Structure de nos Base de données :

Nous avons 4 base de données dont 3 gérée avec MySQL qui sont Users, Friends et Login et une gérée avec MongoDB qui est celle des messages.

Users :

```
CREATE TABLE IF NOT EXISTS users (\
    login VARCHAR(256) NOT NULL PRIMARY KEY, \
    password VARCHAR(256) NOT NULL, \
    lastname VARCHAR(256) NOT NULL, \
    firstname VARCHAR(256) NOT NULL);
```

Friends :

Ce système n'est pas vraiment un système d'amitié car en effet une amitié va dans les deux sens or ici on a plus un système de follow.
Nous voulions ajouter une option pour bloquer (avec l'attribut flag les utilisateur indésirable seulement on a jugé que ce n'était pas très important face au travail demandé et que l'on implémenter ça une fois le cœur du projet terminé.

```
"CREATE TABLE IF NOT EXISTS friends ( \
            from_user INTEGER NOT NULL, \
            to_user INTEGER NOT NULL, \
            since TIMESTAMPS NOT NULL, \
            PRIMARY KEY (from_user, to_user), \
            FOREIGN KEY (from_user) REFERENCES users(rowid), \
            FOREIGN KEY (to_user) REFERENCES users(rowid) )"
            //flag VARCHAR(256) NOT NULL, \
```

Login :

Cette base nous permet de garder une trace sur les utilisateur connecté c'est avec celle-ci que l'on gère les multiples connection. On stocke dedans les login des utilisateurs actuellement connecté ainsi que le token de leurs session en cours, une fois l'utilisateur déconnecté sont entrée correspondant est supprimer de la base de donnée.

```
CREATE TABLE IF NOT EXISTS login (\
        login VARCHAR(256) NOT NULL PRIMARY KEY, \
        token VARCHAR(256) NOT NULL);
    )
```

Messages :

Cette base est gérée avec MongoDB avec chaque entrée de cette base est un document JSON ayant un champ émetteur, destinataire, le message en question, la date de création du message. On a aussi un champ listLike qui est une liste des user ayant liké le message, on a implémenté ceci pour pouvoir stocker les utilisateur ayant déjà liké le post pour éviter d'avoir une situation oú l'utilisateur peut like a l'infinie

```
        const doc = {
            from: from_user,
            to: to_user,
            msg: message,
            date: Date.now(),
            like: 0,
            listComment : [],
            listLike : []
        };
```

On a implémenté un champ listComment qui est une liste des commentaires posté sous le message. Ces commentaires sont eux aussi des documents JSON contenant un champ émetteur, le commentaire en question, la date ainsi que les likes et une liste des utilisateurs ayant liké le commentaire.

```
        const document = {
            from: login,
            msg: message,
            date: Date.now(),
            like: 0,
            listLike : []
        };
```