

# Rapport de Projet BD/Réseaux

*CineMAX : Gestion d'entrées de cinéma*

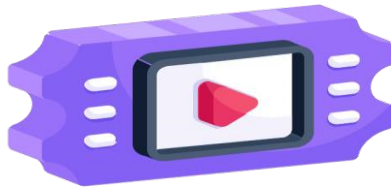
Version 4 du Groupe D1 L3I

Fait le 01-12-2024

FWALA YENGA MUBY WENU Yvon

JIN Zhuoyuan

OUATATI Kamil



# Tableau de Revision

Date	Auteur	Section(s)	Commentaires
19/09/2024	Tous	Introduction, Dictionnaire de données, D'Echanges	Version 1.1, Manque beaucoup de contrainte et d'exemple
30/09/2024	Tous		Version2.0
03/10/2024	Yvon	Modifier et améliorer le dictionnaire de données et ajouté le modèle relationnel	Version2.1
03/10/2024	Elie	Ajouter le diagramme de séquence pour les utilisateurs et les administrateurs	Version2.2
05/10/2024	Yvon, Kamil	Rajouter le Schéma E/A et Schéma relationnel	Version 2.3
06/10/2024	Elie,Yvon	Améliorer le diagramme ER	Version 2.4
14/10	Yvon	Corriger le dictionnaire, schema E/R et logique	Version 2.5
18/10	Elie	Ajouter la partie réseau pour réaliser la connexion entre le client et le serveur via le protocole TCP	Version2.6
19/10	Elie	Modifier le diagramme de séquence pour client et admin	Version2.7
20/10	Elie	formater le document Word	Version2.7.1
20/10	Yvon	Ajouter le jeu de donnée	Version2.8
01/11	Yvon, Kamil , Elie	Continuer le jeu de donne	Version2.9
09/11	Yvon, Kamil , Elie	Créer des scenario d'erreur dans le diagramme applicatif	Version3.0
10/11	Yvon, Kamil , Elie	Améliorer le jeu de données et des échanges client et serveur	Version3.1
28/11	Yvon, Kamil , Elie	Rajouter les requêtes dml et corriger les erreurs	Version3.5
01/12	Yvon, Kamil , Elie	Finaliser le rapport	Version 4

# Table des matières

<u>Introduction</u> .....	6
<u>3.1 Contexte et Objectif</u> .....	6
<u>Planning</u> .....	7
<u>4.1 Répartitions des taches</u> .....	7
<u>Base de données</u> .....	7
<u>5.1 Dictionnaire de données</u> .....	8
<u>5.2 Schéma E/A (MCD)</u> .....	10
<u>5.3 Schéma relationnel (MLD)</u> .....	10
<u>Jeu de données</u> .....	12
<u>6.1 Table Cinema</u> .....	12
<u>6.2 Table Personne</u> .....	12
<u>6.3 Table Film</u> .....	13
<u>6.5 Table Seance</u> .....	15
<u>6.6 Table Client</u> .....	15
<u>6.7 Table Employee</u> .....	15
<u>6.8 Table Acteur</u> .....	16
<u>6.9 Table Machine</u> .....	16
<u>6.10 Table Ticket</u> .....	16
<u>6.11 Table Siege</u> .....	16
<u>6.12 Table Jouer</u> .....	17
<u>Diagramme applicatif</u> .....	18
<u>7.1 Cas Parfait</u> .....	18
<u>7.2 Cas Premier Probleme</u> .....	19
<u>7.3 Cas Deuxième Problème</u> .....	20
<u>7.4 Cas Troisième Problème</u> .....	21
<u>7.5 Cas Quatrième Problème</u> .....	22
<u>Réseau</u> .....	23
<u>8.1 Serveur en python</u> .....	23
<u>8.2 Client en java</u> .....	23
<u>8.3 Echange via Netcat</u> .....	24
<u>ANNEXE</u> .....	28
<u>Code de test de connectivité réseau</u> .....	28
<u>Requêtes DML (en SQL)</u> .....	34

<u>SITE-WEB.....</u>	<u>44</u>
<u>L'objectif et les fonctionnalités du site .....</u>	<u>44</u>

## Tables des figures

<u>Figure 1 diagramme de GANTT.....</u>	<u>5</u>
<u>Figure 2 diagramme des ressources .....</u>	<u>5</u>
<u>Figure 3 schéma E/A.....</u>	<u>8</u>
<u>Figure 4 schéma application Client.....</u>	<u>11</u>
<u>Figure 5 schéma application Adminstrateur .....</u>	<u>12</u>
<u>Figure 6 Exemple de tableau.....</u>	<u>13</u>
<u>Figure 7 Reseau-TCP-Serveur .....</u>	<u>13</u>
<u>Figure 8 Reseau-TCP-Client.....</u>	<u>13</u>

# Introduction

## 3.1 Contexte et Objectif

Avec l'ère numérique qui progresse, de plus en plus de consommateurs optent pour l'achat de billets en ligne afin d'améliorer leur expérience cinématographique et leur commodité. Pour répondre à cette demande, nous avons pour objectif de créer un système de contrôle d'accès rapide basé sur des codes QR pour les cinémas, qui simplifie considérablement le processus d'entrée, réduit le temps d'attente pour les clients et augmente leur satisfaction. En parallèle, ce système vise à fournir aux administrateurs de cinéma une plateforme efficace et facile à utiliser pour la gestion des données et l'analyse, facilitant la gestion des séances de films et l'analyse des données clients.

### Les objectifs principaux du système incluent :

1. **Améliorer l'expérience utilisateur** : Grâce à l'achat en ligne de billets et à l'entrée par code QR, le temps d'attente des spectateurs est réduit, leur permettant d'entrer plus facilement dans le cinéma.
2. **Gestion des billets et analyse des données** : Le système fournit aux cinémas des données en temps réel sur la billetterie, facilitant ainsi la gestion des séances, la répartition des sièges et les ventes de billets. Il permet également d'analyser la demande du marché pour mieux organiser les projections, améliorant ainsi l'efficacité opérationnelle globale.

3.

## Planning

Nous avons décidé de faire un diagramme de Gantt pour avoir une approche plus structurée et itérative de notre projet afin de finir le projet dans les temps.



Figure 1 diagramme de GANTT

## 4.1 Répartitions des taches

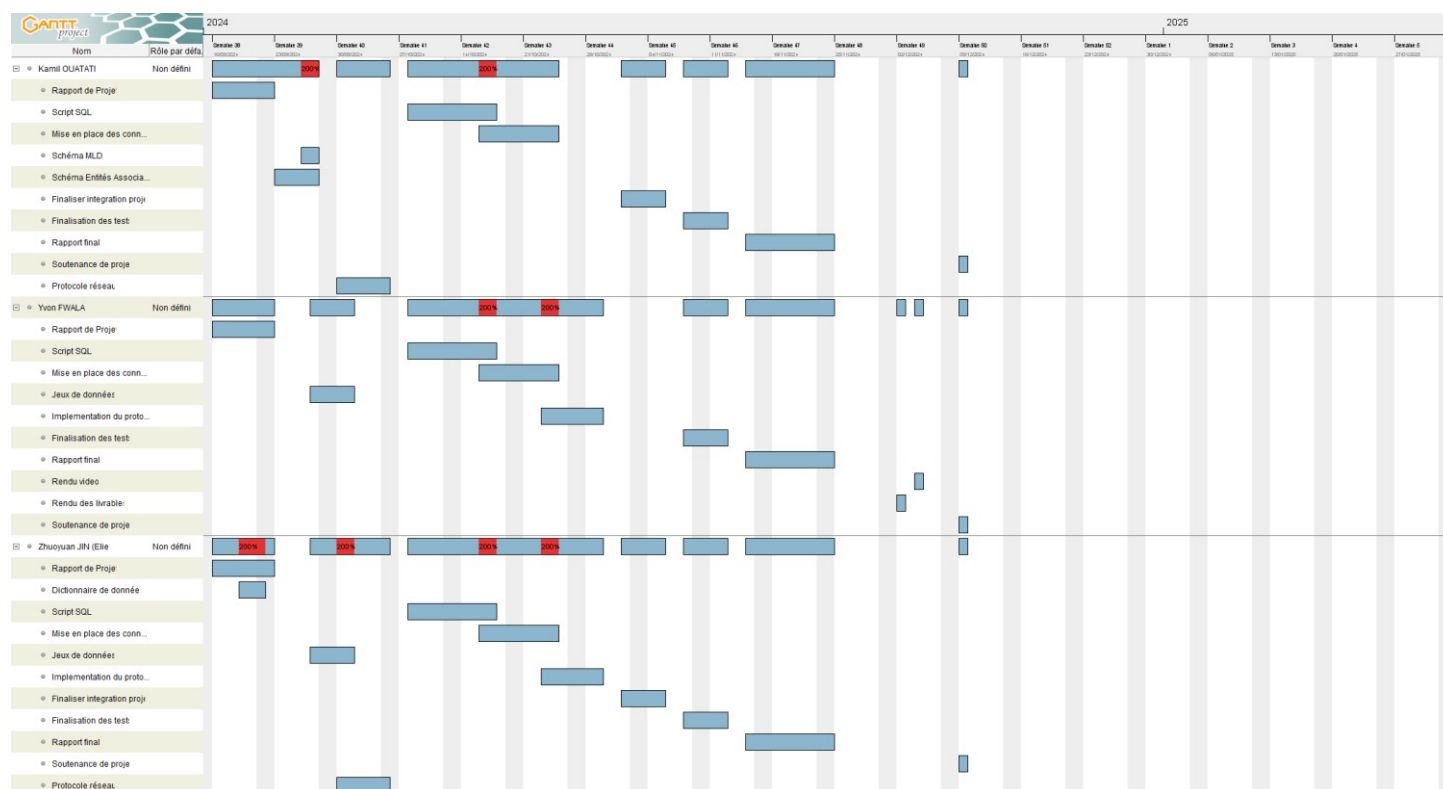


Figure 2 diagramme des ressources

# Base de données

## 5.1 Dictionnaire de données

	Champ	Type et Taille	NNN	Contraintes	Remarque	Exemple
Cinema	id_cinema	CHAR (8)	NN	PK	CHECK (id_cinema ~* '^CIN-[0-9]{4}\$')	CIN-9501
	nom_cin	VARCHAR (100)	NN	alphanumerique		UGC
	adresse	VARCHAR (225)	NN	alphanumerique	Check ^([0-9]+[ ]+[A-Za-z ]+(St Street Ave Avenue Blvd Boulevard Rue rue)?[ ]+[A-Za-z ]+\$	1 Avenue de Verdun
	tel	CHAR (10)	NN	numerique	CHECK (tel ~ '^+?[0-9\s-()]+\$')	06123456
	web	VARCHAR (40)	NN	alphanumerique	Check ^(http https)://[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}(/.*)?\$	<a href="http://www.ugc.fr">www.ugc.fr</a>
Film	id_film	CHAR(10)	NN	PK	CHECK (id_film ~* '^F-\d{8}\$')	F-18122024
	titre	VARCHAR (50)	NN	alphanumerique		SONIC 3
	date_sortie	DATE	NN	DATE		2024-12-18
	duree	TIME	NN	TEMPS	(par minute)	02:28:00
	director	VARCHAR (100)	NN	alphabetique		Joss Whedon
	genre	VARCHAR (50)	N	alphabetique		Horreur
	nationalité	CHAR(2)	NN	alphabetique	code ISO de la nationalité 'FR', 'EN', 'SP'	FR
Seance	id_seance	CHAR (7)	NN	PK	CHECK (id_seance ~* '^SEA\d{4}\$')	SEA1078
	heure_debut	TIMESTAMP	NN	DATE, TIME		2024-12-01 18:00
	heure_fin	TIMESTAMP	NN	DATE, TIME		2024-12-01 20:30
	prix	FLOAT	NN	Numerique > 0	CHECK (prix > 0)	15,75
	langue	VARCHAR (15)	NN	Alphabetique		FR
	status	BOOLEAN	NN	Alphabetique	CHECK Y/N	Y
	play	CHAR(1)	NN	Alphabetique	CHECK (play = 'E' OR play = 'S' OR play = 'F') Version de lecture : E = version originale, S = sous-titres, F = doublage	S

Salle	id_salle	CHAR (7)	NN	PK	CHECK (id_salle ~* '^SAL\d{4}\$')	SAL0455
	nom_salle	VARCHAR (50)	NN	alphanumerique		Salle 9
	active	Boolean	NN	Alphabétique		Y
	type_projection	VARCHAR (50)	NN	alphanumerique		IMAX
	capacite	INT	NN	Numerique	Nombre de sièges CHECK (capacite > 0)	50
Sieges	id_siege	CHAR (7)	NN	PK	CHECK (id_siege ~* '^SIE\d{4}\$')	SIE0595
	num_place	INT	NN	numerique		28
	num_rangé	INT	NN	numerique		3
	type	CHAR(1)	NN	alphabétique	CHECK (type_p = 'N' OR type_p = 'V') VIP/NORMAL	V
	statut	CHAR (1)	NN	alphabétique	CHECK (st = 'F' OR st = 'T') F:occupé T:disponible	F
Client	id_client	CHAR (8)	NN	PK	CHECK (id_client ~* '^CLI-\d{4}\$')	CLI-0001
	birth	DATE	NN	DATE	CHECK (birth BETWEEN '1900-01-01' AND CURRENT_DATE - INTERVAL '13 years')	1900-01-01
	lvl	CHAR (1)	NN	Enum	CHECK (lvl = 'V' OR lvl = 'N'), DEFAULT 'N'	N
	date_insc	DATE	N	DATE	DEFAULT CURRENT_DATE, CHECK (date_insc BETWEEN '1900-01-01' AND CURRENT_DATE)	2003-01-01
Acteur	id_acteur	CHAR (8)	NN	PK	CHECK (id_acteur ~* '^ACT-\d{4}\$')	ACT-8076
	nom_acteur	VARCHAR (30)	NN	alphabétique		Holland
	prenom_acteur	VARCHAR (30)	NN	alphabétique		Tom
Ticket	id_ticket	CHAR (8)	NN	PK	CHECK (id_ticket ~* '^TIC-\d{4}\$')	TIC-0201
	date_creation	DATE	NN	DATE	DEFAULT CURRENT_TIMESTAMP	2024-11-29 10:00
	date_expiration	DATE	NN	DATE	DEFAULT CURRENT_TIMESTAMP + INTERVAL '1 year'	2025-11-29 10:00
	statut	CHAR(1)	N	alphabétique	CHECK (used = 'U' OR used = 'N') DEFAULT 'N' Utilisé/Non utilise	U
	premium	CHAR(1)	NN	alphanumerique	CHECK (premium = 'V' OR premium = 'N') DEFAULT 'N'	V
Employé	id_employé	CHAR (8)	NN	PK	CHECK (id_employé ~* '^EMP-\d{4}\$')	EMP-3071
	poste	VARCHAR (50)	NN	alphanumérique		Gestionnaire



	lvl_acces	CHAR(1)	NN	Nnum	CHECK (lvl_acces = 'M' OR lvl_acces = 'I' OR lvl_acces = 'N')	I
Machine	id_machine	CHAR (8)	NN	PK	CHECK (id_machine ~* '^MAC-\d{4}\$')	MAC-0002
	fabricant	VARCHAR (25)	NN	alphabetique		MetaQuip
	date-achat	DATE	NN	DATE	DEFAULT CURRENT_DATE, CHECK (date_achat BETWEEN '1900-01-01' AND CURRENT_DATE)	2024-01-01
	statut	CHAR(1)	NN	alphabetique	CHECK (stat = 'W' OR stat = 'N') DEFAULT 'W' W:working/N:no work	W
Personne	id_personne	SERIAL	NN	PK		1
	nom_per	VARCHAR (50)	NN	alphabetique		Fwala
	prenom_per	VARCHAR (50)	NN	alphabetique		Yvon
	sexe	CHAR(1)	N	alphabetique	CHECK (sexe = 'H' OR sexe = 'F')	H
	login_per	VARCHAR (50)	NN	alphanumerique		Yveezy95
	mot_de_passe_per	VARCHAR (200)	NN	alphanumerique		*****
	email_per	VARCHAR (100)	NN	alphanumerique	CHECK (email ~* '^[@]+@[^@]+.[^@]+\$')	yvon.fwala92@gmail.com

## 5.2 Schéma E/A (MCD)

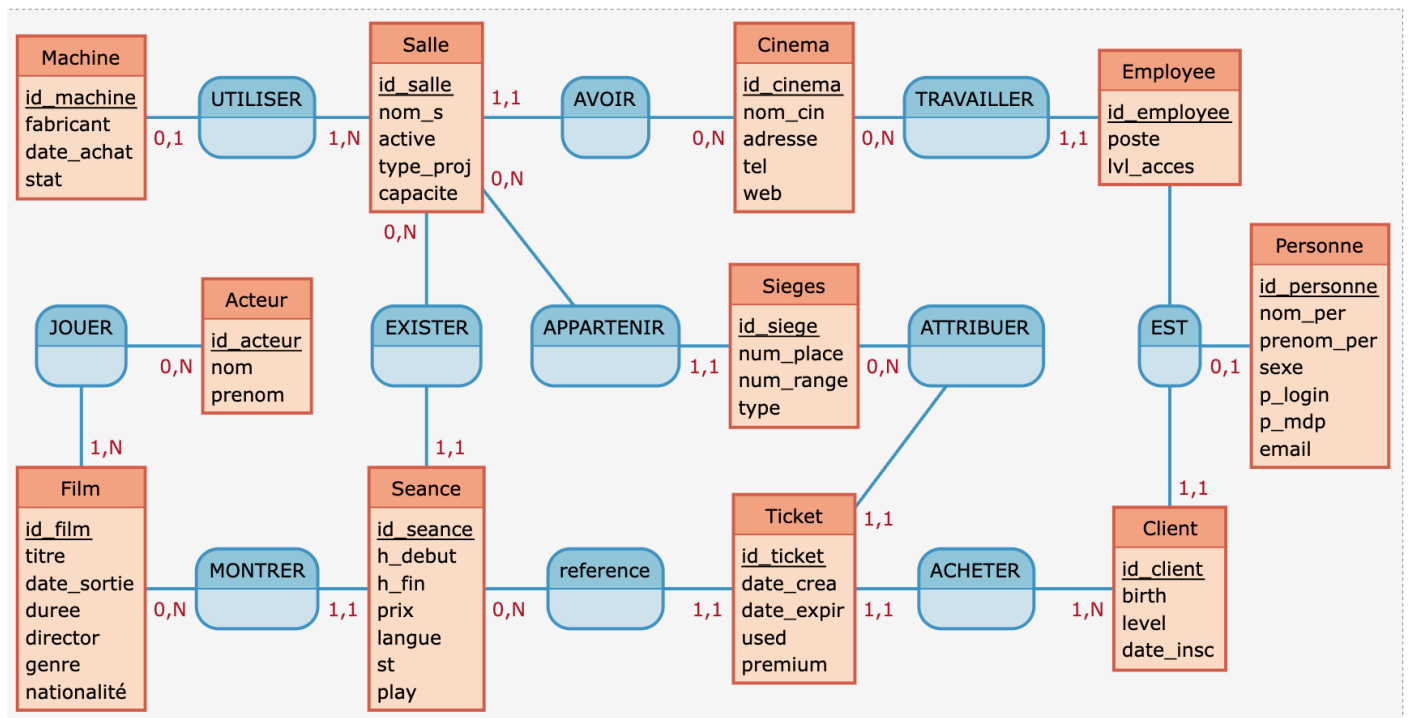


Figure 3 schéma E/A

## 5.3 Schéma relationnel (MLD)

**PERSONNE** (id\_personne, nom\_per, nom\_prenom, sexe, p\_login, p\_mdp, email);  
**CLIENT** (id\_client, #id\_personne, birth, level, date\_insc);  
**EMPLOYEE** (id\_employee, #id\_personne, poste, lvl\_acces, #id\_cinema);  
**CINEMA** (id\_cinema, nom\_cin, adresse, tel, web);  
**FILM** (id\_film, titre, date\_sortie, duree, director, genre, nationalité);  
**ACTEUR** (id\_acteur, nom, prenom);  
**SALLE** (id\_salle, nom\_s, active, type\_proj, capacite, #id\_cinema);  
**SEANCE** (id\_seance, h\_debut, h\_fin, prix, langue, st, play, #id\_salle, #id\_film);  
**SIEGE** (id\_siege, num\_place, num\_rangé, type, st, #id\_salle);  
**TICKET** (id\_ticket, date\_crea, date\_expir, used, premium, #id\_siege, #id\_client, #id\_seance);  
**MACHINE** (id\_machine, fabricant, date\_achat, stat, #id\_salle);  
**JOUER** (#id\_film, #id\_acteur);

# Jeu de données

## 6.1 Table Cinema

CINEMA (**id\_cinema**, nom\_cin, adresse, tel, web);

id_cinema	nom_cin	Adresse	tel	web
CIN-0001	UGC Cine Cite	75 rue de Paris	+33123456789	<a href="http://www.ugc.fr">http://www.ugc.fr</a>
CIN-0002	Gaumont Wilson	30 Boulevard de Strasbourg	+33567891234	<a href="http://www.gaumont.fr">http://www.gaumont.fr</a>
CIN-0003	MK2 Bibliothèque	162 Rue Tolbiac	+33678901234	<a href="http://www.mk2.fr">http://www.mk2.fr</a>
CIN-0004	Cinéma du Panthéon	13 Rue Victor Cousin	+33167890123	<a href="http://www.pantheon.fr">http://www.pantheon.fr</a>
CIN-0005	Le Louxor	170 Boulevard Magenta	+33234567890	<a href="http://www.louxor.fr">http://www.louxor.fr</a>
CIN-0006	Cinémathèque Française	51 Rue de Bercy	+33378901234	<a href="http://www.cinematheque.fr">http://www.cinematheque.fr</a>
CIN-0007	Le Balzac	1 Rue Balzac	+33123456780	<a href="http://www.lebalzac.fr">http://www.lebalzac.fr</a>

## 6.2 Table Personne

PERSONNE (**id\_personne**, nom\_per, nom\_prenom, sexe, p\_login, p\_mdp, email);

Id_Personne	Nom_per	Prenom_per	sexe	P_login	P_mdp	email
1	Dupont	'Jean'	H	jdupont	pass123	jean.dupont@example.com
2	Martin	Sophie	F	smartin	sophie2023	sophie.martin@example.com
3	Lemoine	Alice	F	alemoine	alice567	alice.lemoine@example.org
4	Durand	Pierre	H	pdurand	pierred2022	pierre.durand@example.net
5	Moreau	Julie	F	jmoreau	julie654	julie.moreau@example.com
6	Bernard	Paul	H	pbernard	paul123	paul.bernard@example.org

7	Petit	Marie	F	mpetit	marie2023	marie.petit@example.net
8	Rousseau	Luc	H	lrousseau	luc456	luc.rousseau@example.com
9	Blanc	Emma	F	eblanc	emma2023	emma.blanc@example.org
10	Girard	Nicolas	H	ngirard	nico789	nicolas.girard@example.net
11	Fwala	Yvon	H	yveezy	yvon-z	yvon.fwala@example.com

## 6.3 Table Film

FILM (**id\_film**, titre, date\_sortie, duree, director, genre, nationalité);

Id_film	Titre	Date_sortie	duree	director	Genre	nationalité
F-00000001	Inception	2010-07-16	02:28:00	Christopher Nolan	Science-fiction	US
F-00000002	Parasite	2019-05-30	02:12:00	Bong Joon-ho	Thriller	KR
F-00000003	Avatar	2009-12-18	02:42:00	James Cameron	Action	US
F-00000004	Le Roi Lion	1994-06-15	01:28:00	Roger Allers	Animation	US
F-00000005	Titanic	1997-12-19	03:14:00	James Cameron	Romance	US
F-00000006	Amélie	2001-04-25	02:02:00	Jean-Pierre Jeunet	Comédie	FR
F-00000007	Interstellar	2014-11-07	02:49:00	Christopher Nolan	Science-fiction	US
F-00000008	La Haine	1995-05-31	01:38:00	Mathieu Kassovitz	Drame	FR
F-00000009	Joker	2019-10-04	02:02:00	Todd Phillips	Drame	US
F-00000010	The Dark Knight	2008-07-18	02:32:00	Christopher Nolan	Action	US

## 6.4 Table Salle

SALLE (**id\_salle**, nom\_s, active, type\_proj, capacite, #id\_cinema);

Id_salle	nom_s	active	type_proj	capacite	Id_cinema
SAL0001	Salle 1	TRUE	IMAX	200	CIN-0001
SAL0002	Salle 2	FALSE	Standard	100	CIN-0002
SAL0003	Salle 3	TRUE	4DX	150	CIN-0002
SAL0004	Salle principal	TRUE	IMAX	200	CIN-0002
SAL0005	Salle 5	FALSE	3D	100	CIN-0002
SAL0006	Salle 6	TRUE	Standard	100	CIN-0002
SAL0007	Salle 7	FALSE	IMAX	200	CIN-0002

## 6.5 Table Seance

SEANCE (**id\_seance**, h\_debut, h\_fin, prix, langue, st, play, #id\_salle, #id\_film);

Id_seance	h_debut	h_fin	prix	langue	st	play	#id_film	#id_salle
SEA0001	2024-11-26 15:00:00	2024-11-26 17:28:00	19.75	Français	Disponibl e			SAL-0001

## 6.6 Table Client

CLIENT (**#id\_client**, birth, level, date\_insc);

Id_client	id_personne	birth	level	date_insc
CLI-1001	1	1995-05-15	N	2024-01-01
CLI-1002	3	1988-08-08	V	2024-01-02

## 6.7 Table Employee

EMPLOYEE (**#id\_employee**, poste, lvl\_acces, #id\_cinema);

id_employee	id_personne	poste	lvl_acces	#id_cinema
EMP-0001	11	Manager	M	CIN-0001
EMP-0002	11	Caissier	M	CIN-0002

## 6.8 Table Acteur

ACTEUR (**id\_acteur**, nom ,prénom);

id_acteur	nom	prenom
ACT-0001	Radcliffe	Daniel
ACT-0003	Christensen	Hayden

## 6.9 Table Machine

MACHINE (**id\_machine**, fabricant, date\_achat, stat, #id\_salle) ;

id_machine	fabricant	date_achat	stat	#id_salle
MAC-0027	Sony	05-12-2023	W	SAL-0001
MAC-0011	Panasonic	23-08-2024	N	SAL-0004

## 6.10 Table Ticket

TICKET (**id\_ticket**, date\_crea, date\_expir, used, premium, #id\_siege, #id\_client);

id_ticket	date_crea	date_expira	used	premium	#id_siege	#id_client	#id_seance
TIC-1001	2024-11-25 10:00:00	2025-11-25 10:00:00	N	N	SIE0001	CLI-1001	SEA0001
TIC-1002	2024-11-25 12:00:00	2025-11-25 12:00:00	N	V	SIE0002	CLI-1002	SEA0001

## 6.11 Table Siege

SIEGE (**id\_siege**, num\_place, num\_rangé, type, st, #id\_salle);

<b>id_siege</b> ,	num_place	num_rangé	Type	st	#id_salle
SIE0001	1	1	N	T	SAL0001
SIE0002	2	3	V	F	SAL0002
SIE0003	3	5	V	T	SAL0002

## 6.12 Table Jouer

JOUER ([#id\\_film](#), [#id\\_acteur](#)).

#id_film	#id_acteur
F-000000001	ACT-1001
F-000000002	ACT-1001

# Diagramme applicatif

Nous avons pensé à plusieurs cas d'échange réseaux avec différents problèmes. En effet, dans le sujet, il nous est demandé de concevoir un protocole robuste.

Pour éviter toute grosse erreur entre le client et les serveurs, nous avons pensé à utiliser le protocole TCP. Bien que ce protocole garantisse une connexion fiable, nous avons mis en place un mécanisme permettant au serveur de vérifier l'état de l'échange et de reprendre à partir de l'étape où l'échange a été interrompu, minimisant ainsi les risques de perte de données.

## 7.1 Cas Parfait

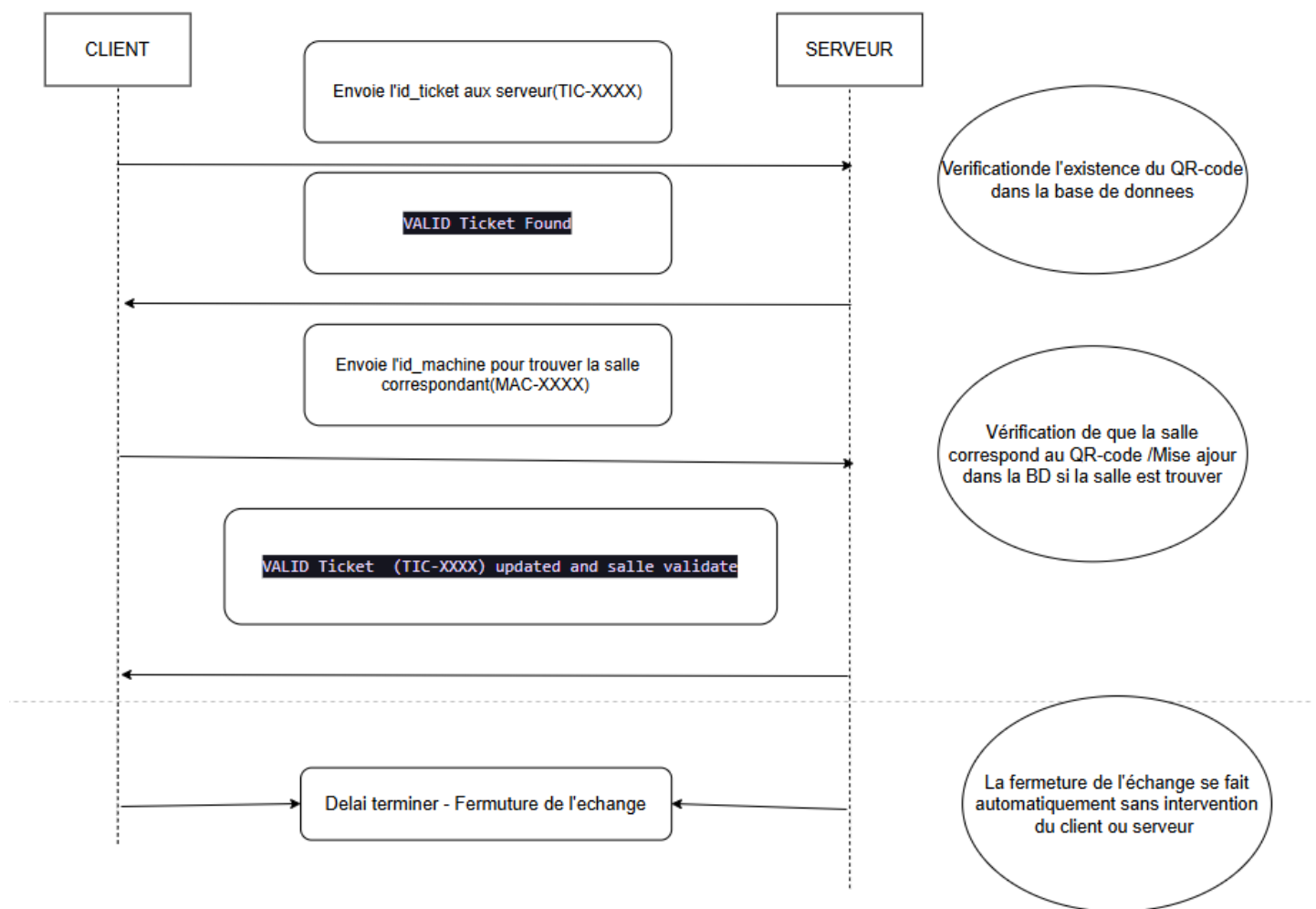


Figure 4 schéma application Client



## 7.2 Cas Premier Probleme

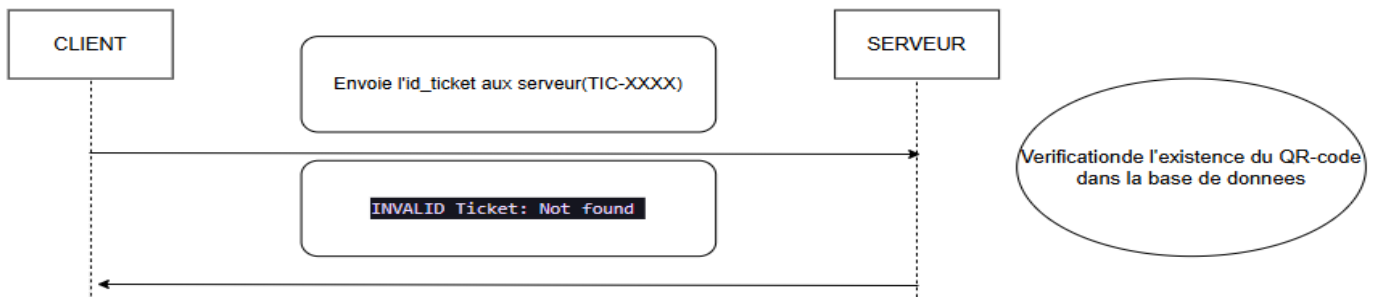


Figure 5 QR code n'existe pas dans la Base de données

## 7.3 Cas Deuxième Problème



## 7.4 Cas Troisième Problème

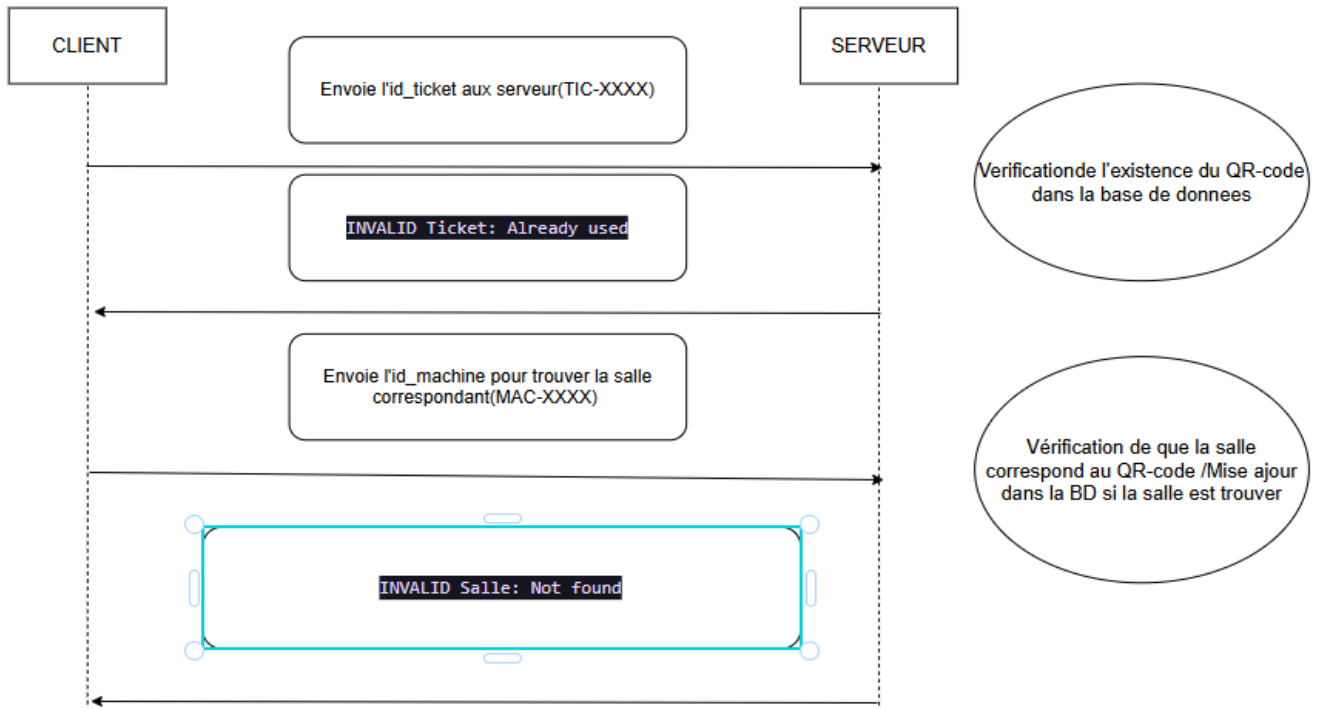
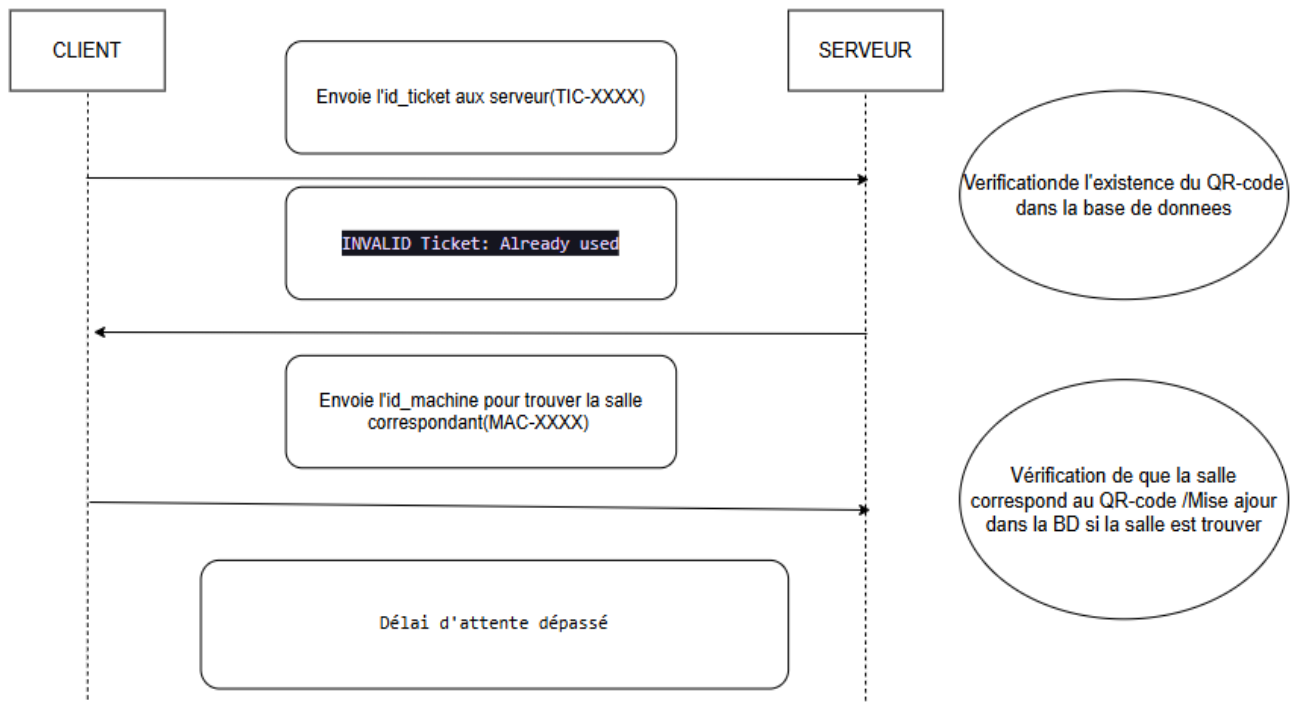


Figure 7 QR-Code déjà utilisé

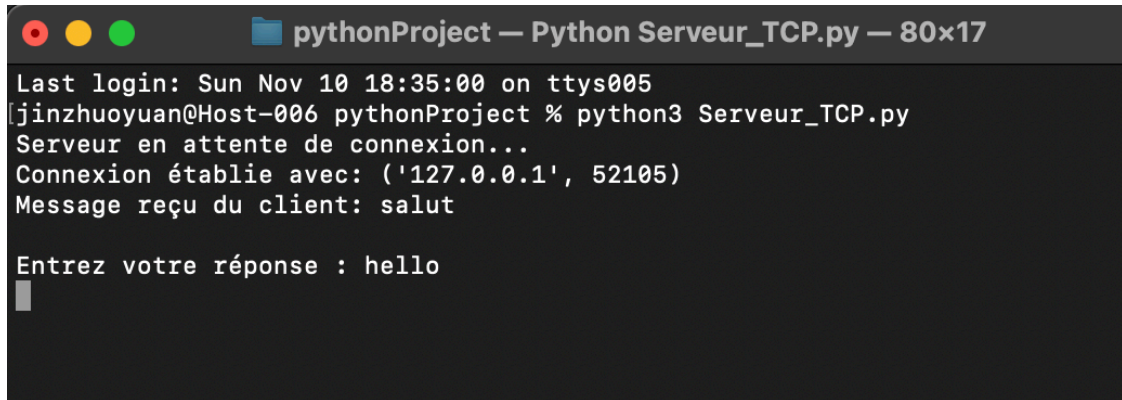
## 7.5 Cas Quatrième Problème



# Réseau

## 8.1 Serveur en python

Nous utilisons python pour implémenter le serveur.

A terminal window titled "pythonProject — Python Serveur\_TCP.py — 80x17". The output shows the server starting, receiving a connection from '127.0.0.1' on port 52105, and receiving the message "salut". It then prompts for a response, which is "hello".

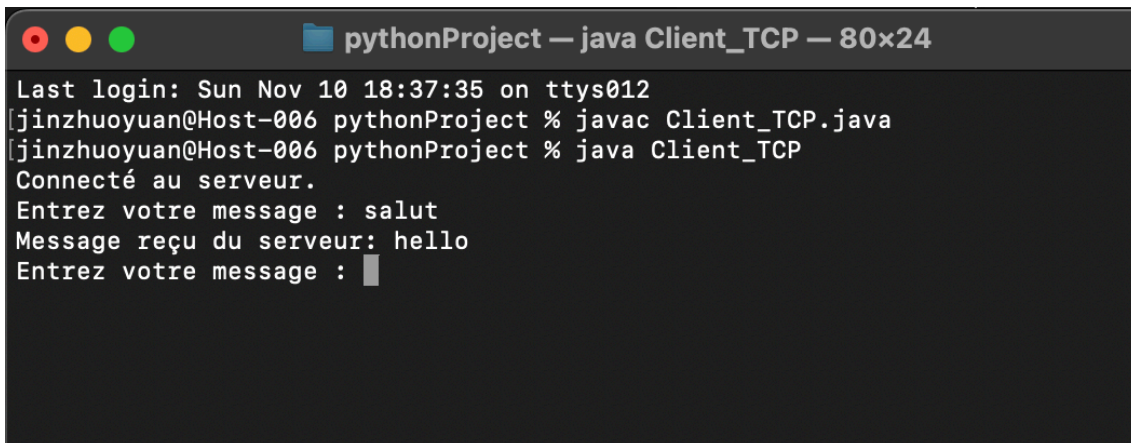
```
Last login: Sun Nov 10 18:35:00 on ttys005
[jinzhuoyuan@Host-006 pythonProject % python3 Serveur_TCP.py
Serveur en attente de connexion...
Connexion établie avec: ('127.0.0.1', 52105)
Message reçu du client: salut

Entrez votre réponse : hello
```

Figure 9 Reseau-TCP-Serveur

## 8.2 Client en java

Nous utilisons Java pour les fonctionnalités côté client

A terminal window titled "pythonProject — java Client\_TCP — 80x24". The output shows the client compiling and running the "Client\_TCP.java" file, connecting to the server, sending the message "salut", and receiving the response "hello". It then prompts for another message.

```
Last login: Sun Nov 10 18:37:35 on ttys012
[jinzhuoyuan@Host-006 pythonProject % javac Client_TCP.java
[jinzhuoyuan@Host-006 pythonProject % java Client_TCP
Connecté au serveur.
Entrez votre message : salut
Message reçu du serveur: hello
Entrez votre message :
```

Figure 10 Reseau-TCP-Client

## 8.3 Echange via Netcat

Ici on teste le serveur via Netcat, on voit que le client peut se connecter dessus et nous envoyer des messages. Ici on a décidé d'appliquer le cas parfait.

```
VALID Ticket foundMAC-1002
VALID Salle foundSERVER TIMEOUT^C
C:\Users\yveezy>ncat localhost 5000
TIC-0025
VALID Ticket foundMAC-1002
VALID Salle foundUPDATE TIC-0025
UNKNOWN COMMANDSERVER TIMEOUTUPDATE TIC-0025
^C
C:\Users\yveezy>ncat localhost 5000
TIC-0025
VALID Ticket foundMAC-1002
VALID Ticket TIC-0025 updated and salle validated
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

```
PS C:\Users\yveezy\Desktop\yvon\S5\BD\projet_1> python serv2.py
Connexion réussie à la base de données.
Serveur TCP démarré sur 127.0.0.1:5000
Connexion établie avec ('127.0.0.1', 7183)
Message reçu : TIC-0025 depuis ('127.0.0.1', 7183)
Ticket ID extrait: 0025
Ticket ID complet pour la recherche : TIC-0025
Ticket trouvé : ('TIC-0025', 'N')
VALID Ticket found
Réponse envoyée : VALID Ticket found
Message reçu : MAC-1002 depuis ('127.0.0.1', 7183)
Machine ID extrait: 1002
Machine ID complet pour la recherche : MAC-1002
Salle trouvée : ('SAL0001', 'Salle 1')
Réponse envoyée : VALID Ticket TIC-0025 updated and salle validated
█
```

Cas ou le ticket est invalide ou déjà utiliser

```
C:\Users\yveezy>ncat localhost 5000
TIC-0025
INVALID Ticket: Already usedSERVER TIMEOUT
Message reçu : TIC-0025 depuis ('127.0.0.1', 7189)
Ticket ID extrait: 0025
Ticket ID complet pour la recherche : TIC-0025
Ticket trouvé : ('TIC-0025', 'U')
INVALID Ticket: Already used
Réponse envoyée : INVALID Ticket: Already used
```

```

C:\Users\yveezy>ncat localhost 5000
TIC-1001
INVALID Ticket: Not found (Ticket ID: TIC-1001)
PS C:\Users\yveezy\Desktop\yvon\S5\BD\projet_1> python serv2.py
Connexion réussie à la base de données.
Serveur TCP démarré sur 127.0.0.1:5000
Connexion établie avec ('127.0.0.1', 51841)
Message reçu : TIC-1001 depuis ('127.0.0.1', 51841)
Ticket ID extrait: 1001
Ticket ID complet pour la recherche : TIC-1001
INVALID Ticket: Not found (Ticket ID: TIC-1001)
Réponse envoyée : INVALID Ticket: Not found (Ticket ID: TIC-1001)

```

Cas où la salle n'est pas valide

```

C:\Users\yveezy>ncat localhost 5000
TIC-0050
VALID Ticket foundMAC-1005
INVALID Salle: Not found (Machine ID: MAC-1005)
PS C:\Users\yveezy\Desktop\yvon\S5\BD\projet_1> python serv2.py
Connexion réussie à la base de données.
Serveur TCP démarré sur 127.0.0.1:5000
Connexion établie avec ('127.0.0.1', 7297)
Message reçu : TIC-0025 depuis ('127.0.0.1', 7297)
Ticket ID extrait: 0025
Ticket ID complet pour la recherche : TIC-0025
Ticket trouvé : ('TIC-0025', 'U')
INVALID Ticket: Already used
Réponse envoyée : INVALID Ticket: Already used
Le délai d'attente du serveur a été dépassé.
Connexion avec ('127.0.0.1', 7297) fermée.
Connexion établie avec ('127.0.0.1', 7345)
Message reçu : TIC-0050 depuis ('127.0.0.1', 7345)
Ticket ID extrait: 0050
Ticket ID complet pour la recherche : TIC-0050
Ticket trouvé : ('TIC-0050', 'N')
VALID Ticket found
Réponse envoyée : VALID Ticket found
Message reçu : MAC-1005 depuis ('127.0.0.1', 7345)
Machine ID extrait: 1005
Machine ID complet pour la recherche : MAC-1005
Réponse envoyée : INVALID Salle: Not found (Machine ID: MAC-1005)

```

Cas où le délai est passé

```

C:\Users\yveezy>ncat localhost 5000
SERVER TIMEOUT
Connexion établie avec ('127.0.0.1', 4066)
Le délai d'attente du serveur a été dépassé.
Connexion avec ('127.0.0.1', 4066) fermée.

```

# ANNEXE

## Code de test de connectivité réseau

Code Serveur :

```
import socket
import psycopg2
import time

# Connexion à La base de données PostgreSQL
try:
    conn = psycopg2.connect(
        host="postgresql-cinemaproject.alwaysdata.net",
        database="cinemaproject_cine",
        user="cinemaproject",
        password="thecinemaproject2",
        port=5432
    )
    cursor = conn.cursor()
    print("Connexion réussie à la base de données.")
except psycopg2.OperationalError as db_err:
    print(f"Erreur de connexion à la base de données : {db_err}")
    exit()

# Paramètres du serveur TCP
HOST = "127.0.0.1"
PORT = 5000

# Création du socket TCP
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((HOST, PORT))
server.listen(5)

print(f"Serveur TCP démarré sur {HOST}:{PORT}")

# Délai d'attente pour Les opérations en secondes
TIMEOUT = 30

while True:
    try:
        # Accepter une connexion entrante
        client_socket, client_addr = server.accept()
        print(f"Connexion établie avec {client_addr}")
```

```

# Définir Le timeout de la connexion
client_socket.settimeout(TIMEOUT)

start_time = time.time()

while True:
    try:
        # Recevoir un message du client
        data = client_socket.recv(4096)
        if not data:
            break # Arrêter si le client ferme la connexion

        message = data.decode("utf-8").strip()
        print(f"Message reçu : {message} depuis {client_addr}")

        response = "UNKNOWN COMMAND" # Réponse par défaut

        # Extraction des identifiants
        if message.startswith("TIC-"):
            ticket_id = f"TIC-{message[4:]}"
            print(f"Ticket ID extrait : {ticket_id}")

            # Rechercher le ticket dans la base de données
            cursor.execute("SELECT id_ticket, used FROM ticket WHERE id_ticket
= %s", (ticket_id,))
            ticket = cursor.fetchone()

            if ticket:
                print(f"Ticket trouvé : {ticket}")
                if ticket[1] == "N": # Valide
                    response = "VALID Ticket"
                else:
                    response = "INVALID Ticket: Already used"
            else:
                response = "INVALID Ticket: Not found"

        elif message.startswith("MAC-"):
            machine_id = f"MAC-{message[4:]}"
            print(f"Machine ID extrait : {machine_id}")

            # Rechercher la salle via la machine
            cursor.execute("""
                SELECT s.id_salle, s.nom_s
                FROM machine m
                JOIN salle s ON m.id_salle = s.id_salle
                WHERE m.id_machine = %s;
            """, (machine_id,))
            salle = cursor.fetchone()

```



```

        if salle:
            print(f"Salle trouvée : {salle}")
            response = "VALID Salle"
        else:
            response = "INVALID Salle: Not found"

    # Mise à jour conditionnelle
    if ticket and ticket[1] == "N" and salle:
        cursor.execute(
            "UPDATE ticket SET used = 'U' WHERE id_ticket = %s RETURNING
id_ticket",
            (ticket_id,)
        )
        updated_ticket = cursor.fetchone()
        if updated_ticket:
            conn.commit()
            response = f"VALID Ticket {updated_ticket[0]} updated"
        else:
            response = "INVALID Ticket: Update failed"
    elif not ticket or ticket[1] != "N":
        response = "INVALID Ticket: Not valid for update"
    elif not salle:
        response = "INVALID Salle: Not valid for update"

    # Envoyer la réponse au client
    client_socket.sendall(response.encode("utf-8"))
    print(f"Réponse envoyée : {response}")

    # Vérification du délai
    if time.time() - start_time > TIMEOUT:
        print("Temps d'attente dépassé, fermeture de la connexion...")
        client_socket.close()
        break

except socket.timeout:
    print("Délai d'attente dépassé.")
    client_socket.sendall("SERVER TIMEOUT".encode("utf-8"))
    client_socket.close()
    break

client_socket.close()
print(f"Connexion avec {client_addr} fermée.")

except psycopg2.Error as db_error:
    print(f"Erreur liée à la base de données : {db_error}")

except Exception as e:

```

```
print(f"Erreur : {e}")
```

```
# Fermer le serveur après un certain délai ou une condition  
server.close()
```

#### Code Client:

```
import java.io.*;
import java.net.*;
import java.util.Scanner;

public class ClientTCP {
    private static final String SERVER_HOST = "127.0.0.1"; // Adresse IP du serveur
    private static final int SERVER_PORT = 5000;           // Port utilisé par le serveur

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("== Client pour serveur de validation ==");
        System.out.println("1. Vérifier un ticket (TIC-XXXX)");
        System.out.println("2. Vérifier une machine (MAC-XXXX)");
        System.out.println("3. Quitter");

        while (true) {
            System.out.print("\nChoisissez une option : ");
            String choix = scanner.nextLine();

            String commande = null;
            switch (choix) {
                case "1":
                    System.out.print("Entrez l'ID du ticket (sans 'TIC-') : ");
                    String ticketId = scanner.nextLine();
                    commande = "TIC-" + ticketId;
                    break;
                case "2":
                    System.out.print("Entrez l'ID de la machine (sans 'MAC-') : ");
                    String machineId = scanner.nextLine();
                    commande = "MAC-" + machineId;
                    break;
                case "3":
                    System.out.println("Fermeture du client.");
                    scanner.close();
                    return;
                default:
                    System.out.println("Option invalide. Veuillez réessayer.");
                    continue;
            }

            if (commande != null) {
                envoyerCommande(commande);
            }
        }
    }
}
```

```

private static void envoyerCommande(String commande) {
    try (Socket socket = new Socket(SERVER_HOST, SERVER_PORT);
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
        BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream())) {

        System.out.println("Connecté au serveur " + SERVER_HOST + ":" + SERVER_PORT);

        // Envoi de la commande au serveur
        out.println(commande);
        System.out.println("Commande envoyée : " + commande);

        // Réception de la réponse du serveur
        String response = in.readLine();
        System.out.println("Réponse du serveur : " + response);







    } catch (IOException e) {
        System.err.println("Erreur lors de la communication avec le serveur : " +
e.getMessage());
    }
}

```

## Requêtes DML (en SQL)

-- Sélection des informations sur les séances d'un film particulier qui ne sont pas en cours.

```
-- Remplacez 'F-00000001' inception.
SELECT
    seance.id_seance,
    seance.h_debut,
    seance.h_fin,
    seance.prix,
    seance.langue,
    seance.st AS sous_titres,
    seance.play AS type_projection,
    salle.nom_s AS nom_salle,
    salle.type_proj AS type_projection_salle,
    cinema.nom_cin AS nom_cinema,
    cinema.adresse AS adresse_cinema
FROM
    seance
JOIN
    salle ON seance.id_seance = salle.id_seance
JOIN
    cinema ON salle.id_cinema = cinema.id_cinema
WHERE
    seance.id_film = 'F-00000001'
    AND NOT (CURRENT_TIMESTAMP BETWEEN seance.h_debut AND seance.h_fin);
```

	id_seance 	h_debut 	h_fin 	prix 	langue 	sous_titres 	type_projection
	character (7)	timestamp without time zone	timestamp without time zone	double precision	character (2)	boolean	character (1)
1	SEA0001	2024-11-26 15:00:00	2024-11-26 17:28:00	12.5	EN	true	F





-- Sélection de tous les billets en joignant naturellement les tables Siege, Salle et Cinema, filtrant les résultats pour inclure uniquement les billets avec un statut d'utilisation en 'Utilisé'

```
SELECT
    ticket.id_ticket,
    ticket.date_crea,
    ticket.date_expir,
    ticket.used AS statut_utilisation,
    siege.id_siege,
    siege.num_place,
    siege.num_range,
    salle.id_salle,
    salle.nom_s AS nom_salle,
    cinema.id_cinema,
    cinema.nom_cin AS nom_cinema
FROM
    ticket
NATURAL JOIN siege
NATURAL JOIN salle
NATURAL JOIN cinema
WHERE
    ticket.used = 'U';
```

	id_ticket character (8) 🔒	date_crea timestamp without time zone 🔒	date_expir timestamp without time zone 🔒	statut_utilisation character (1) 🔒
1	TIC-1003	2024-11-25 10:00:00	2025-11-25 10:00:00	U
2	TIC-1004	2024-11-25 12:00:00	2025-11-25 12:00:00	U

-- Sélection des identifiants des sièges disponibles (statut = 'Libre') pour une séance spécifique.

```
SELECT
    siege.id_siege,
    siege.num_place,
    siege.num_range,
    siege.type_p AS type_place
FROM
    siege
JOIN
    salle ON siege.id_salle = salle.id_salle
JOIN
    seance ON salle.id_seance = seance.id_seance
WHERE
    seance.id_seance = 'SEA0001'
    AND siege.st = 'F';    --F=Free=Libre
```

	id_siege [PK] character (7) 	num_place integer 	num_range integer 	type_place character (1) 
1	SIE0002	2	1	V

-- Sélection des détails des billets achetés par un client spécifique, incluant l'identifiant du billet, le titre du film, le nom du cinéma, le nom de la salle, l'heure de projection, le numéro du siège et le statut d'utilisation du billet.


```
SELECT
    ticket.id_ticket,
    film.titre AS titre_film,
    cinema.nom_cin AS nom_cinema,
    salle.nom_s AS nom_salle,
    seance.h_debut AS heure_projection,
    siege.num_place AS numero_place,
    siege.num_range AS numero_range,
    ticket.used AS statut_utilisation
FROM
    ticket
JOIN
    siege ON ticket.id_siege = siege.id_siege
JOIN
    salle ON siege.id_salle = salle.id_salle
JOIN
    seance ON salle.id_seance = seance.id_seance
JOIN
    film ON seance.id_film = film.id_film
JOIN
    cinema ON salle.id_cinema = cinema.id_cinema
WHERE
    ticket.id_client = 'CLI-1001';
```

	id_ticket character (8) 🔒	titre_film character varying (50) 🔒	nom_cinema character varying (100) 🔒	nom_salle character varying (50) 🔒	heure_projection timestamp without time zone
1	TIC-1001	Inception	UGC Ciné Cité	Salle 1	2024-11-26 15:00:00
2	TIC-1004	Inception	UGC Ciné Cité	Salle 1	2024-11-26 15:00:00






-- Sélection du statut d'utilisation et de la date d'expiration d'un ticket spécifique.

```
SELECT
    used AS statut_utilisation,
    date_expir AS date_expiration
FROM
    ticket
WHERE
    id_ticket = 'TIC-1001';
```

	statut_utilisation character (1) 	date_expiration timestamp without time zone 
1	N	2025-11-25 10:00:00

--Permet de récupérer le chiffre d'affaire total de chaque cinémas et classé par ordre décroissant

```
SELECT
    cinema.id_cinema,
    cinema.nom_cin AS nom_cinema,
    SUM(seance.prix) AS chiffre_affaire_total
FROM
    cinema
JOIN
    salle ON cinema.id_cinema = salle.id_cinema
JOIN
    seance ON salle.id_seance = seance.id_seance
GROUP BY
    cinema.id_cinema, cinema.nom_cin
ORDER BY
    chiffre_affaire_total DESC;
```

	id_cinema [PK] character varying (8) 	nom_cinema character varying (100) 	chiffre_affaire_total double precision 
1	CIN-7501	UGC Ciné Cité	47.5
2	CIN-7502	Pathé Bellecour	32.5
3	CIN-0003	MK2 Bibliothèque	10

--Permet de récupérer les clients les plus fidèles (ceux qui ont dépensé le plus) et trie par ordre décroissant sur notre site nous récupérons seulement le top 5

```
SELECT
    client.id_client,
    personne.nom AS nom_client,
    personne.prenom AS prenom_client,
    SUM(seance.prix) AS montant_total_depense
FROM
    ticket
JOIN
    client ON ticket.id_client = client.id_client
JOIN
    personne ON client.id_personne = personne.id_personne
JOIN
    siege ON ticket.id_siege = siege.id_siege
JOIN
    salle ON siege.id_salle = salle.id_salle
JOIN
    seance ON salle.id_seance = seance.id_seance
GROUP BY
    client.id_client, personne.nom, personne.prenom
ORDER BY
    montant_total_depense DESC
LIMIT 5;
```

	id_client character (8) 🔒	nom_client character varying (50) 🔒	prenom_client character varying (50) 🔒	montant_total_depense double precision 🔒
1	CLI-1001	Lemoine	Alice	25
2	CLI-1002	Dupont	Jean	22.5

--Permet de savoir pour un client donné, combien d'argent il a dépensé au total sur notre site internet

```
SELECT
    client.id_client,
    personne.nom AS nom_client,
    personne.prenom AS prenom_client,
    SUM(seance.prix) AS total_depense
FROM
    ticket
JOIN
    client ON ticket.id_client = client.id_client
JOIN
    personne ON client.id_personne = personne.id_personne
JOIN
    siege ON ticket.id_siege = siege.id_siege
JOIN
    salle ON siege.id_salle = salle.id_salle
JOIN
    seance ON salle.id_seance = seance.id_seance
WHERE
    client.id_client = 'CLI-1001' -- Remplacez par l'ID du client spécifique
GROUP BY
    client.id_client, personne.nom, personne.prenom;
```

	id_client character (8) 🔒	nom_client character varying (50) 🔒	prenom_client character varying (50) 🔒	total_depense double precision 🔒
1	CLI-1001	Lemoine	Alice	25

--Lister les clients avec leur niveau (niveau normal ou VIP)

```
SELECT p.nom, p.prenom, c.lvl, c.date_insc
FROM client c
JOIN personne p ON c.id_personne = p.id_personne;
```

	nom character varying (50) 🔒	prenom character varying (50) 🔒	lvl character (1) 🔒	date_insc date 🔒
1	Lemoine	Alice	N	2024-01-01
2	Dupont	Jean	V	2024-01-02

--Liste des films joués par un acteur spécifique

```
SELECT f.titre, f.date_sortie
FROM film f
JOIN jouer j ON f.id_film = j.id_film
WHERE j.id_acteur = 'ACT-1001';
```

	titre character varying (50) 🔒	date_sortie date 🔒
1	Inception	2010-07-16
2	Titanic	1997-12-19
3	Amélie	2001-04-25
4	Interstellar	2014-11-07
5	La Haine	1995-05-31
6	Joker	2019-10-04
7	The Dark Knight	2008-07-18

--Récupérer les employés ayant un niveau d'accès "Master" dans un cinéma spécifique

```
SELECT p.nom, p.prenom, e.poste
FROM employee e
JOIN personne p ON e.id_personne = p.id_personne
WHERE e.id_cinema = 'CIN-7501' AND e.lvl_acces = 'M';
```

	nom character varying (50) 🔒	prenom character varying (50) 🔒	poste character varying (50) 🔒
1	Girard	Nicolas	Manager

# SITE-WEB

## L'objectif et les fonctionnalités du site

Le site web **CineMAX** représente la composante visuelle de ce projet, conçue spécialement pour les exploitants de cinémas et les utilisateurs. Pour les utilisateurs qui souhaitent utiliser notre site vous pouvez créer un compte utilisateur dont votre adresse-email et mot de passe sera sauvegarder si vous souhaité utiliser une nouvelle fois le site.

Il vise principalement à remplir deux fonctions distinctes : l'une dédiée aux utilisateurs, l'autre orientée vers les gestionnaires.

Pour les utilisateurs, **CineMAX** offre les fonctionnalités suivantes :

1. Un aperçu des films actuellement à l'affiche.
- 2.Un aperçu des séances actuelles à l’affiche
- 3.De réserver une séance et la place voulus
- 4.Un aperçu des réservations effectuée

Pour les gestionnaires, **CineMAX** propose les fonctionnalités suivantes

- 1.Un aperçu des films actuellement à l’affiche et les séances disponibles
- 2.La gestion des utilisateurs
- 3.La gestion des films (ajouter de nouveau film ou retirer)