

M18 Master ingénieur en informatique

UE: INFORMATIQUE : Programmation réseau et sécurité internet

Projet transversal : Centre cinématographique

Version 1 du 12/09/2023

2023-2024

Samuel Hiard

Patrick Quettier

1. Introduction

Ce document explicatif a pour vocation de décrire le projet de programmation que vous devrez réaliser pendant le quadrimestre. Il est peut être vu comme un « cahier des charges » décrivant le contexte, l'objectif, les fonctionnalités, les modalités, ce qui est attendu, autorisé, et interdit. Il peut être soumis à changement.

2. Contexte

Un centre cinématographique souhaite informatiser sa gestion. Le cinéma ayant un réel succès, il devient difficile de répondre efficacement à la demande d'achats de tickets sans l'aide d'un outil informatique, ce pourquoi ils font appel à vous. Grâce à votre système, les utilisateurs pourront se connecter sur le site web du cinéma, commander leur place puis recevoir une confirmation par e-mail.

3. Fonctionnalités souhaitées

Nous allons bien évidemment simplifier à l'extrême et ne modéliser qu'une partie de ce beau projet

Le système informatique est constitué de plusieurs parties :

- Un serveur HTTPS, qui permet à un client de commander une ou plusieurs places pour une séance. Le design du site est laissé à votre appréciation.
- Un serveur mail. Ce serveur sera capable d'envoyer des mails via le protocole SMTP.
- Un serveur C multi-thread qui maintient à jour, dans un fichier, la liste des séances ainsi que le nombre de places libres restantes
- Un client C permettant de rajouter une séance et de préciser le nombre de place de cette séance.

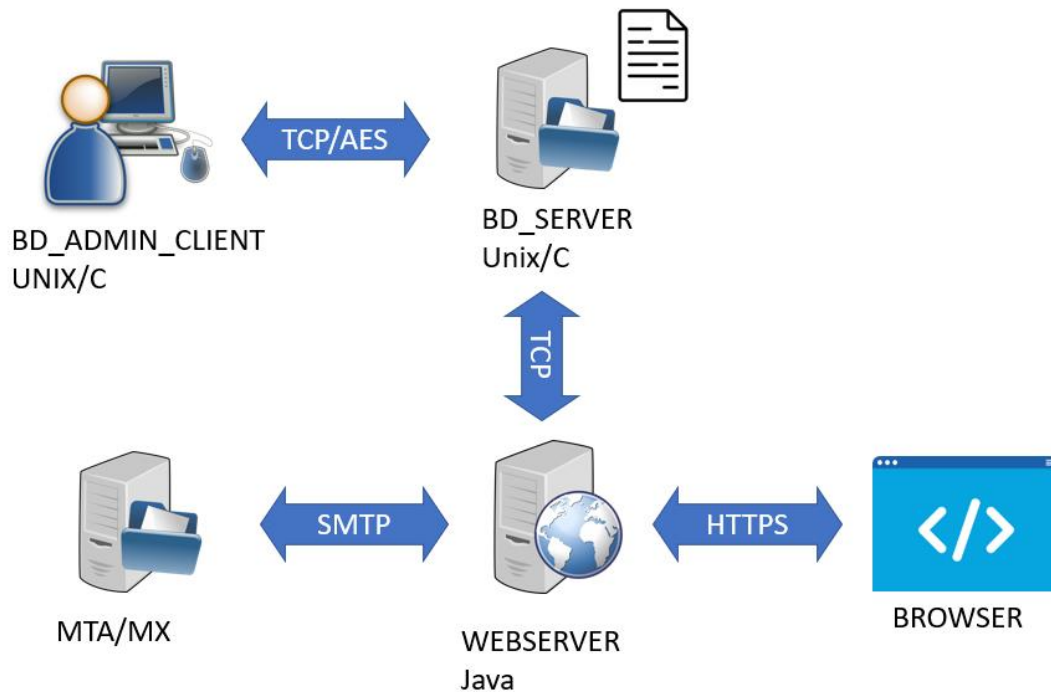
Le scénario typique est le suivant :

1. Le client (navigateur) se connecte au serveur HTTPS pour acheter une ou plusieurs places, il choisit donc la séance parmi une liste et sélectionne le nombre de places souhaitées. Il fournit également son adresse mail.
2. Le serveur HTTPS se connecte au serveur C et lui transmet cette demande
3. Le serveur C vérifie, par rapport au fichier, que la vente est possible (ou non) et renvoie une réponse au serveur HTTPS. Il modifie le cas échéant le nombre de places disponibles.
4. Le serveur HTTPS renvoie la réponse au client (navigateur).
5. En cas de succès, un mail de confirmation est envoyé à l'adresse donnée

Et entre le client C et le serveur C

1. Le client C se connecte au serveur C. Il fournit un login et un mot de passe
2. Le serveur C compare le login et le mot de passe avec les informations qu'il possède (de préférence avec un fichier properties)
3. Une fois l'accès autorisé, le client peut encoder une séance assorti d'un nombre de places disponibles. Il peut répéter cette opération autant que nécessaire
4. Lorsqu'il le désire, le client indique au serveur qu'il va se déconnecter, mettant fin à l'échange.

Schématiquement :



[Cette photo](#) par Auteur inconnu est soumise à la licence [CC BY-NC-ND](#)
[Cette photo](#) par Auteur inconnu est soumise à la licence [CC BY-NC](#)
[Cette photo](#) par Auteur inconnu est soumise à la licence [CC BY-NC](#)

Simplifications autorisées :

- Le serveur mail n'est pas obligé de se connecter réellement à un vrai MTA (port 25 souvent bloqué par FAI). Vous pouvez écrire un petit serveur qui agit comme un MTA (tout en respectant le protocole SMTP).

Au niveau de la sécurité :

- Le client C chiffrera ses échanges avec le serveur C via le protocole AES en mode ECB.
- Le client C devra s'authentifier au serveur C en lui fournissant un login et un mot de passe. Le mot de passe sera stocké haché/salé sur le serveur.
- Le webserveur java possèdera un certificat auto-signé
- L'envoi de mail ne sera pas sécurisé
- Le serveur Java utilisera un pool de threads afin de mitiger l'attaque DDoS et ne permettra que 200 connections en simultané
- Bonus : l'échange entre le serveur Java et le serveur C pourra être sécurisé

4. Consignes supplémentaires

Ce travail est à réaliser seul.

Il compte pour 50% de l'UE, moyenne arithmétique

Vous ferez une présentation de ce travail pendant la session d'examens. Vous ferez une petite démonstration, puis nous regarderons le code source, le tout formant une discussion informelle, les concepts théoriques pouvant être abordés également. Une petite modification du code source en direct peut être demandée.

Le code source devra être envoyé au plus tard ce même jour, par mail à samuel.hiard@hepl.be. L'archive .7z, .zip ou .rar doit bien préciser votre nom, votre section (M18) ainsi que la mention « ProgResSecProj ».

Amusez-vous bien !