Projet base de données et réseau

Gestion d'entrée automatisée pour les salles de sport via des cartes RFID

Réalisé par :

Gouarab Ali

Boukais ines

Année universitaire 2024/2025

Table des matières :

Table des Figures :

Introduction

Introduction

1. *Contexte du projet :*

L'objectif d'un système de gestion d'entrée automatisée dans une salle de sport via des cartes RFID est de faciliter l'accès des membres et de gérer les abonnements.

Les cartes RFID permettent un accès sans contact et rapide. Un membre n'a qu'à approcher sa carte du lecteur pour entrer.

Les cartes RFID sont sécurisées, empêchant les utilisateurs non autorisés d'entrer. Le système peut vérifier l'identité du membre en temps réel, par exemple restreindre l'entrée en cas de non-paiement ou de problème lié à l'abonnement.

2. *Planning :*

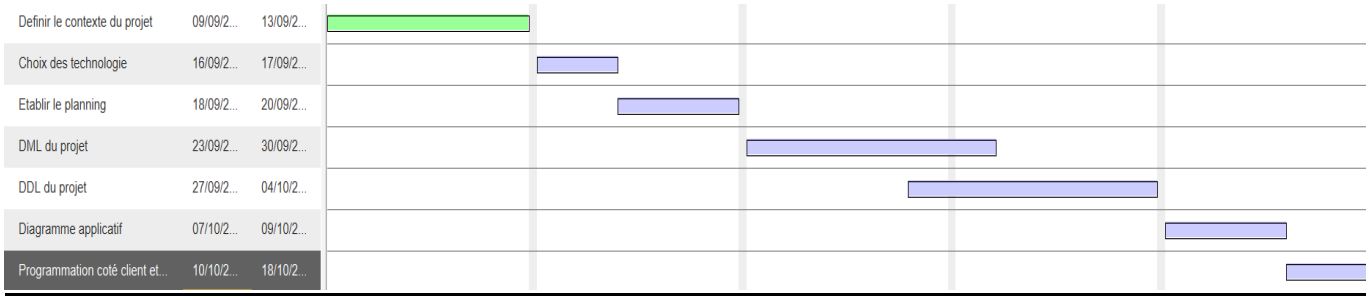

Figure 1 : Diagramme de Gantt.

Analyse et conception

Analyse et conception

1. *Dictionnaire des données :*

Utilisateur	Type	NULL / NOT NULL	Domaine	Remarque
Id_utilisateur	INT	NOT NULL	Numérique	PK
nom	VARCHAR(50)	NOT NULL	Alphabétique	
prenom	VARCHAR(50)	NOT NULL	Alphabétique	
pseudo	VARCHAR(50)	NULL	Alphanumérique	Unique
email	VARCHAR(50)	NOT NULL	Alphanumérique	Unique
mot_de_passe	VARCHAR(50)	NOT NULL	Alphanumérique	

Carte	Type	NULL / NOT NULL	Domaine	Remarque
Id_carte	INT	NOT NULL	Numérique	PK
date_creation	DATE	NOT NULL	Numérique	

Abonnement	Type	NULL / NOT NULL	Domaine	Remarque
Id_abonnement	INT	NOT NULL	Numérique	PK
type_abonnement	ENUM	NOT NULL	Alphabétique	{Basic, Premium}
date_debut	DATE	NOT NULL	Numérique	
date_fin	DATE	NOT NULL	Numérique	

Salle	Type	NULL / NOT NULL	Domaine	Remarque
Id_salle	INT	NOT NULL	Numérique	PK
capacite	INT	NOT NULL	Numérique	capacite>0

Ville	Type	NULL / NOT NULL	Domaine	Remarque
Code_postal	INT	NOT NULL	Numérique	PK
Nom	VARCHAR(30)	NOT NULL	Alphabétique	
Rue	VARCHAR(50)	NOT NULL	Alphanumérique	

2. Modèle conceptuel des données :

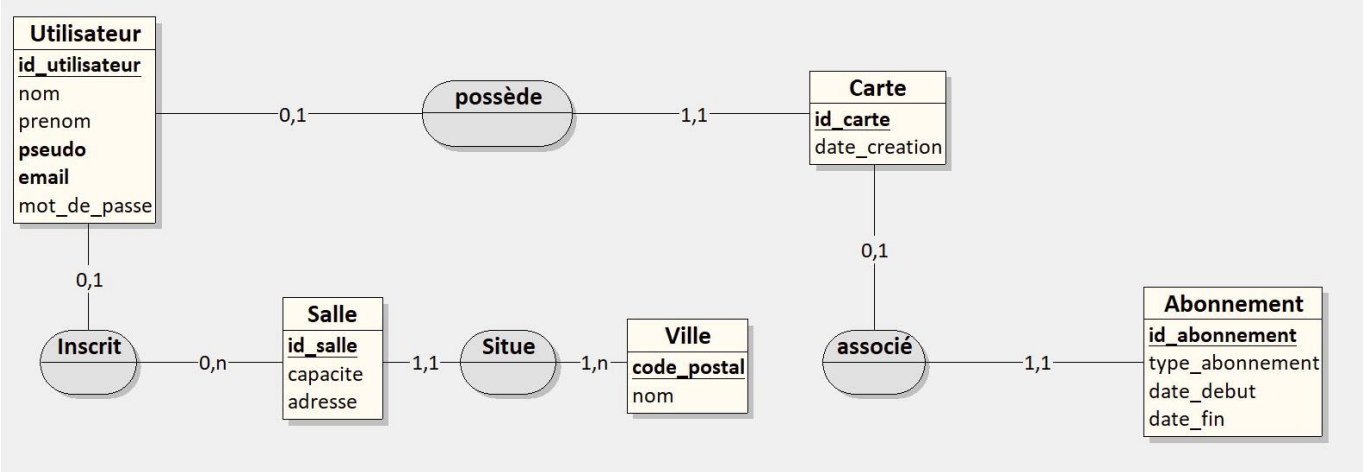

Figure 2 : Diagramme E/A.

3. Modèle logique des données :

Ville = (**code_postal**, nom, rue)

Salle = (**id_salle**, capacite, #code_postal)

Utilisateur = (**id_utilisateur**, prenom, nom, pseudo, email, mot_de_passe)

Carte = (**id_carte**, date_creation, #id_utilisateur)

Abonnement = (**id_abonnement**, type_abonnement, date_debut, date_fin, #id_carte)

4. *Diagramme applicatif :*

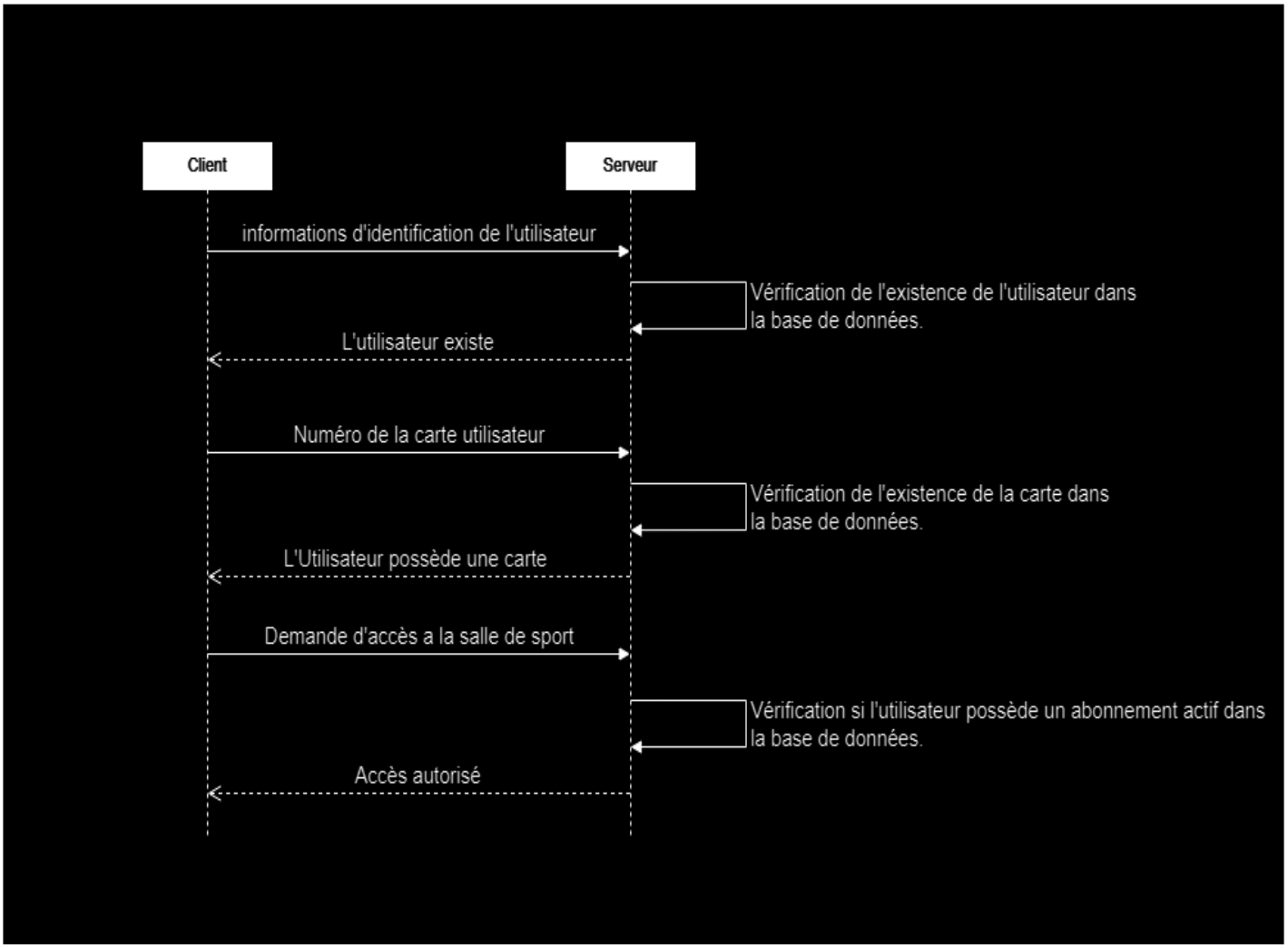

Figure 3 : Diagramme applicatif dans le meilleur des cas.

5. *Jeu de données :*

Ville = (**code_postal**, nom)

Code_postal	**nom**	**rue**
94000	CRETEIL	Rue de l'Espérance
94110	ARCUEIL	Avenue Massenet
95810	Arronville	Rue de Berville
95400	Arnouville	Avenue Diderot
92800	PUTEAUX	Rue Agathe
92400	COURBEVOIE	Rue Fallet

Analyse et conception

Salle = (**id_salle**, capacite, #code_postal)

Id_salle	capacite	#code_postal
S25	200	94000
S124	100	94110
S200	250	95810
S40	150	95400
S230	300	92800
S87	170	92400

Utilisateur = (**id_utilisateur**, nom, prenom, pseudo, email, mot_de_passe)

Id_utilisateur	nom	prenom	pseudo	email	Mot_de passe
U1	Gouarab	Ali	GA1	gouarab.ali@gmail.com	GA123!
U20	Armand	Marie	AM20	armand.marie@gmail.com	AM123!
U34	Gustave	Roger	GR34	gustave.roger@gmail.com	GR123!
U102	Augustin	Boyer	AB102	augustin.boyer@gmail.com	AB123!
U64	Manon	Rey	MR64	manon.rey@gmail.com	MR123!
U41	Maxime	Renaud	MR41	maxime.renaud@gmail.com	MR123 !

Carte = (**id_carte**, date_creation, #id_utilisateur)

Id_carte	Date_creation	#id_utilisateur
C25	10/10/2024	U1
C15	09/10/2024	U20
C47	10/09/2024	U34
C100	23/07/2024	U102
C41	10/08/2024	U64
C87	03/06/2024	U41

Abonnement = (**id_abonnement**, type_abonnement, date_debut, date_fin, #id_carte)

Id_abonnement	type_abonnement	Date_debut	Date_fin	#id_carte
A14	Basic	08/10/2024	08/01/2025	C25
A26	Basic	07/10/2024	07/11/2024	C15
A789	Premium	06/06/2024	06/06/2025	C47
A21	Basic	07/08/2024	07/11/2024	C100
A47	Basic	10/10/2024	10/01/2025	C41
A1000	Premium	29/09/2024	29/12/2024	C87

Implémentation

Implémentation

1. Client et Serveur :

Pour cette version du compte rendu le serveur et le client ont été tous les deux écrits en Python.

Mais pour les prochaines versions le serveur restera en python mais le client sera écrit soit en « java » ou bien en « C »

Test du client/serveur Dans le meilleur des cas :

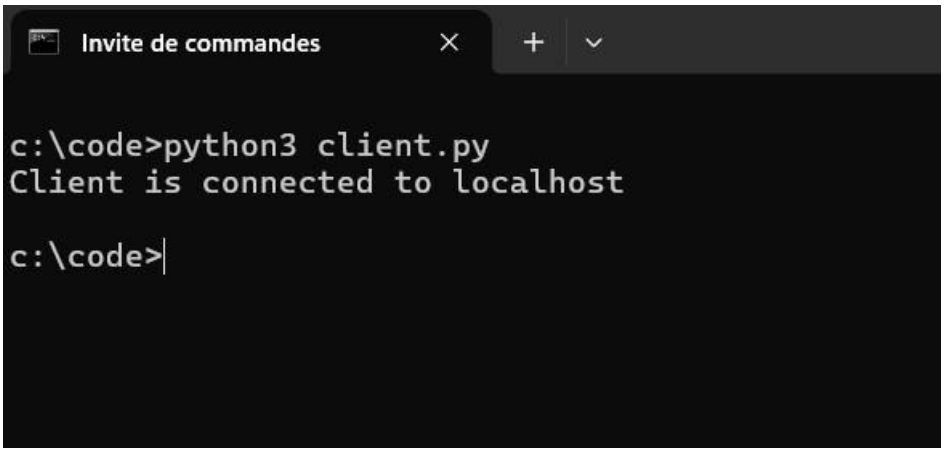

Figure 4 : Le client effectue une requête valide vers le serveur.

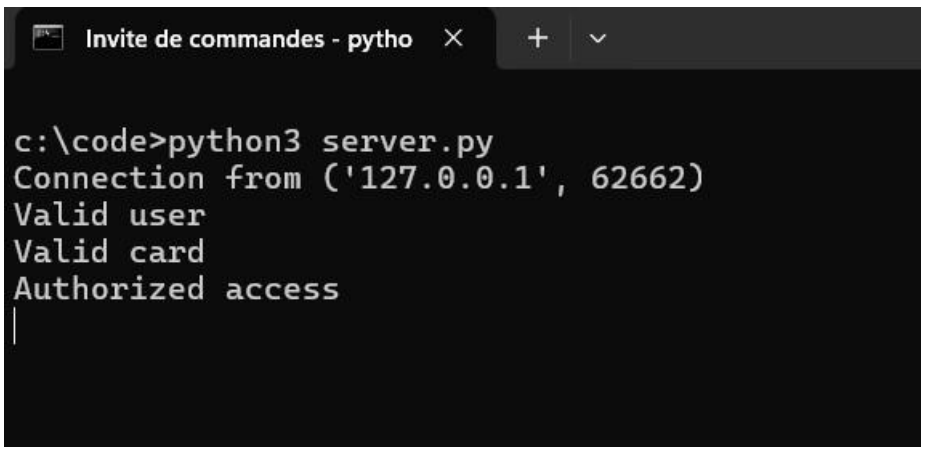

Figure 4.1 : Le serveur autorise l'accès au client.

Test du client/serveur Dans le cas où l'utilisateur ne possède pas d'abonnement actif :

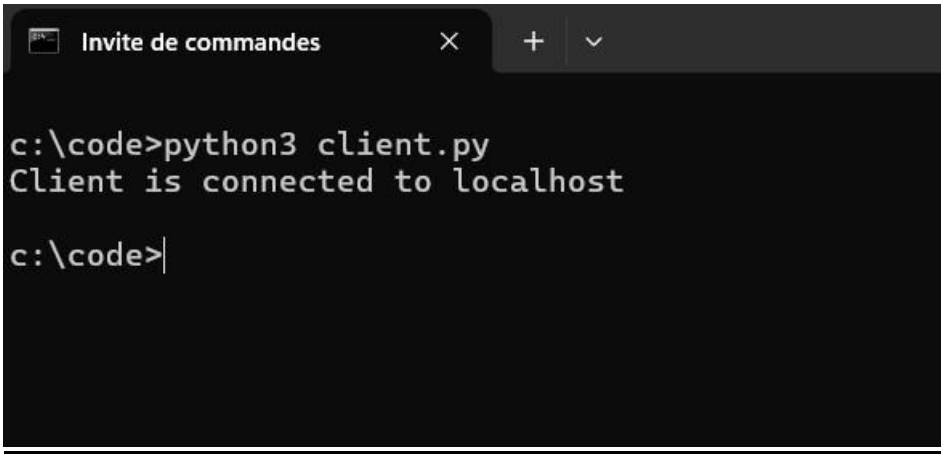

Figure 5 : Le client effectue une requête invalide vers le serveur.

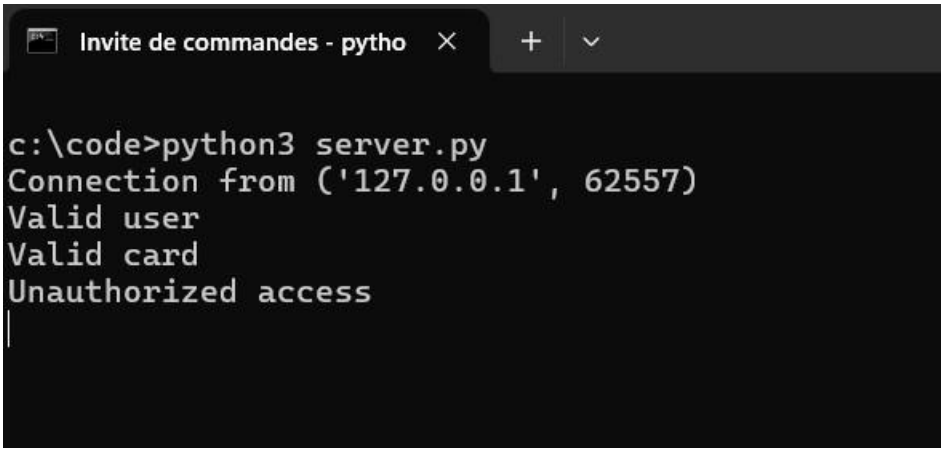

Figure 5.1 : Le serveur refuse l'accès au client.

Test du client/serveur Dans le meilleur des cas avec netcat :

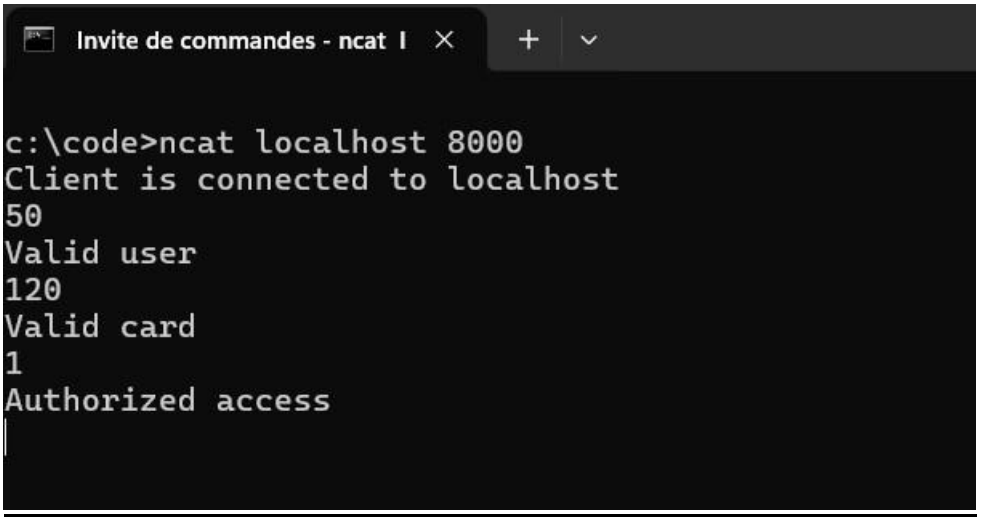

Figure 6 : requête valide vers le serveur avec netcat.

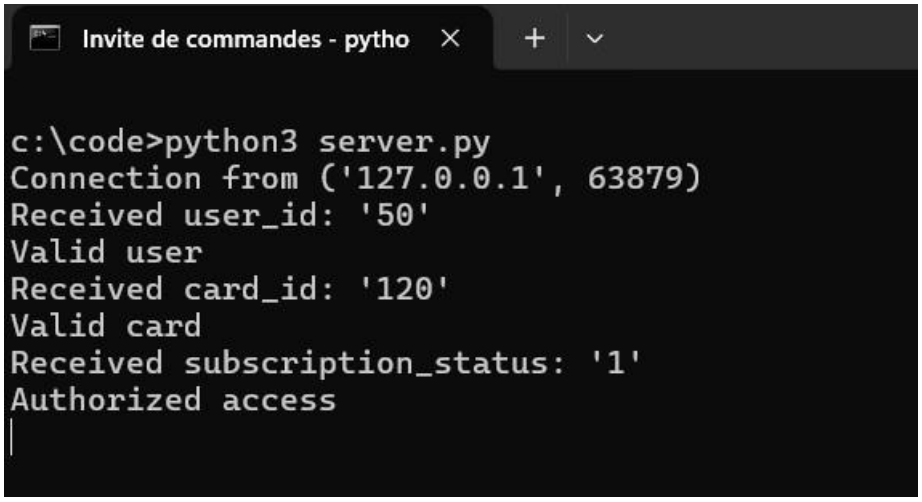

Figure 6.1 : réponse du serveur a la requête netcat.

Test du client/serveur Dans le cas où l'utilisateur ne possède pas d'abonnement actif avec netcat :

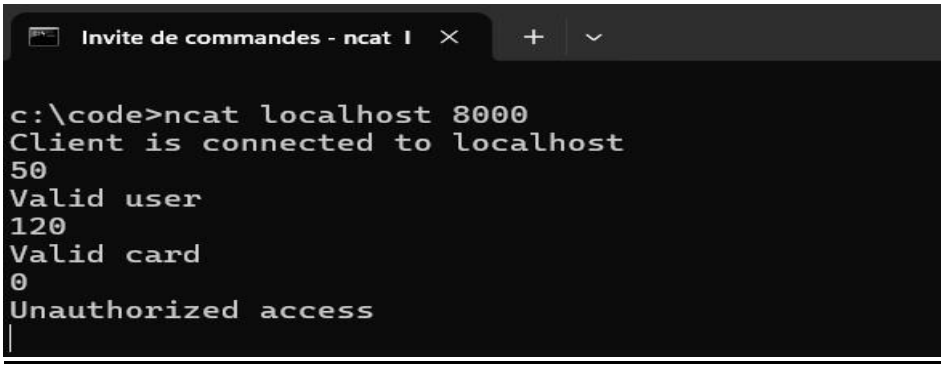

Figure 7 : requête invalide vers le serveur avec netcat.

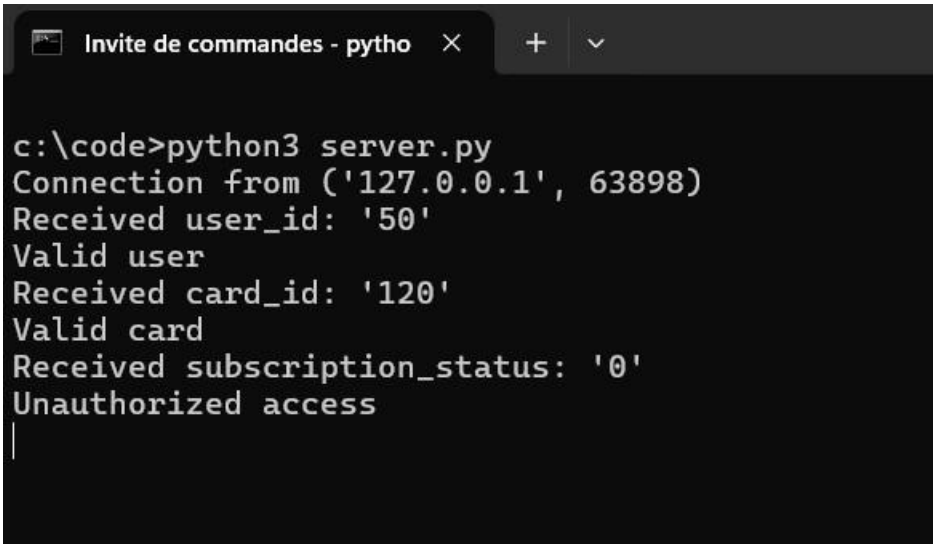

Figure 7.1 : réponse du serveur a la requête netcat.