

Document Explicatif

Binôme: CHITER-ZENINE

Répartition des tâches:

CHITER	ZENINE
<ul style="list-style-type: none">• Composant de connexion.• Composants d'inscription.• Composant d'accueil.• Composant du mot de passe oublié.• Composant Feed (Fil d'actualité).• Composant SearchBar.• Composant NavBar.• Composant de suggestions.• Composant ListFriends.• Composant Friends.• Composant Comments.• Composant ListComments.• API Friends (Tous les services).	<ul style="list-style-type: none">• Composant Profil.• Composant Twist.• Composant ListTwists.• Composant NewTwist.• Composant SideBar.• Composant ListUser.• Composant User.• Composants des dialogs (ProfilDialog, ModifyCommentDialog...)• API Messages (Tous les services).• API Users (Tous les services).• Services Login et Logout.

Remarque: On a codé le projet ensemble, donc il n'y avait pas vraiment une vraie partition de tâches, aussi, dans plusieurs fonctionnalités, chacun de nous a codé une solution et on a choisi la meilleur de solutions.

Problèmes rencontrés:

Lors de l'élaboration du projet, on a eu quelques difficultés:

- Le CSS: C'est un langage qui demande beaucoup de pratique cependant on a pas eu vraiment le temps de pratiquer sur le CSS.
- La gestion des sessions: Au départ, on avait mal compris ou utiliser les sessions (On avait mal démarré).
- Suppression d'un commentaire: Quand on supprimait un commentaire il fallait supprimer sa référence dans la liste commentaires du message.

Ce qui reste à faire:

- Ajouter une photo de profil pour l'utilisateur.
- Rechercher des utilisateurs.
- Supprimer son compte.
- Système de notifications.
- Améliorer le système d'amitiés (Invitations).
- Modification du profil (Nom, Prénom...)
- ...

Choix de modélisation:

- On a utilisé Mongoose comme client MongoDB sur expressJs car il permettait de créer des modèles et facilitait la communication avec la base de données.
- On a utilisé une bibliothèque de composants React pour nous éviter le CSS qui demandait du temps pour le maîtriser.