

# Rapport intermédiaire du projet de développement web 2

Bastien Duraj - Carreteros Laetitia - Gentile Pierre - Didier-Roche François

19 mai 2018

# Table des matières

<b>1</b>	<b>Description de l'application</b>	<b>3</b>
<b>2</b>	<b>Diagrammes</b>	<b>3</b>
2.1	Schéma entité-association . . . . .	3
2.2	Diagramme des tables de la BDD . . . . .	3
2.3	Diagramme des cas d'utilisation . . . . .	3
2.4	Diagramme de séquence . . . . .	4
2.4.1	Scénario 1 . . . . .	5
2.4.2	Scénario 2 . . . . .	6
2.4.3	Scénario 3 . . . . .	7
2.5	Diagramme des classes . . . . .	7
<b>3</b>	<b>Application des technologies</b>	<b>8</b>
3.1	Client Léger . . . . .	8
3.2	Client Lourd . . . . .	8
<b>4</b>	<b>Origine du code</b>	<b>8</b>
<b>5</b>	<b>Distribution et temps de travail</b>	<b>9</b>
<b>6</b>	<b>Les ressources utilisées</b>	<b>10</b>
<b>7</b>	<b>Annexe</b>	<b>11</b>
7.1	Schéma entité-association . . . . .	11
7.2	Diagramme des tables de la BDD . . . . .	12
7.3	Diagramme des cas d'utilisation . . . . .	13
7.4	Diagramme de séquence 1 . . . . .	14
7.5	Diagramme de séquence 2 . . . . .	15
7.6	Diagramme de séquence 3 . . . . .	16
7.7	Diagramme des classes . . . . .	16

# 1 Description de l'application

Pour ce projet de développement web nous avons choisi de faire une application permettant de créer de manière collaborative une **carte mentale**. Elles seront composées de nœuds reliés entre eux et il existera une hiérarchisation entre ceux-ci (un nœud est fils d'un autre, ...). Il y aura également la possibilité de mettre du texte dans un nœud pour représenter l'idée associée à celui-ci. En ce qui concerne le document qui sera partagé sur le serveur il s'agira directement d'un fichier XML permettant de représenter de l'information de manière hiérarchisée et donc parfaitement adapté à notre type de document.

Pour finir avec la description globale de notre application voici une liste des différentes fonctionnalités que nous avons mise en place :

- Pour le client lourd et léger :
  1. Intégration d'un chat interne à l'application.
  2. Possibilité de se connecter à son compte.
  3. Possibilité de gérer son compte.
  4. Possibilité de créer des document.
  5. Possibilité de zoomer ou dézoomer le graphe.
- Pour les nœuds :
  1. Modification des nœuds : nom, dimensions.
  2. Ajout, suppression et déplacement d'un nœud.
  3. Possibilité de décrire un nœud pour préciser sa fonction.
- Pour la communauté :
  1. Possibilité de discuter les membre du groupe associés a un document.
  2. Possibilité de créer des groupes.

Dans la suite de ce rapport, nous verrons dans un premier temps les différents schémas que nous avons utilisée comme ligne directrice tout au long de notre projet.

## 2 Diagrammes

Pour notre application nous avons également crée 4 types de diagrammes UML (Classe, BDD, Cas d'utilisation et Séquence) avec un schéma entité-association.

**Remarque : Tous les diagrammes présentés ne sont pas définitif et sujet à modification. De plus une version en grand format de chacun des diagrammes est fournit en Annexe à la fin de ce rapport.**

### 2.1 Schéma entité-association

Ce schéma représente les relations entre les différents objets de notre application, il s'agit du premier schéma que nous avons établi ce qui a permis par la suite de mettre en place une base de données claire et complète.

### 2.2 Diagramme des tables de la BDD

La base de donnée représenté ici nous permettra de stocker les différentes informations relative à un utilisateur ou à un groupe de travail avec les messages envoyés entre les différents utilisateurs.

### 2.3 Diagramme des cas d'utilisation

Ce schéma est une représentation simple et claire des différentes possibilités auxquelles un type d'utilisateur peut accéder. Nous avons dans ce schéma 'hiérarchisé' les différents acteurs suivant leur importance dans le document, le chef de groupe étant au-dessus (en terme de choix d'action) des autres

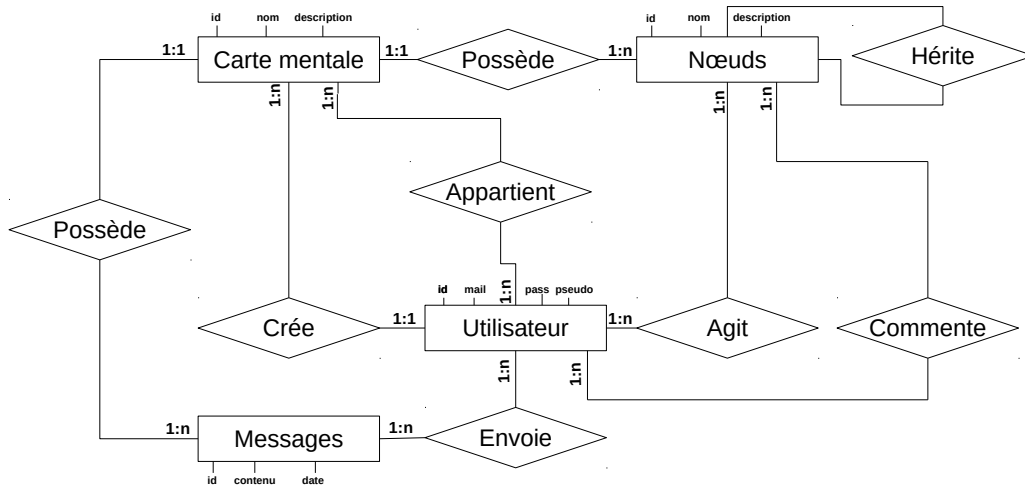


FIGURE 1 – Schéma entité-association de l'application

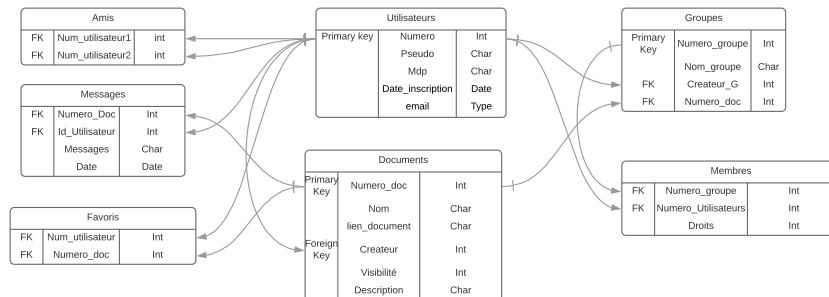


FIGURE 2 – Tables et relation entre les données de la BDD

acteurs.

**Remarque :** Dans un souci de lisibilité des flèches n'ont pas été montré, le chef du groupe peut très bien faire les actions d'un membre du groupe ou d'un spectateur, ect... .

## 2.4 Diagramme de séquence

Nous avons divisé notre diagramme de séquence en 3 parties, un par scénario, où nous mettons en scène à chaque fois un type d'utilisateur : le créateur du document, un membre du groupe de travail et un simple spectateur.

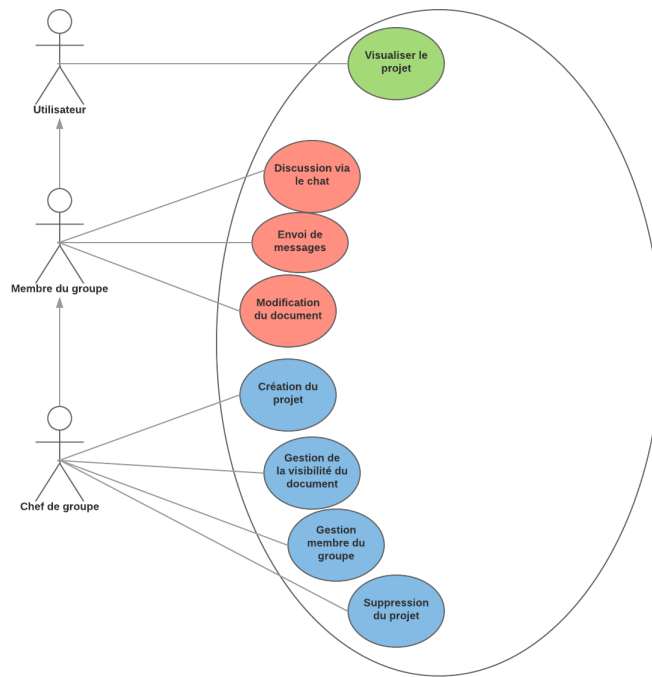


FIGURE 3 – Représente les différentes actions possibles pour chaque type d'utilisateur

#### 2.4.1 Scénario 1

Le premier scénario met en scène l'utilisateur créant un document, on y voit les interactions entre le serveur, le client et la base de données.

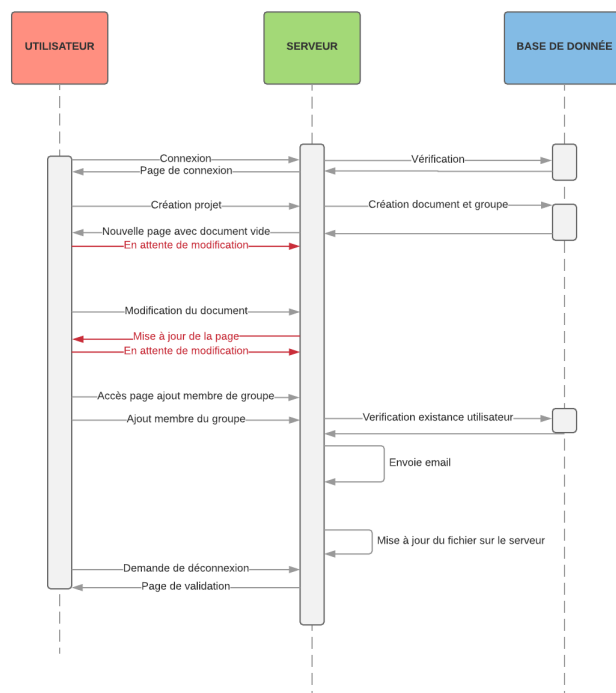


FIGURE 4 – Représente le déroulement d'un scénario entre trois acteurs : Utilisateur, serveur, BDD

### 2.4.2 Scénario 2

Dans ce scénario on retrouve 2 utilisateurs qui discutent via le 'tchatche' interne à l'application et unique à chaque projet, et via la messagerie qui concerne toute l'application.

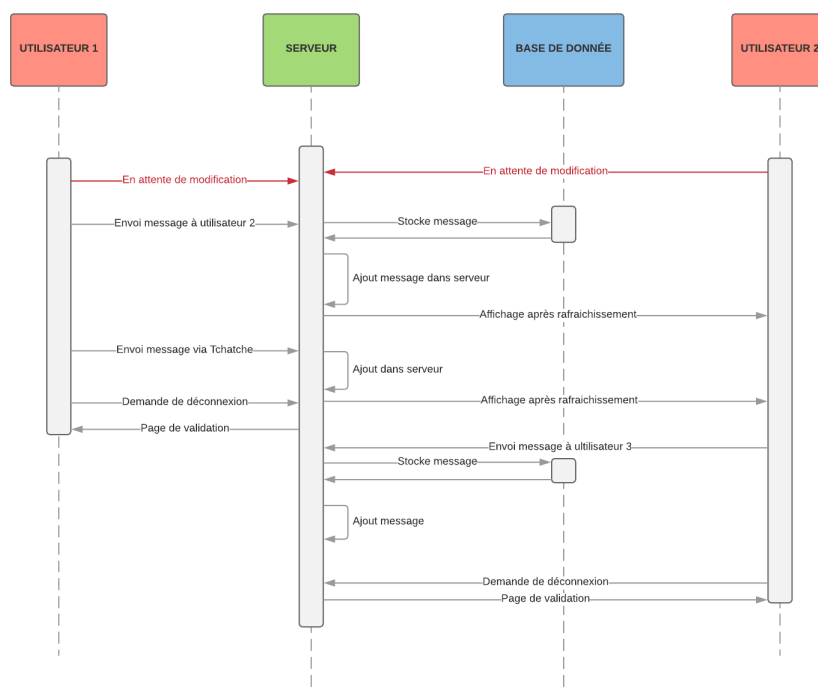


FIGURE 5 – Représente le déroulement d'un scénario entre quatre acteurs : Utilisateur, serveur, BDD et un autre utilisateur

### 2.4.3 Scénario 3

Le troisième scénario met en scène un spectateur qui fait une demande d'inscription puis demande la visualisation d'un document.

DIAGRAMME\_SEQUENCE3

Bastien Duraj | April 13, 2018

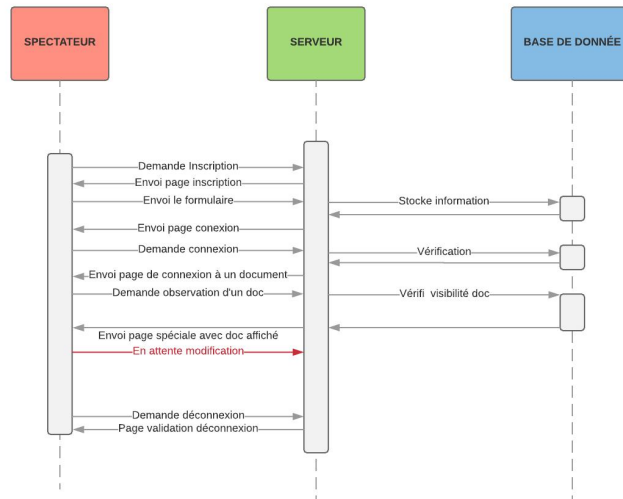


FIGURE 6 – Représente le déroulement d'un scénario entre trois acteurs : Spectateur, serveur, BDD

### 2.5 Diagramme des classes

Pour finir nous présentons ici **un premier jet** de notre diagramme des classes.

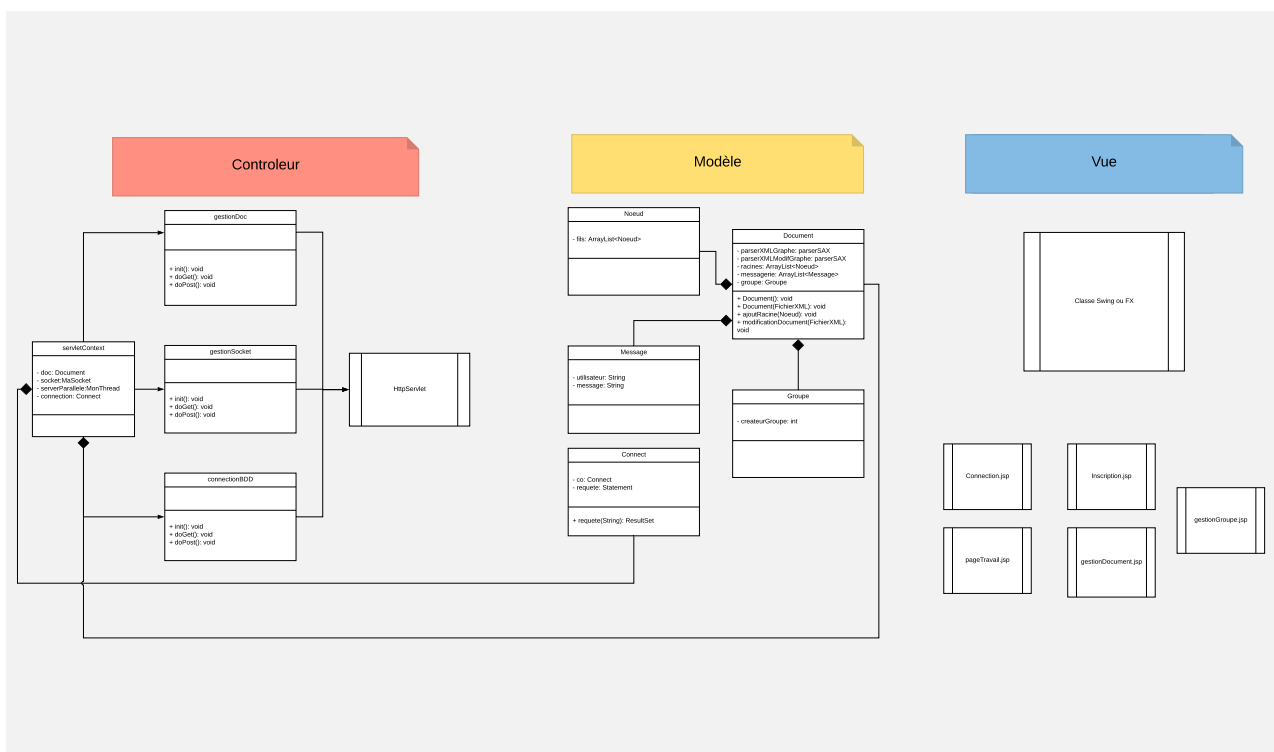


FIGURE 7 – Diagramme des classes de notre application divisé en 3 parties suivant l'architecture MVC pour plus de visibilité

## 3 Application des technologies

Avant de passer à la distribution du travail entre les différents membres de notre groupe nous allons revenir de manière plus précise sur l'utilisation des différentes technologies dans notre application.

### 3.1 Client Léger

- **Utilisation des Servlets**

Elles permettent de contrôler les accès aux données qu'ils s'agissent de la BDD ou des données représentées en mémoire comme notre document. Lorsqu'un client envoie une requête faisant intervenir ces données, la servlet interne à la JSP fait appel à une servlet spéciale faisant office de contrôleur. Une servlet spéciale servira également à gérer la socket (récupération des données) permettant de faire le lien entre la partie serveur et le client lourd.

- **Utilisation des JSPs**

Les JSP dans notre application servent à représenter les différentes pages du client léger mais également à communiquer avec les contrôleurs. Elles communiqueront avec le serveur grâce à des formulaires.

- **Utilisation de la librairie JDBC**

Elle nous permet de créer une communication entre le serveur et la base de données et de soumettre nos différentes requêtes.

### 3.2 Client Lourd

- **Utilisation de la librairie JDBC**

Elle nous permet de créer une communication entre le serveur et la base de données et de soumettre nos différentes requêtes.

- **Utilisation des fichiers XML**

Les fichiers XML servent à représenter notre graphe pour et de pouvoir le modifier grâce à un parseur DOM.

- **Utilisation de java FX**

La bibliothèque Java FX nous a permis de créer les interfaces de connexion, inscription, gestion de compte pour le client lourd. Elle nous a aussi permis de représenter le document partagé par un fichier XML. En effet, la représentation des vues avec JavaFX utilise des fichiers XML représentant les éléments affichés à l'écran. Dans la zone de représentation du document, les noeuds et leurs sont donc représentés par un fichier XML.

- **Utilisation des Sockets**

Les sockets nous ont permis de gérer la communication entre le client lourd et la partie serveur. Lorsque l'utilisateur apporte certaines modifications à un fichier XML (représentation du graphe) l'information est transmise au serveur afin de faire les modifications sur ce graphe. Cette transmission d'information permet aussi de vérifier la connexion d'un utilisateur à partir du client lourd, d'envoyer les données nécessaires lors de l'inscription d'un utilisateur ou de la modification de son compte.

## 4 Origine du code

Le code est majoritairement 100% personnel. Certaines parties du code sont faites avec l'appui de tutoriel. Les parties copiées telles qu'elles sont indiquées dans le code source.



## 5 Distribution et temps de travail

Pour conclure ce rapport intermédiaire nous parlerons de la distribution du travail entre les différents membres du groupe :

- Pierre Gentile a majoritairement fait la partie JSP/Servlet et plus précisément le divers menus du client léger. (15-25h de travail).
  - Bastien Duraj s'est occupé de la partie base de données et sockets. Il a aussi développé certaines JSP et Servlets que l'on retrouve dans le client léger.
  - François Didier-Roche s'est occupé de développer la totalité du client lourd.
  - Laetitia Carreteros s'est occupée de la partie mise en forme du client léger. Elle a aussi participé au développement du client léger.
- Chef d'équipe :** Duraj Bastien.

## 6 Les ressources utilisées

- Utilisation de git et github pour la gestion de sources.
- Utilisation de JDBC pour la partie base de données.
- Utilisation de la JSTL pour les JSP.
- [https ://openclassrooms.com/](https://openclassrooms.com/)
- [https ://stackoverflow.com/](https://stackoverflow.com/)

## 7 Annexe

### 7.1 Schéma entité-association

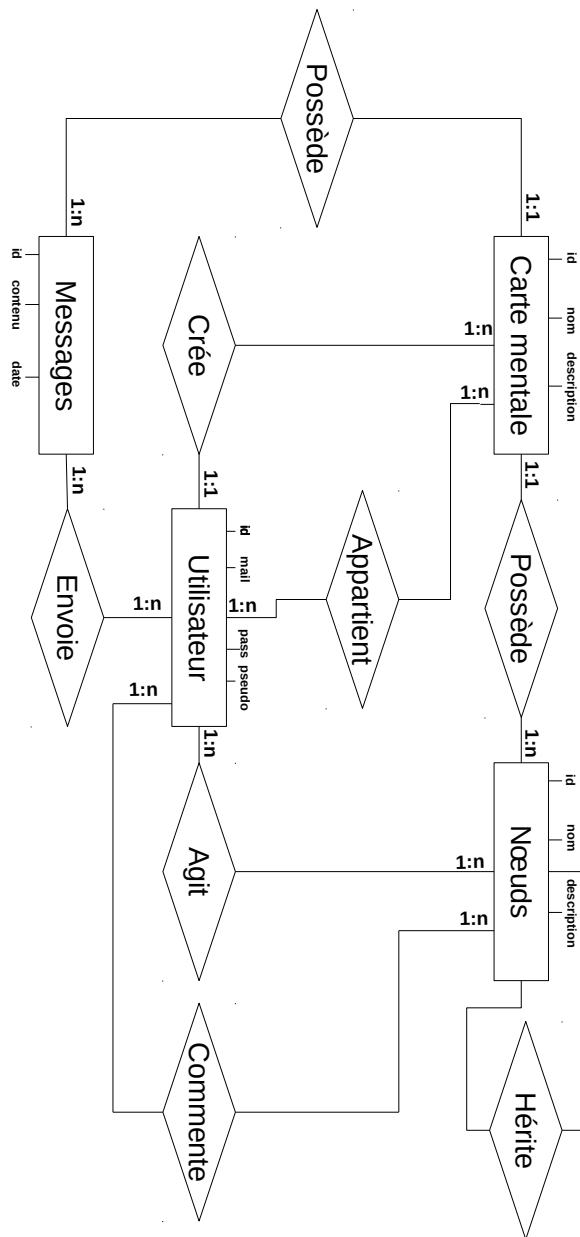


FIGURE 8 – Schéma entité-association de l'application

## 7.2 Diagramme des tables de la BDD

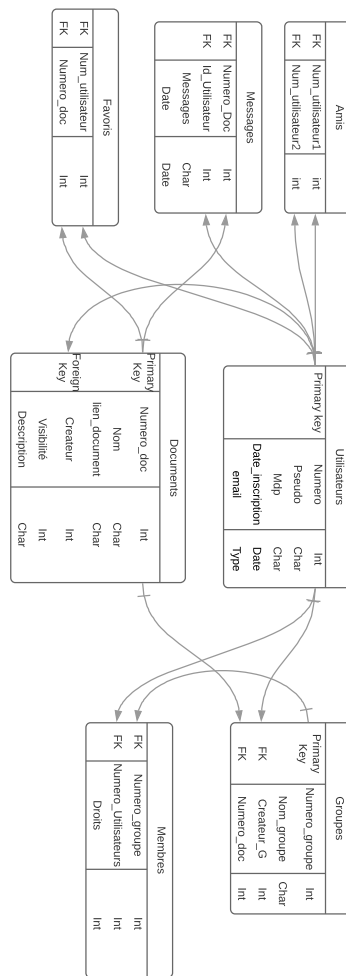


FIGURE 9 – Tables et relation entre les données de la BDD

### 7.3 Diagramme des cas d'utilisation

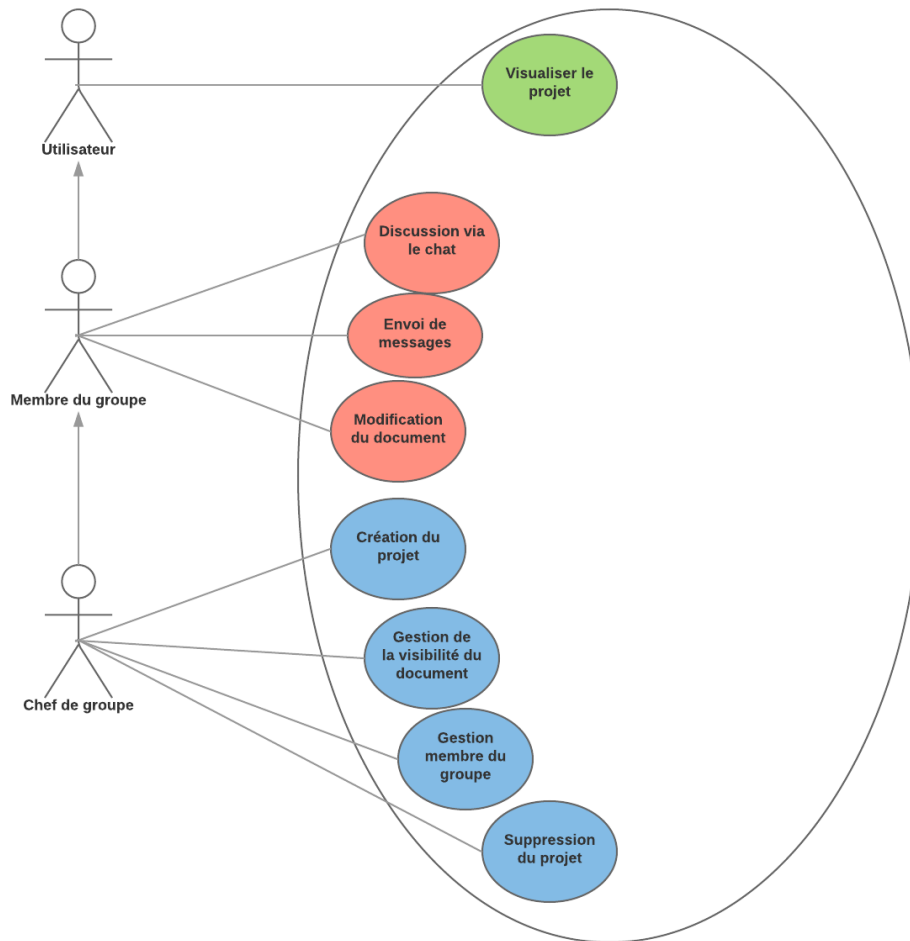


FIGURE 10 – Représente les différentes actions possibles pour chaque type d'utilisateur

## 7.4 Diagramme de séquence 1

### DIAGRAMME\_SEQUENCE1

malyasse | April 13, 20

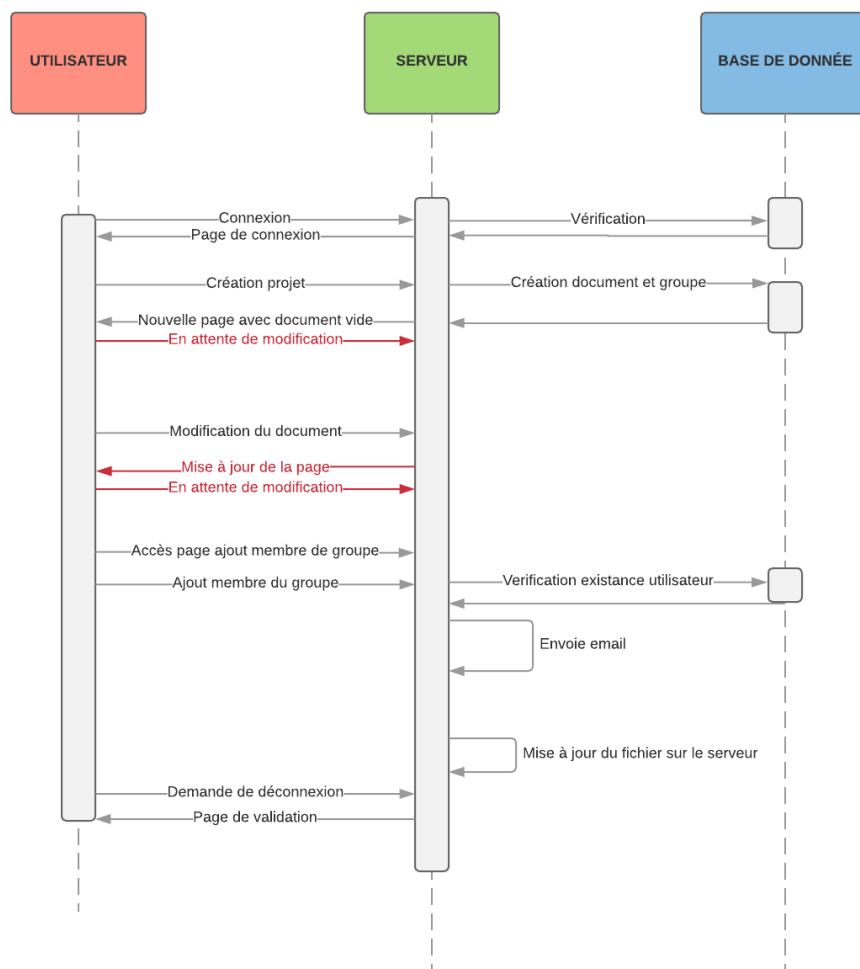


FIGURE 11 – Représente le déroulement d'un scénario entre trois acteurs : Utilisateur, serveur, BDD

## 7.5 Diagramme de séquence 2

DIAGRAMME\_SEQUENCE2

Bastien Duraj | April 13, 2018

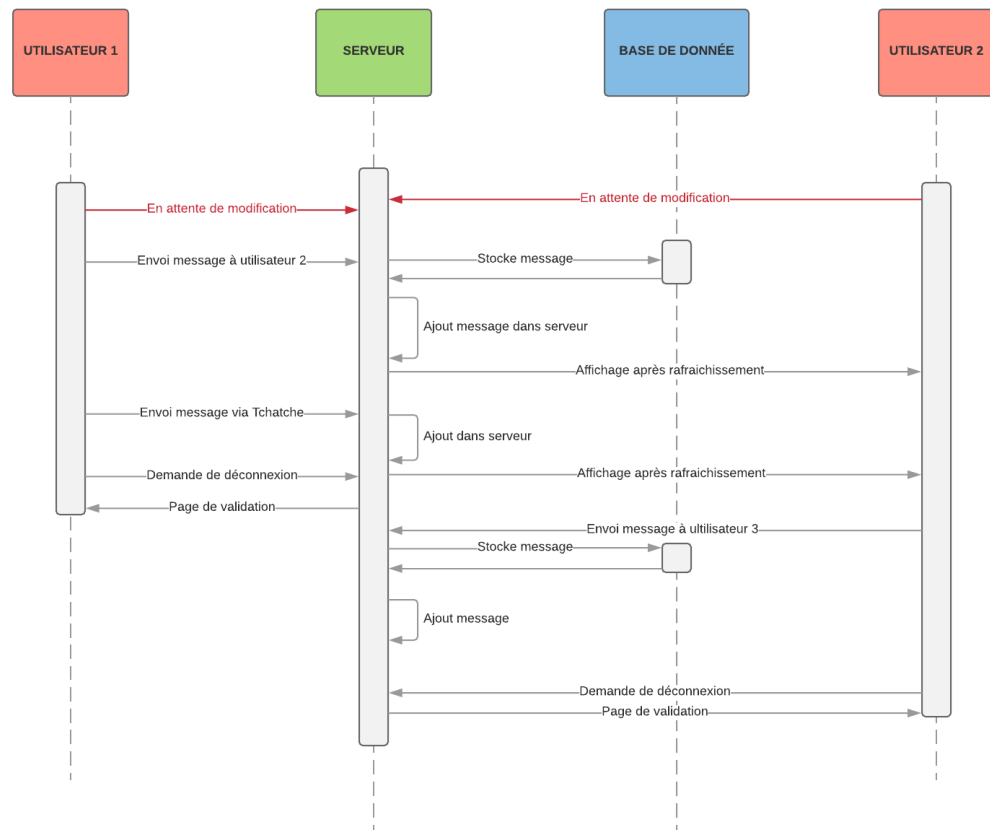


FIGURE 12 – Représente le déroulement d'un scénario entre quatre acteurs : Utilisateur, serveur, BDD et un autre utilisateur

## 7.6 Diagramme de séquence 3

### DIAGRAMME\_SEQUENCE3

Bastien Duraj | April

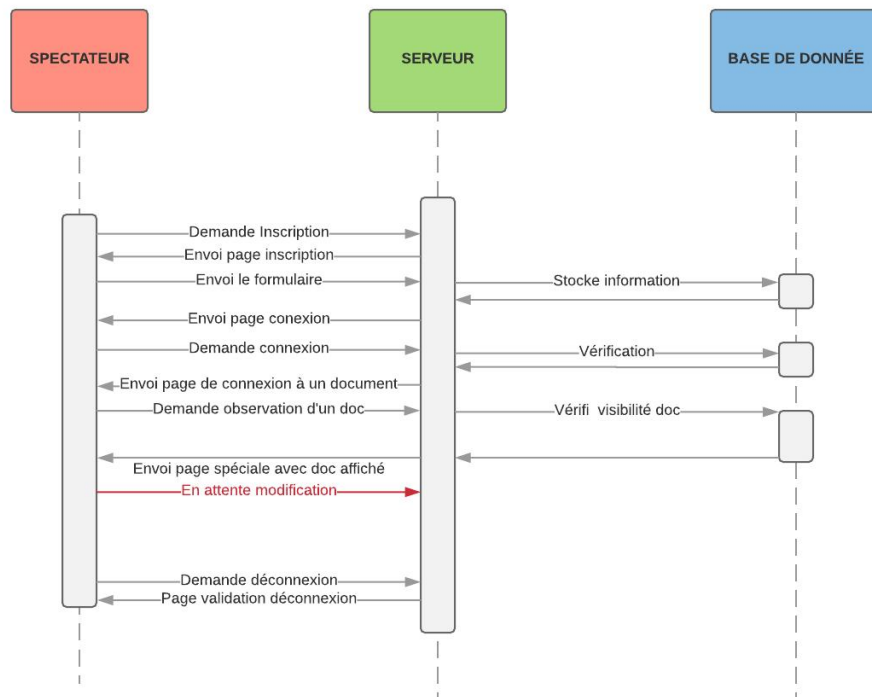


FIGURE 13 – Représente le déroulement d'un scénario entre trois acteurs : Spectateur, serveur, BDD

## 7.7 Diagramme des classes



