**Projet Qt**

On souhaite réaliser un système de chat fonctionnant avec une architecture client / serveur.

Contraintes techniques :

* Les utilisateurs devront s’identifier avec un couple nom de compte / mot de passe.
* Les comptes utilisateurs devront être stockés dans une base de données.
* La base de données devra être sécurisée (les mots de passe ne doivent pas être en clair).
* Il devra être possible d’envoyer et recevoir les messages textuels du chat depuis une application C++ et une page web (HTML5/JS). Vous devrez donc mettre en place à la fois un serveur TCP mais également un serveur WebSocket et diffuser les messages à tous les clients connectés.
* Lorsqu’un utilisateur envoi un message, son pseudo est associé au message (on doit pouvoir identifier de qui vient le message).
* Lorsqu’un utilisateur se connecte il doit pouvoir voir les 100 derniers messages avant sa connexion.
* Les comptes utilisateurs doivent pouvoir être créés depuis l’application cliente et la page web. C’est le serveur C++ qui envoi les données en base.
* Vous utiliserez la librairie Qt pour les applications réalisées en C++.

Contraintes organisationnelles :

* Vous travaillerez par équipe de 3 ou 4. Les groupes vous seront communiqués.
* Un dépôt Github (privé) sera utilisé pour la gestion des sources. Vous partagerez ce dépôt avec votre professeur.

Livrables attendus :

* Analyse UML :
  + Diagramme des cas d’utilisation.
  + Diagramme d’exigences.
  + Diagramme de classe.
  + Diagramme de déploiement.
  + MCD (Modèle conceptuel de données = Structure de la BDD).
* Applications finales :
  + Une application cliente C++ (TCP).
  + Une application cliente HTML5/JS (Web Socket).
  + Une application serveur C++ (TCP & Web Socket).
* Serveur :
  + Un serveur de base de données SQL ou NoSQL (à vous de choisir) sous Linux.
* Rapport de projet
* Bonus :
  + Une application mobile iOS et/ou Android cliente C++ (TCP).
  + Le serveur C++ tourne en console.
  + Le serveur C++ est installé sur une machine sous Linux.
  + Il est possible d’envoyer des images en + des messages textuels.
  + L’authentification est chiffrée (login et mot de passe ne sont pas en clair sur le réseau lors de l’authentification).

Exemple de répartition des tâches possible à quatre étudiants :

* E1 : Serveur TCP & Web Socket
* E2 : Client TCP C++
* E3 : Client web (HTML5/JS)
* E4 : Fonctions BDD

D’autres répartitions des tâches sont possibles mais chaque étudiant doit maîtriser au moins une des facettes techniques du projet (serveur TCP, client TCP, web sockets, hachage de mot de passe, cryptographie, base de données).

La répartition des tâches devra apparaitre clairement dans votre rapport de projet.

Deadlines :

* Début du projet : 14/09/2021.
* Analyse à rendre pour le 17/09/2021.
* Version finale à rendre / présenter pour le 28/09/2021.
* Rapport de projet à rendre pour le 01/10/2021.