enfin pour changer leurs permissions .

- **Gestion Profil**
- **Système suivis (Follow)**
- **Actualités amis (Friends News)**

Un profil, où l'utilisateur il peut modifier tous ses informations personnelles ou voir ses followers et les gens qu'il follow.

Grâce à la fonction follow on peut follow des utilisateurs pour voir leur actualité et leur historique sur les projets dans la partie

« My Friends News »

Les technologies utilisées : JS, CSS

- **Messagerie (inbox)**

Inbox, une petite boîte messagerie pour envoyer des messages privés.

- **Recherche dans le site(Search)**

Un search input dans la barre de navigation, pour chercher des utilisateurs, projets, fichiers ... avec les mots clés

Les technologies utilisées : JS, CSS

Grâce à la CSS et JS ici nous avons une belle animation de la barre de recherche.

- **Chat pour chaque projet**

Un chat en temps réel, permet les utilisateurs qui travaillent sur le même projet de discuter entre eux.

Les technologies utilisées : Ajax, JS, CSS

Grace à l'AJAX nous avons un chat en temps réel qui actualise la div de affichage de message à chaque fois que il y'a un nouveau message inséré dans la base .

DIAGRAMMES DE SEQUENCES

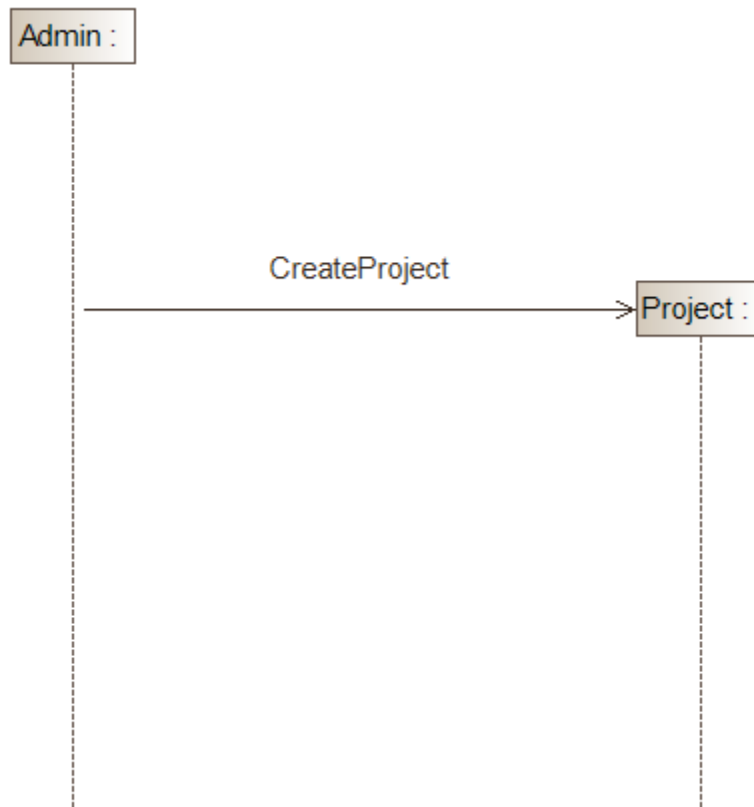


Fig 3 : Créer un projet

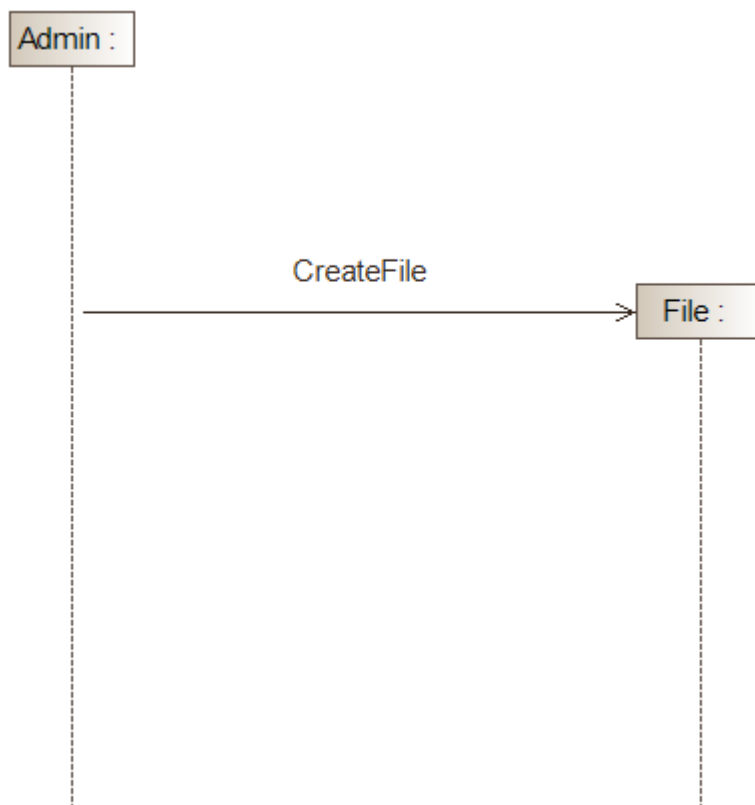


Fig 4 : Créer un fichier

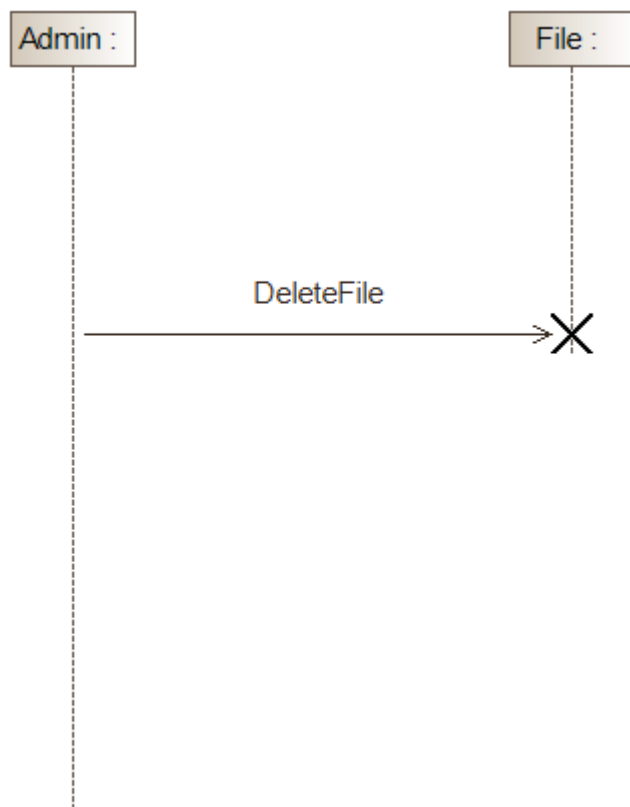


Fig 5 : Supprimer fichier

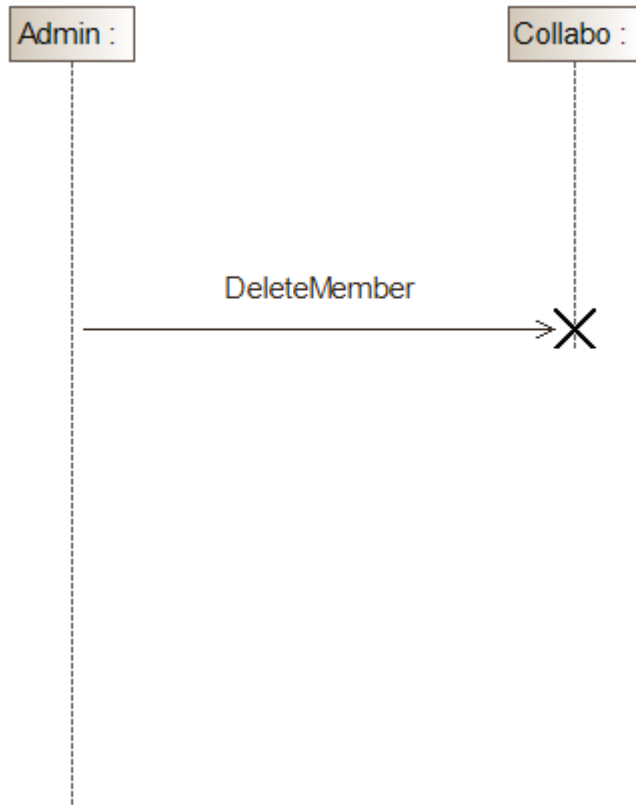


Fig 6 : Supprimer Collaborateur

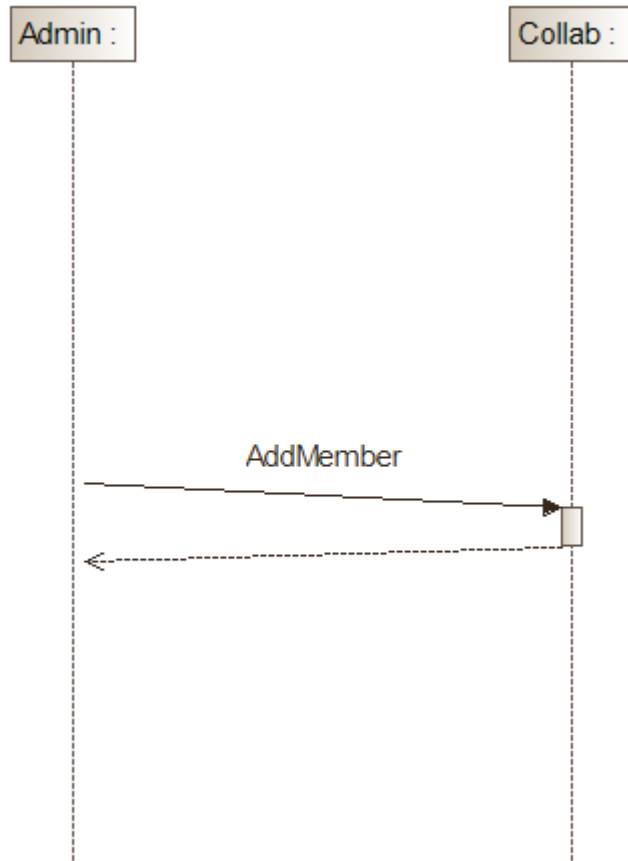


Fig 7 : Ajouter Collaborateur

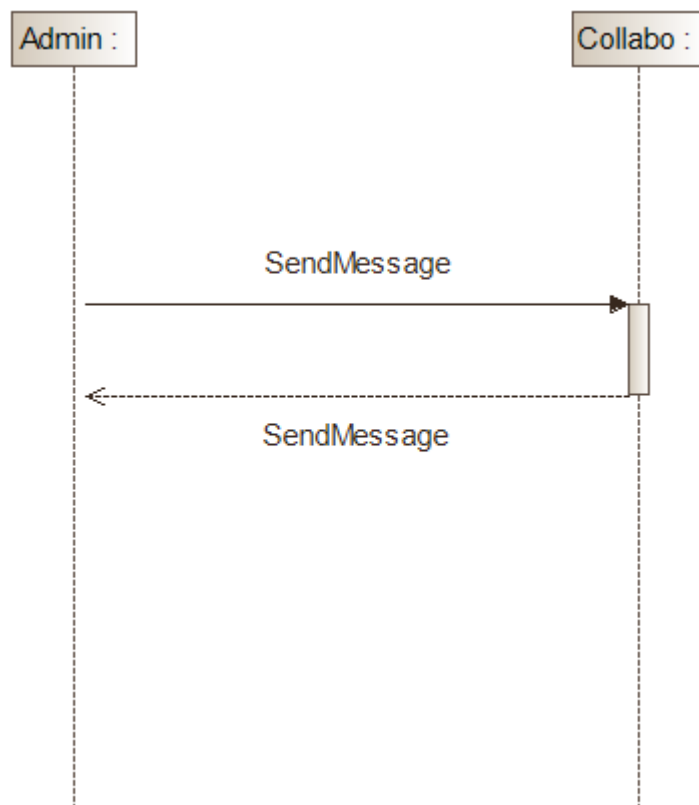


Fig 8 : Envoyer Message