



## École Polytechnique de Montréal Département Génie Informatique et Génie Logiciel

### INF3405 – Réseaux Informatiques

#### TP1 : Projet en réseaux informatiques Système de clavardage interactif

#### 1. Informations générales

Session	Hiver 2024
Public cible	Étudiants de 1 <sup>er</sup> cycle
Date et Lieu de réalisation	À partir du 8 janvier 2024
Taille de l'équipe	3 étudiants
Pondération	15 %
Date de remise du projet	<b>Remise pour tous les groupes le 16 février 2024 (23h55 au plus tard)</b>
Directives particulières	<ul style="list-style-type: none"><li>✓ Tout rapport sera pénalisé de <b>5 points</b> s'il est soumis par une équipe dont la taille est différente de trois (3) étudiants sans l'approbation préalable du chargé de laboratoire.</li><li>✓ Soumission par <b>moodle</b> uniquement (<a href="http://moodle.polymtl.ca">http://moodle.polymtl.ca</a>).</li><li>✓ <b><u>Soumission d'une archive compressée (.zip / .rar) contenant le rapport (en format PDF), les projets Eclipse contenant le client et le serveur et les exécutable .jar</u></b></li><li>✓ Toute soumission de l'archive en retard est pénalisée à raison de 5 points par heure de retard.</li></ul>
Chargé de laboratoire	Groupe 01/02/06 - Mehdi Kadi ( <a href="mailto:mehdi.kadi@polymtl.ca">mehdi.kadi@polymtl.ca</a> ) Groupe 03/04/05 - Bilal Itani ( <a href="mailto:bilal.itani@polymtl.ca">bilal.itani@polymtl.ca</a> )
Auteurs :	Mehdi Kadi, Bilal Itani
Révision	Mehdi Kadi, Bilal Itani

#### 2. Connaissances requises

- Sockets
- Threads
- Programmation **Java uniquement**

### 3. Objectifs du laboratoire

L'objectif de ce laboratoire est de familiariser l'étudiant :

- aux échanges Client/Serveur en utilisant les sockets;
- au développement « d'applications réseau » en utilisant les threads.

Ce travail pratique consiste, par la même occasion, à évaluer deux des 12 qualités de l'ingénieur définies par le BCAPG (Bureau canadien d'agrément des programmes de génie). Le Bureau d'agrément a pour mandat d'attester que les futurs ingénieurs ont atteint ces 12 qualités à un niveau acceptable. Les deux qualités en question sont:

**Qualité 4 (Conception)** : capacité de concevoir des solutions à des problèmes d'ingénierie complexes et évolutifs et de concevoir des systèmes, des composants ou des processus qui répondent aux besoins spécifiés, tout en tenant compte des risques pour la santé et la sécurité publiques, des aspects législatifs et réglementaires, ainsi que des incidences économiques, environnementales, culturelle et sociales.

**Qualité 7 (Communication)** : habileté à communiquer efficacement des concepts d'ingénierie complexes, au sein de la profession et au public en général, notamment lire, rédiger, parler et écouter, comprendre et rédiger de façon efficace des rapports et de la documentation pour la conception, ainsi qu'à énoncer des directives claires et y donner suite.

### 4. Description

#### Mise en contexte

Vous en avez marre de vous faire espionner par les géants des réseaux sociaux. À chaque fois que vous discutez de quelque chose avec vos amis vous êtes ciblé par des publicités douteuses. Les géants de l'infonuagique essaient de vous soutirer de l'argent de vos petites poches pour vous vendre encore plus d'objets inutiles. La vie d'étudiant étant rude, et sachant que vous n'avez pas d'argent à donner à ces compagnies vous ciblant, vous choisissez de vous révolter. Étant étudiant à Polytechnique Montréal et surtout, expert en réseau informatique avant même d'avoir fini le cours, vous choisissez de développer votre propre application client-serveur permettant de discuter avec vos amis. Votre grand-mère, une fidèle supportrice, vous encourage dans votre quête et vous fait don de son bon vieux Pentium 3. Vous choisissez de l'utiliser comme serveur de clavardage.

Pour l'instant, vous prévoyez qu'une simple **interface console**, car vous voulez assurer uniquement le fonctionnement de votre application client-serveur et une interface graphique n'est pas une de vos priorités.

## Requis :

Vous aurez à faire 2 produits, un serveur, et un client pour communiquer avec le serveur. Au démarrage du serveur, celui-ci demande à l'utilisateur d'entrer les informations suivantes : adresse IP du poste sur lequel s'exécute le serveur, port d'écoute (**un port entre 5000 et 5050 uniquement**). Une vérification doit être faite pour s'assurer que l'utilisateur a bien entré des données cohérentes. Par exemple, le serveur doit détecter que l'adresse 777.-202.666.888 et le port -9999 sont incohérents. Il devra aussi s'assurer que l'adresse IP entrée est bien sur quatre octets.

Au lancement du client, celui-ci demande à l'utilisateur d'entrer l'adresse IP du serveur, le port du serveur (**entre 5000 et 5050**), un nom d'utilisateur, et un mot de passe. Le client doit vérifier la validité du format de l'adresse IP et du port entrés par l'utilisateur (de la même façon que le fait le serveur) et afficher une erreur en cas d'invalidité. Ces informations sont utilisées pour tenter de se connecter au serveur.

Le serveur vérifie alors la correspondance nom d'utilisateur / mot de passe, et accepte la connexion si le nom d'utilisateur et le mot de passe est valide. Sinon, il rejette la connexion du client, et le client affiche « Erreur dans la saisie du mot de passe ». Si le nom d'utilisateur n'existe pas dans la base de données locale du serveur, le serveur va créer automatiquement un compte pour l'usager et y assigne le mot de passe. Si la connexion a été acceptée, le serveur connecte l'utilisateur à la salle de clavardage puis lui envoie les 15 derniers messages (**donc les 15 plus récents**), ou moins s'il y en a moins que 15. Le client peut ensuite envoyer et recevoir des messages (**taille ne dépassant pas les 200 caractères**) en temps réel avec les autres utilisateurs connectés au serveur.

Quand le serveur affiche un message de la part d'un client, il doit l'afficher dans le format suivant: [Nom d'utilisateur - Adresse IP : Port client - Date et Heure (min, sec)]: Message. Voir l'exemple d'affichage attendu plus bas.

Notez que le programme client et le programme serveur ne sont (en théorie) pas exécutés sur la même machine et que, bien évidemment, plusieurs clients peuvent tenter de se connecter au serveur en même temps. Cependant, chaque client connaît d'avance l'adresse IP du serveur ainsi que le numéro de port écouté. Durant la session de clavardage, le serveur affiche et stocke dans un fichier les messages reçus, ainsi que les informations mentionnées précédemment: adresse IP, port utilisé (celui de l'application client) et réponse. Le mécanisme de stockage des données de la conversation est laissé à votre discrétion, tant que le stockage est fait adéquatement et sans qu'il n'y a d'incohérence. Lors de l'affichage des messages, on s'attend à une sortie comme suit:

[Utilisateur 1 - 132.207.29.107:46202 - 2017-10-13@13:02:01]: Salut Utilisateur 2 !

[Utilisateur 2 - 132.207.29.117:37608 - 2017-10-13@13:03:24]: Yo Utilisateur 1 !

Le serveur doit aussi pouvoir stocker la base de données (dans un fichier dans le format de votre choix (e.g .csv/.txt/.json/.xml ou autre...)) des correspondances nom d'utilisateur et mot de passe.

## 5. Résumé des requis fonctionnels

### Qualité évaluée :

#### 4.3 Procéder à la conception

*Critère d'évaluation : Intégrer les concepts de programmation en réseautique retenus au premier laboratoire en répondant aux besoins et en respectant les requis fonctionnels du projet courant.*

Les fonctionnalités attendues sont indiquées comme suit :

- Serveur
  - Saisie des paramètres du serveur (adresse IP, port d'écoute entre 5000 et 5050)
  - Pouvoir connecter des utilisateurs avec leurs mots de passe
  - Pouvoir envoyer un historique des sessions de clavardage
  - Pouvoir fermer le serveur puis le rouvrir, et avoir tous les profils usagers et messages disponibles (écriture sur disque et pas uniquement sur RAM)
  - Recevoir les messages des clients
  - Tenir un historique de tous les messages.
  - Tenir une base de données des usagers et leurs mots de passe
  - Afficher en temps réel les messages
  - Effectuer correctement la vérification nom d'utilisateur/ mot de passe
  - Créer les comptes automatiquement s'ils n'existent pas
- Client
  - Saisir au clavier l'adresse IP du serveur et le port sur lequel le clavardage se déroule.
  - Vérifier la validité de l'adresse IP saisie (uniquement le format) et le numéro de port (entre 5000 et 5050)
  - Se connecter au serveur.
  - Réceptionner les messages ou l'erreur de mot de passe.
  - Saisir une réponse (200 caractères maximum).
  - Transmettre la réponse au serveur.
  - Se déconnecter.

## 6. Livrable

### 1. Langages et bibliothèques autorisés

- Le client et le serveur doivent être développés uniquement en Java.
- Usage permis de librairies externes comme JSON.simple ou GSON. Si vous utilisez ces librairies, veuillez justifier dans la section « Présentation » de votre rapport la raison de leur utilisation.

Tout non-respect de ces consignes entraînera la note de 0.

### 2. Soumission

Le livrable est une archive (ZIP ou RAR) ayant le format suivant:

INF3405\_TP1\_matriculeX\_matriculeY où  $\text{matriculeX} > \text{matriculeY}$

Votre archive contiendra les fichiers suivants :

- Les projets Eclipse du client et du serveur incluant les fichiers sources (.java), autrement dit le dossier en entier contenant votre projet
- Le rapport au format PDF
- Les fichiers exécutables de votre client et votre serveur (.jar)

**\*\*Assurez-vous que les livrables compilent et s'exécutent adéquatement sur les ordinateurs du laboratoire ! \*\***

### 3. Rapport

Qualité évaluée :

#### 7.1 Lire et rédiger de la documentation

*Critère d'évaluation : Rédiger un rapport technique documentant efficacement le travail d'ingénierie réalisé dans ce projet en utilisant différentes formes de langage (naturel, informatique, etc.)*

Le rapport, d'une longueur maximale de 3 pages (excluant la page de présentation), doit comporter les éléments suivants :

- **Page présentation** qui doit contenir le nom ou le logo de l'école, le libellé et l'identifiant du cours, la session, le numéro et l'identification du projet, la date de remise, les matricules et noms des membres de l'équipe, la mention « Soumis à : **nom et prénom du chargé de laboratoire** ».
- **Introduction** en vos propres mots pour mettre en évidence le contexte et les objectifs du TP.

- **Présentation** de vos travaux. Une explication de votre solution mettant en lumière la prise en compte des principaux requis du système. Si vous utilisez des configurations particulières des bibliothèques ou des projets, précisez-les également.
- **Difficultés rencontrées** lors de l'élaboration du TP et les éventuelles solutions apportées.
- **Critiques et améliorations** : Il serait intéressant d'inclure vos suggestions pour améliorer le laboratoire.
- **Conclusion** : Expliquez en quoi ce laboratoire vous a été utile, ce que vous avez appris, si vos attentes ont été comblées, etc.

## 7. Évaluation

Évaluation de l'exécutable	<b>8</b>
Évaluation de l'implémentation : gestion adéquate des variables et de toute ressource (création, utilisation, libération), gestion des erreurs, logique de développement, <b>documentation du code</b> , etc.	<b>6</b>
Rapport	<b>6</b>
<b>Total des points</b>	<b>20</b>