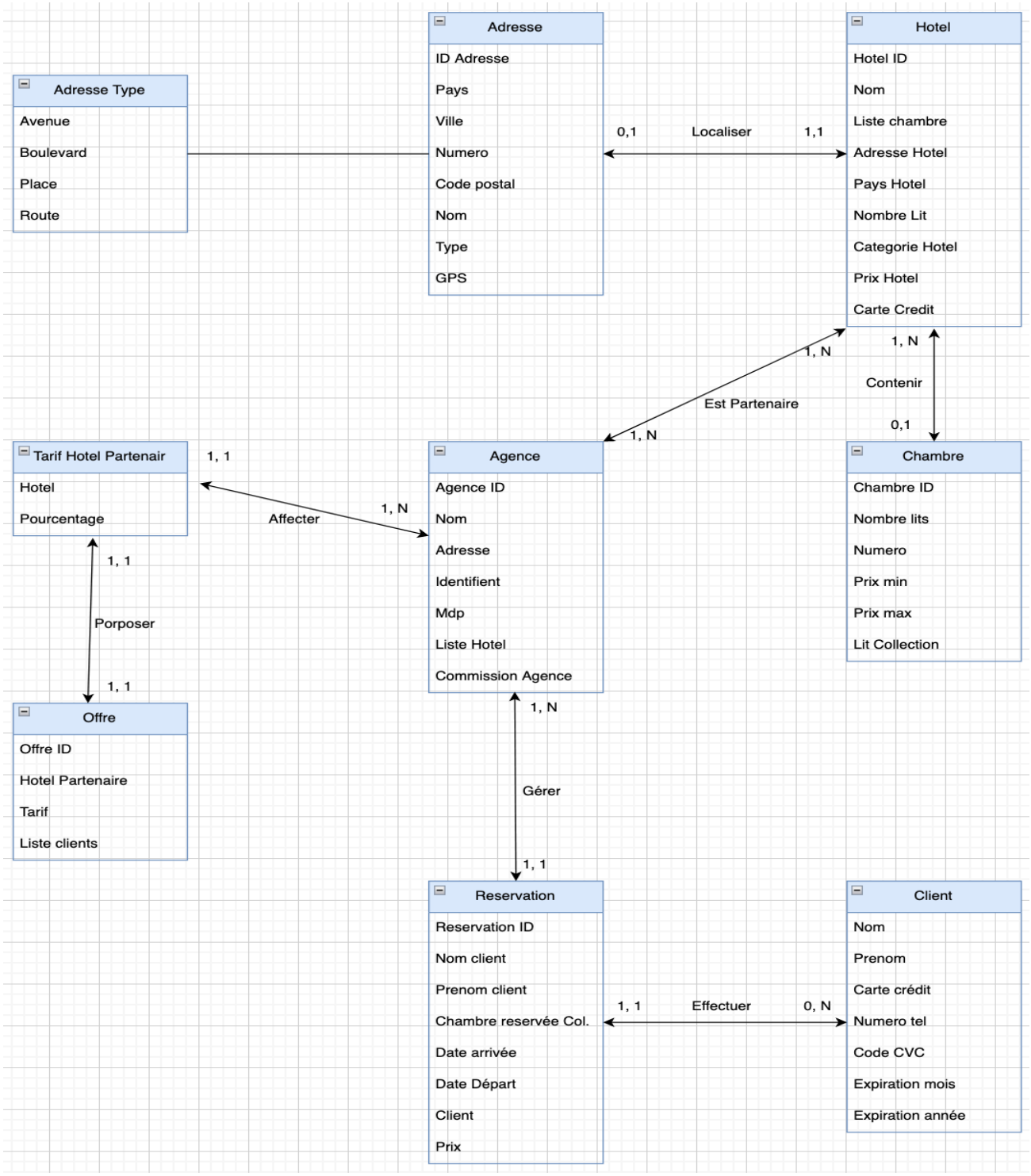
Rapport : TP SOAP

Étudiants : Shaibi Youssef & Izouka Safae

Schéma UML :



Objectifs :
Ce TP sur les services web SOAP a pour but de nous faire comprendre les services web en C .NET avec des exemples Exercices et outils. S'ensuit la mise en place d'exemples concrets sur le thème de la gestion hôtelière.

SOAP :
SOAP est un protocole de communication basé sur XML pour permettre aux applications de s'échanger des informations via HTTP. Il permet ainsi l'accès aux services web et l'interopérabilité des applications à travers le web.
Nous avons suivi ce tutoriel :
https://www.youtube.com/playlist?list=PLd2zFiP7vZos0iluc-J3CY4SopYQ28oQ-

L'idée principale de ce programme est la suivante :
Ce programme est divisé en trois projets distincts : *TPQ1*, *TPQ2IdArtifact* et *TPQ2ClientAgence*.

I) TPQ1 : Un programme de réservation d'hôtel utilisant une interface de ligne de commande (CLI) et des collections en java. L'application de réservation d'hôtel permettra aux clients de trouver et de réserver une chambre d'hôtel en fonction de la disponibilité des chambres. Dans ce projet, nous avons démontré la capacité de concevoir des classes à l'aide de la POO, d'organiser et de traiter des données avec des collections et d'utiliser des types Java courants.

II) TPQ2IdArtifact : Un projet Java qui publie et consomme un service Web JAX-WS basé sur SOAP à l'aide de dépendances Maven.

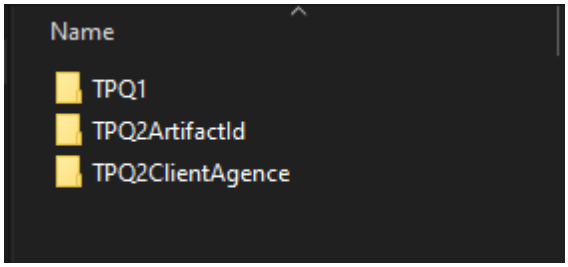
III) TPQ2ClientAgence : Un logiciel d'agence de voyage qui utilise une interface de ligne de commande (CLI). Cette application permettra aux agences de voyages de s'inscrire, de se connecter, de faire des réservations d'hôtel et d'accéder aux données des hôtels via leurs propres services Web.

⇒ Les technologies utilisées sont :

- Java : https://www.java.com/en
- JDK : https://www.oracle.com/java/technologies/downloads
- SOAP : https://www.javatpoint.com/soap-web-services
- Maven Repository : https://mvnrepository.com
- Eclipse : https://www.eclipse.org
- Postman API Platform : https://www.postman.com

⇒ Comment exécutez-vous les programmes sur Eclipse ?
Lors de la décompression du dossier ZIP sous le nom Java_SOAP_Web_service, vous trouverez trois dossiers distincts : *TPQ1*, *TPQ2IdArtifact* et *TPQ2ClientAgence*.

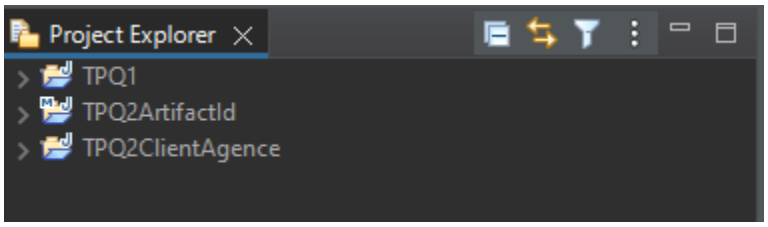
Voir l'image ci-dessous :



Après cela, ouvrez “Eclipse”, choisissez le dossier “Java_SOAP_Web_service” comme espace de travail puis sur le côté droit vous trouverez “Package Explorer” Cliquez sur “Import” ⇒ “General” ⇒ “Existing Project into Workspace” ⇒ “Next” et choisissez le premier dossier de projet sous le nom *TPQ1*. Faites la même chose pour *TPQ2ClientAgence*.

Concernant le projet *TPQ2IdArtifact* Cliquez sur “Import” ⇒ “Maven” ⇒ “Existing Maven Projects” ⇒ “Next” et choisissez le dossier du projet sous le nom *TPQ2IdArtifact*.

Le résultat attendu est dans l'image ci-dessous :



Vérifions maintenant le JRE System Library dans Eclipse IDE:

- Sélectionnez le premier projet (*TPQ1*), faites un clic droit et sélectionnez “Properties” ⇒ “Java Build Path” ⇒ “Libraries” puis cliquez sur “Edit” ⇒ “Workspace default JRE” ⇒ “Finish” ⇒ “Apply and Close”.
- Sélectionnez le deuxième projet (*TPQ2IdArtifact*), faites un clic droit et sélectionnez “Properties” ⇒ “Java Build Path” ⇒ “Libraries” puis cliquez sur “Edit” ⇒ “Excursion environment (JavaSE-1.8)” ⇒ “Finish” ⇒ “Apply and Close”.
- Répétez : Sélectionnez le deuxième projet (*TPQ2IdArtifact*), faites un clic droit et sélectionnez “Properties” ⇒ “Java Build Path” ⇒ “Libraries” puis cliquez sur “Edit” ⇒ “Workspace default JRE” ⇒ “Finish” ⇒ “Apply and Close”.
- Sélectionnez le troisième projet (*TPQ2ClientAgence*), faites un clic droit et sélectionnez “Properties” ⇒ “Java Build Path” ⇒ “Libraries” puis cliquez sur “Edit” ⇒ “Workspace default JRE” ⇒ “Finish” ⇒ “Apply and Close”.

⇒ Comment exécuter les projets sur le serveur ?

I) TPQ1 :

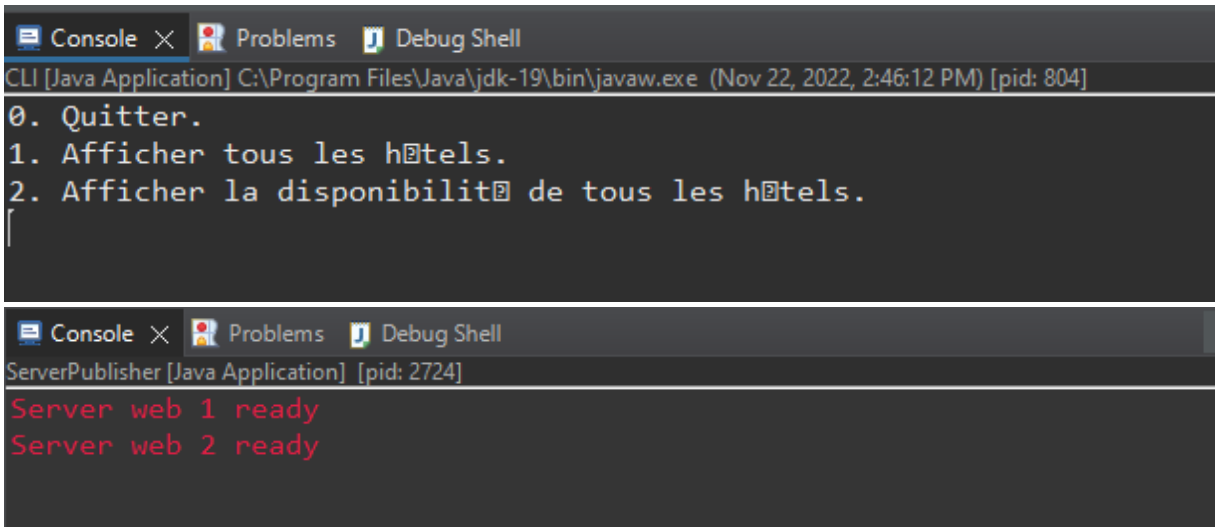
Aller à “src/Q1/CLI.java” et exécuter le serveur. Le résultat attendu est dans l'image ci-dessous :

II) TPQ2IdArtifact:

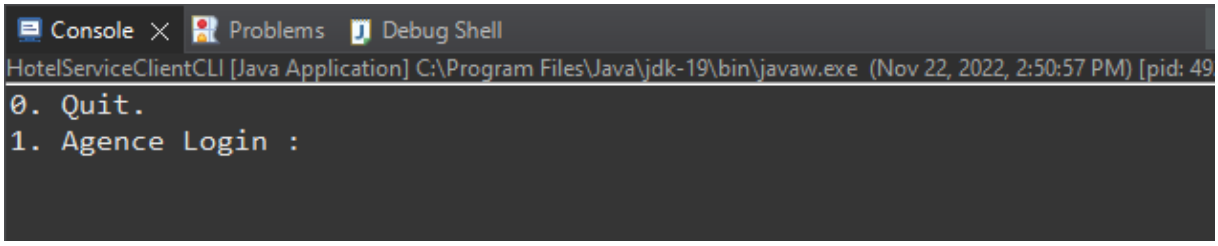
Aller à “src/main/java/serverPublisher/serverPublisher.java” et exécuter le serveur.

Le résultat attendu est dans l'image ci-dessous :

III) TPQ2ClientAgence : Aller à “src/main/HotelServiceClientCLI.java” et exécuter le serveur.



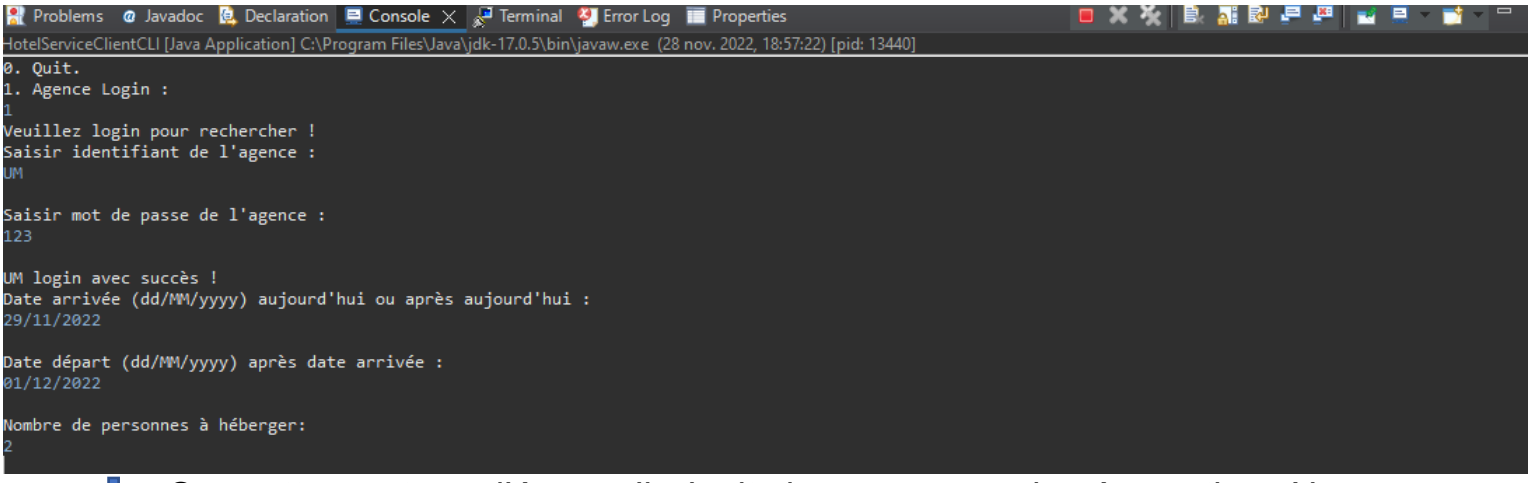
Le résultat attendu est dans l'image ci-dessous :



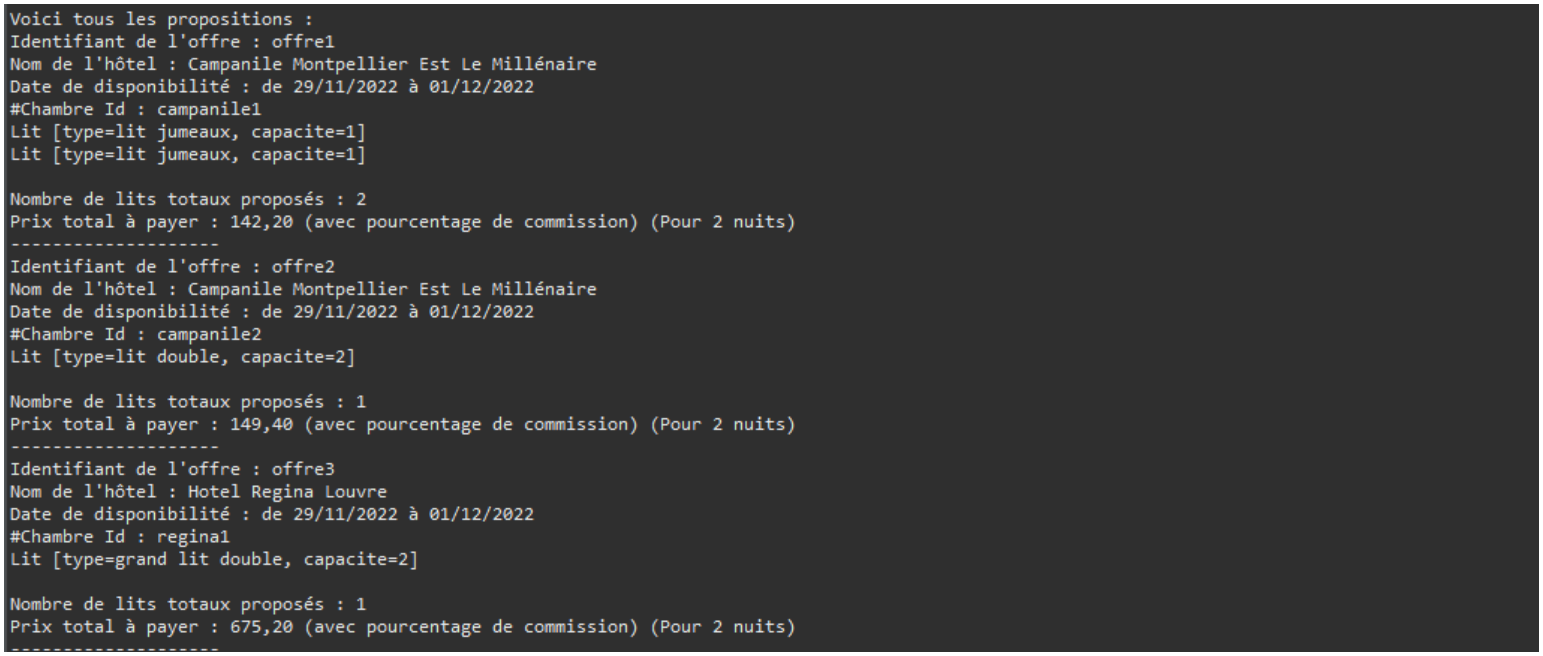
Pour s'identifier à l'agence :

- Identifiant : UM
- Mot de passe : 123

Nous avons utilisé des images qui se téléchargent sur l'ordinateur de l'utilisateur, suivant le path indiqué dans l'application de l'agence. Cette fonctionnalité ne marchera pas si le path n'est pas modifié. L'application permet de faire une réservation parmi la liste d'offre proposées par l'agence dans les hôtels partenaires.



Sur cette capture d'écran, Il s'agit du processus de réservation. Nous pouvons voir qu'il est d'abord nécessaire de se connecter à l'aide de ses identifiants, puis de rentrer les dates de réservations voulues ainsi que le nombre de personnes.



Ensuite le programme propose au client les « hotels », « chambres » et « lits » disponibles et correspondant à sa recherche. Le prix lui est également indiqué.

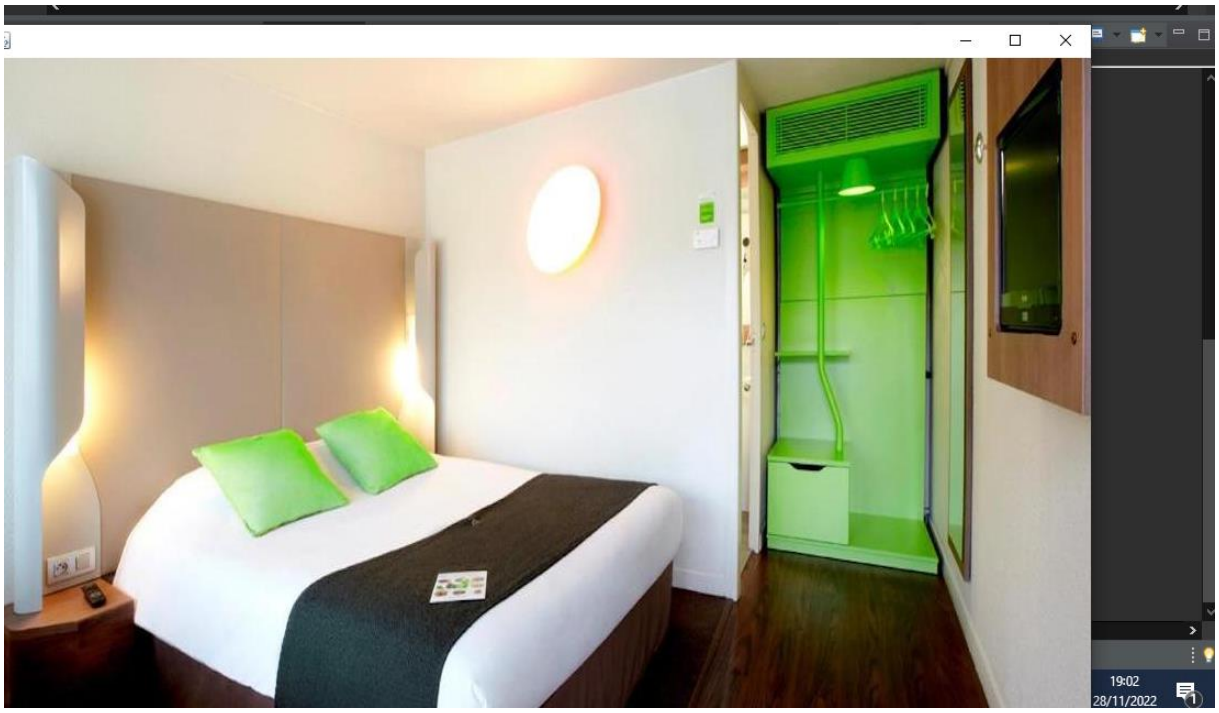
```
--------------------
Display image de chambre ? y/n


Désolé, mauvaise saisie. Veuillez réessayer.
Display image de chambre ? y/n
y

Chambre Id ?
 regina1
Chambre Id n'est pas correct.
Display image de chambre ? y/n
y

Chambre Id ?
campanile2
Display image de chambre ? y/n
```

Après cela, le programme propose au client s'il veut voir la phot de la chambre. S'il entre « y » pour « oui », alors le programme lui demande quel identifiant de l'offre préfère-t-il réserver. Une fois l'ID rentré dans le programme, la photo de la chambre choisie s'affiche (voir ci-dessous).



REMARQUE :
Assurez-vous d'exécuter "src/main/java/serverPublisher/serverPublisher.java" puis exécutez "src/main/HotelServiceClientCLI.java" car ils communiquent ensemble.