

# Université de Cergy-Pontoise

## RAPPORT

pour le projet UE BD/Réseaux  
**Licence d'Informatique troisième année**

sur le sujet

Gestion des tickets d'un événement

Rédigé par  
**OFFIONG Dara, BOUKILI Inas, LAMARI Azzeddine**



Version	Date	Modifications	Commentaire
1	14/09/2024	0	0
2	23/09/2024	2	Changement de sujet, définition d'un nouveau contexte
3	27/09/2024	4	Ajout de dictionnaire de données, MLD, MCD, diagrammes applicatifs
4	12/10/2024	3	Ajout du jeu de données, modification du dictionnaire de données, schéma applicatif
5	25/10/2024	5	Modif jeu de données, dictionnaire, MCD, MLD, codes
6	23/11/2024	4	Modif jeu de données, dictionnaire, MCD, schéma applicatif

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Contexte du projet . . . . .	3
<b>2</b>	<b>Gestion du projet</b>	<b>3</b>
2.1	Planning . . . . .	3
<b>3</b>	<b>Ebauche du dictionnaire de données</b>	<b>4</b>
3.1	Administrateur . . . . .	4
3.2	AdminDiscipline . . . . .	5
3.3	AdminEvenement . . . . .	5
3.4	Discipline . . . . .	5
3.5	Evenement . . . . .	5
3.6	Spectateur . . . . .	6
3.7	Ticket . . . . .	6
3.8	Scanner . . . . .	6
3.9	Ticket valide . . . . .	6
3.10	Place . . . . .	6
3.11	Zone . . . . .	7
<b>4</b>	<b>Schéma E/A (MCD)</b>	<b>7</b>
<b>5</b>	<b>Schéma relationnel (MLD)</b>	<b>9</b>
<b>6</b>	<b>Jeu de données</b>	<b>10</b>
<b>7</b>	<b>Diagrammes applicatifs</b>	<b>14</b>
7.1	Scénario 1 : Scan d'un ticket valide . . . . .	14
7.2	Scénario 2 : Scan d'un ticket expiré . . . . .	15
7.3	Scénario 3 : Zone de ticket invalide . . . . .	16
7.4	Scénario 4 : Ticket déjà scanné . . . . .	17
7.5	Scénario 5 : Scan d'un ticket inexistant . . . . .	18
7.6	Scénario 6 : Accès à un événement non autorisé . . . . .	19
<b>8</b>	<b>Annexe</b>	<b>21</b>
8.1	Code côté client . . . . .	21
8.2	Fichier de configuration . . . . .	25
8.3	Code côté serveur . . . . .	27
8.4	Les principaux tests réseaux mis en oeuvre NCAT avec le code du serveur en python (comme client) . . . . .	33
8.5	Les principaux tests réseaux mis en oeuvre NCAT avec le code du client en C (comme serveur) . . . . .	37
8.6	URL et captures d'écran du site web . . . . .	40
8.7	Les 9 requêtes SQL SELECT utilisées . . . . .	44

## Table des figures

1	Diagramme de Gantt 1 . . . . .	4
2	Diagramme de Gantt 2 . . . . .	4
3	Schéma MCD . . . . .	7
4	Scénario 1 : Scan d'un ticket valide . . . . .	33
5	Scénario 2 : Scan d'un ticket expiré . . . . .	34
6	Scénario 3 : Mauvaise zone de ticket . . . . .	34
7	Scénario 4 : Ticket valide déjà scanné . . . . .	35

8	Scénario 5 : Scan d'un ticket inexistant . . . . .	35
9	Scénario 6 : Accès à un événement non autorisé . . . . .	36
10	Scénario 1 : Scan d'un ticket valide . . . . .	37
11	Scénario 2 : Scan d'un ticket expiré . . . . .	38
12	Scénario 3 : Mauvaise zone de ticket . . . . .	38
13	Scénario 4 : Ticket valide déjà scanné . . . . .	39
14	Scénario 5 : Scan d'un ticket inexistant . . . . .	39
15	Scénario 6 : Accès à un événement non autorisé . . . . .	40
16	Selectionner une discipline . . . . .	41
17	Selectionner un evenement . . . . .	41
18	Choisir l'emplacement . . . . .	42
19	Entrer les détails de paiement . . . . .	43
20	Mail de confirmation . . . . .	44

## Liste des tableaux

1	Planning du projet . . . . .	3
---	------------------------------	---

# 1 Introduction

## 1.1 Contexte du projet

Un système d'accréditation pour les événements sportifs. L'objectif est de scanner les tickets des spectateurs à l'aide de QR codes et de scanners portatifs afin de vérifier les tickets à l'entrée ainsi que dans d'autres points stratégiques de l'événement.

Le système permettra au personnel de la caisse et aux autres membres du personnel équipés de scanners de vérifier qui possède un ticket valide et peut donc assister à l'événement. Il fonctionne également comme un filtre qui élimine les tickets non valables ou déjà utilisés afin d'éviter les utilisations frauduleuses ou aberrantes. Par exemple, si un ticket a déjà été scanné ou s'il est expiré, le système enverra une notification indiquant que l'accès est refusé. De même, si un ticket est scanné dans une zone pour laquelle il n'est pas valide, le personnel recevra un message signalant que l'accès à cette zone est interdit.

Gestion des tickets : La gestion des tickets sera centralisée dans une base de données regroupant tous les tickets vendus pour l'événement concerné. Cette base de données est consultée à chaque scan afin de vérifier la validité des tickets ou d'éliminer ceux qui ne le sont pas. Par exemple, le système pourra reconnaître automatiquement si un ticket a été falsifié ou s'il a déjà été utilisé dans une autre zone.

Gestion des spectateurs : Le système permettra également de vérifier l'identité et le nombre de spectateurs participant à chaque événement sportif, selon la discipline sélectionnée. Par exemple, il sera possible de savoir qu'il y a 10000 personnes pour un match de football spécifique, simplement en comptant les tickets scannés et validés à l'entrée ou à d'autres points de contrôle.

## 2 Gestion du projet

### 2.1 Planning

Semaines	Tâches
37 / 38	<ul style="list-style-type: none"><li>- Choix définitif de la composition de l'équipe</li><li>- Description définitive du contexte du projet</li><li>- Ébauche du dictionnaire de données</li><li>- Réflexion sur les données échangées via le réseau (client / serveur)</li><li>- Début de la modélisation</li></ul>
39 / 40	<ul style="list-style-type: none"><li>- Finalisation du dictionnaire de données</li><li>- Modélisation complète : schéma E/A et schéma relationnel</li><li>- Exemples représentatifs du jeu de données envisagé (tests)</li><li>- Diagramme applicatif du protocole réseau défini</li></ul>
41 / 42	<ul style="list-style-type: none"><li>- Partie DDL du projet</li><li>- Connexions à la base de données depuis le serveur web et le serveur réseau</li></ul>
43 / 45	<ul style="list-style-type: none"><li>- Partie DML du projet</li><li>- Réalisation du client et du serveur réseau</li></ul>
47 / 48	<ul style="list-style-type: none"><li>- Finalisation et tests d'intégration</li></ul>
49	<ul style="list-style-type: none"><li>- Livrables (1/2) : rapport de projet définitif et l'ensemble des codes du projet</li><li>- Livrables (2/2) : vidéos de démonstration du projet</li></ul>
50	<ul style="list-style-type: none"><li>- Soutenance de la partie BD</li><li>- Démonstration de la partie réseau à distance</li></ul>

TABLE 1 – Planning du projet

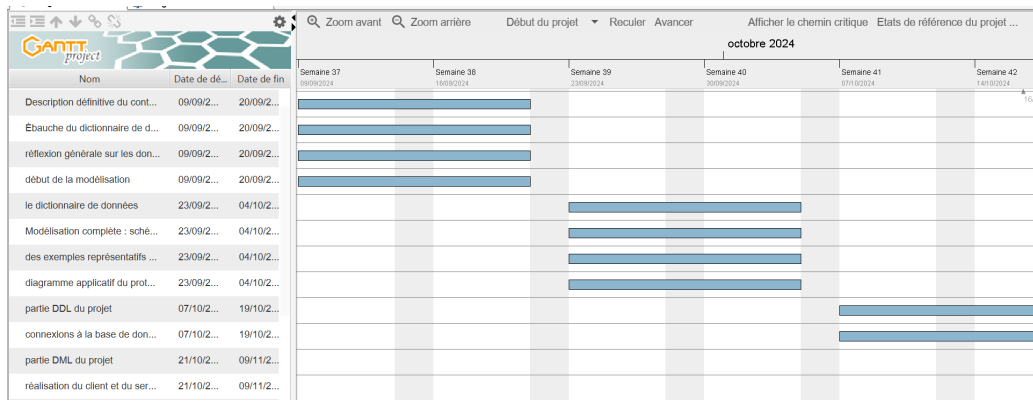


FIGURE 1 – Diagramme de Gantt 1

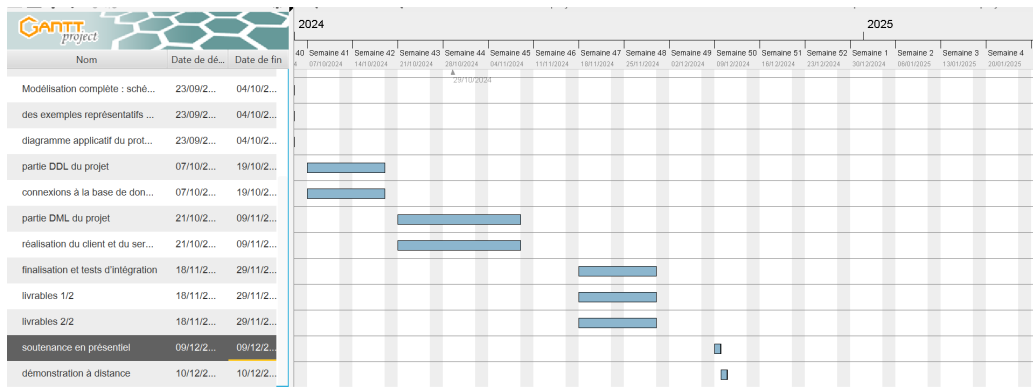


FIGURE 2 – Diagramme de Gantt 2

### 3 Ebauche du dictionnaire de données

#### 3.1 Administrateur

Nom	Type	Null/Not Null	Contraintes	Exemple
IdAdministrateur	VARCHAR(11)	NOT NULL	PK, ALPHANUMERIQUE	ADM00123456
NomAdmin	VARCHAR(30)	NOT NULL	ALPHABETIQUE	Dupont
PrénomAdmin	VARCHAR(30)	NOT NULL	ALPHABETIQUE	Marie
EmailAdmin	VARCHAR(260)	NOT NULL	Unique, contient @	marie.dupont@email.com
TypeAdmin	VARCHAR(60)	NOT NULL	"AdminDiscipline" ou "AdminEvenement"	AdminDiscipline
MotDePasse	TEXT	NOT NULL	UNIQUE, ALPHANUMERIQUE, doit être crypté, doit contenir des caractères spéciaux	gAAAAABnHMx6VVq6PISz lsKvooBvrOgeoj6fudzFa1rT3 Po9SjhDVqDMQ972OkEYF 0zAvCn1AuR9eWbcxRBK pGn - O3C96t2qQ ==

### 3.2 AdminDiscipline

Nom	Type	Null/Not Null	Contraintes	Exemple
IDAdminDisc	VARCHAR(11)	NOT NULL	PK, ALPHANUMERIQUE	ADISC123456
DateDatttributionDisc	Date	NOT NULL	$1975 \leq \text{DateDatttributionDisc} \leq \text{Today}()$	2024-10-12
IDContratDisc	VARCHAR(20)	NOT NULL	ALPHANUMERIQUE	CT-2024-001

### 3.3 AdminEvenement

Nom	Type	Null/Not Null	Contraintes	Exemple
IDAdminEve	VARCHAR(11)	NOT NULL	PK, ALPHANUMERIQUE	AEVE1234567
DateDatttributionEve	Date	NOT NULL	$1975 \leq \text{DateDatttributionEve} \leq \text{Today}()$	2024-10-12
IDContratEve	VARCHAR(20)	NOT NULL	ALPHANUMERIQUE	CT-2024-005

### 3.4 Discipline

Nom	Type	Null/Not Null	Contraintes	Exemple
IDDiscipline	VARCHAR(11)	NOT NULL	PK, ALPHANUMERIQUE	DISC0012345
NomDiscipline	VARCHAR(30)	NOT NULL	Alphabétique	Basketball
LieuDiscipline	VARCHAR(300)	NOT NULL	ALPHANUMERIQUE	Stade Pierre de Coubertin, Paris

### 3.5 Evenement

Nom	Type	Null/Not Null	Contraintes	Exemple
IDEvenement	VARCHAR(11)	NOT NULL	PK, ALPHANUMERIQUE	EVE12345678
NomEvenement	VARCHAR(200)	NOT NULL	ALPHANUMERIQUE	France vs Belgique
DateEvenement	DATE	NOT NULL	$\text{DateEvenement} \geq \text{Today}()$	2024-10-12
HeureEvenement	TIME WITHOUT TIME ZONE	NOT NULL	HH :MM :SS	14 :30 :00
HeureFinEvenement	TIME WITHOUT TIME ZONE	NOT NULL	HH :MM :SS	17 :30 :00
CapaciteMax	INT	NOT NULL	$\text{CapaciteMax} > 0$	100

### 3.6 Spectateur

Nom	Type	Null/Not Null	Contraintes	Exemple
IDSpectateur	VARCHAR(11)	NOT NULL	PK, ALPHANUMERIQUE	SPECT123456
NomSpectateur	VARCHAR(30)	NOT NULL	ALPHABETIQUE	valerie
PrénomSpectateur	VARCHAR(30)	NOT NULL	ALPHABETIQUE	youssra
EmailSpectateur	VARCHAR(260)	NOT NULL	Unique, contient @	youssra.valerie@email.com

### 3.7 Ticket

Nom	Type	Null/Not Null	Contraintes	Exemple
IDTicket	VARCHAR(11)	NOT NULL	PK, Alphanumérique	TICKET12345
DateAchat	DATE	NOT NULL	DateAchat $\leq$ DateEvenement	2024-09-30
PrixTicket	Reel	NOT NULL	PrixTicket $>$ 0	150

### 3.8 Scanner

Nom	Type	Null/Not Null	Contraintes	Exemple
IDScanner	VARCHAR(11)	NOT NULL	PK, Alphanumérique	SCAN1234567
Emplacement	VARCHAR(250)	NOT NULL	Alphanumérique	Entrée principale, Stade de France
ModeleScanner	VARCHAR(50)	NOT NULL	Alphanumérique	Zebra DS2208

### 3.9 Ticket valide

Nom	Type	Null/Not Null	Contraintes	Exemple
IDLogScan	VARCHAR(11)	NOT NULL	PK, Alphanumérique	LOGSCAN1234
idticketvalide	VARCHAR(11)	NOT NULL	Alphanumérique	TCK001
HeureScan	TIMESTAMP WITHOUT TIME ZONE	NOT NULL	YYYY-MM-DD HH :MM :SS	2024-10-26 18 :08 :51

### 3.10 Place

Nom	Type	Null/Not Null	Contraintes	Exemple
IDPlace	VARCHAR(11)	NOT NULL	PK, Alphanumérique	PLACE123456
NumeroPlace	VARCHAR(11)	NOT NULL	Alphanumérique	A1
Typeplace	VARCHAR(20)	NOT NULL	"VIP", "Normale", "Handicapée"	Handicapée

### 3.11 Zone

Nom	Type	Null/Not Null	Contraintes	Exemple
IDZone	VARCHAR(11)	NOT NULL	PK, Alphanumérique	ZONE123456
NomZone	VARCHAR(30)	NOT NULL	Unique, Alphanumérique	Zone A
CapaciteZone	INT	NOT NULL	100000 > CapaciteZone > 0	20000

## 4 Schéma E/A (MCD)

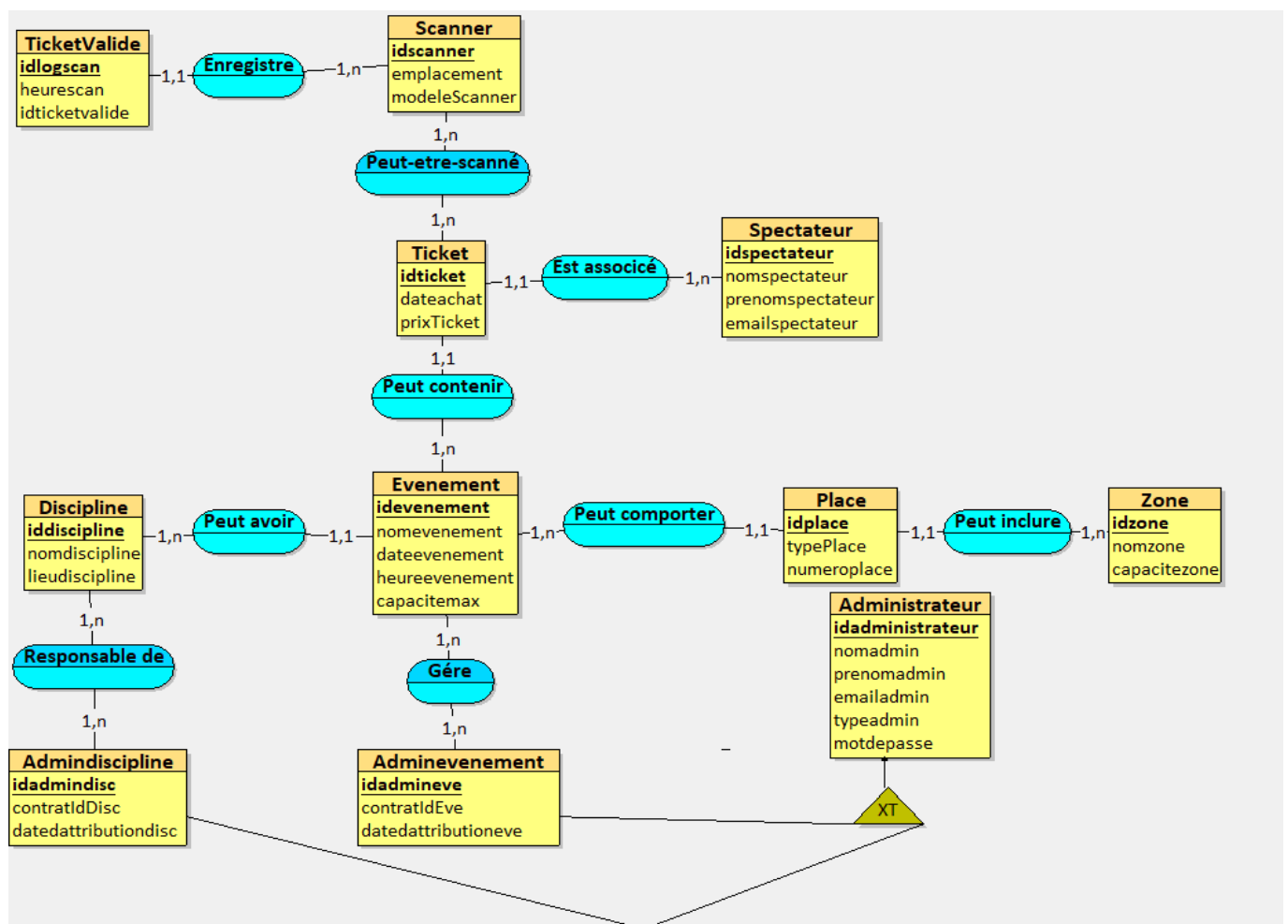


FIGURE 3 – Schéma MCD



## 5 Schéma relationnel (MLD)

### **Le schéma relationnel (MLD)**

Administrateur (**IdAdministrateur**, NomAdmin, PrénomAdmin, EmailAdmin, TypeAdmin, MotDePasse)

AdminDiscipline (**#IDAdminDisc**, ContratIdDisc, DateDattributionDisc)

AdminEvenement(**#IDAdminEve**, ContratIdEve, DateDattributionEve)

Discipline (**IDDiscipline**, NomDiscipline, LieuDiscipline)

Responsable\_de (**#IDAdminDisc**, **#IDDiscipline**)

Evenement (**IDEvenement**, NomEvenement, DateEvenement, HeureEvenement, HeureFinEvenement, CapaciteMax, **#IDDiscipline**)

Gere (**#IDAdminEve**, **#IDEvenement**)

Zone (**IDZone**, NomZone, CapaciteZone)

Place (**IDPlace**, TypePlace, NumeroPlace, **#IDEvenement**, **#IDZone**)

Spectateur (**IDSpectateur**, NomSpectateur, PrénomSpectateur, EmailSpectateur)

Ticket (**IDTicket**, prixTicket, DateAchat, **#IDEvenement**, **#IDSpectateur**)

Scanner (**IDScanner**, ModeleScanner, Emplacement)

Peut\_etre\_scanné ( **#IDTicket**, **#IDScanner**)

Ticket\_valide (**IDLogScan**, HeureScan, id\_ticket\_valide, **#IDScanner**)

## 6 Jeu de données

### Jeu de données

#### Administrateur

<b>IdAdministrateur</b>	<b>NomAdmin</b>	<b>PrénomAdmin</b>	<b>EmailAdmin</b>	<b>TypeAdmin</b>	<b>MotDePasse</b>
ADM001	DUPONT	Jean	jean.dupont@gmail.com	admindiscipline	gAAAAABnH Mx6VVq6PISz lsKvooBvrOge oj6fudzFa1rT 3Po9SjhDVqD MQ972OkEY F_0zAvCn1Au R9eWbcxRBK pGn- O3C96t2qQ= =
ADM002	OFFIONG	Dara	dara.offiang@gmail.com	admindiscipline	gAAAAABnH Mz4TI3AkN8 Mfcs3j17L0m -7_g- YsxL_YXDk0H dNaIFC9eody brj_6wLkrGrf Z6UF5- og9Zm9Qmf VCdtRNwjcs_ NJQ==
ADM003	BOUKILI	Inas	inas.boukili@gmail.com	AdminEvenemen t	gAAAAABnH M1eno3Ewq whdjACO9xu mJzQ17JM9s LhPnt_Yq9T0 moMcSldBZa X5I9LnXg8hj GTvCV- 7NtpW_PQK bYTQU8Jkl5e vw==
ADM004	LAMARI	Azzeddine	Azzeddine.lamari@gmail. com	AdminEvenemen t	gAAAAABnH M2GtGm1W mY96j77eyD d3H9Y_effrNI

					o1eAjRdoG1J 7AxJ9MpjKD hcFukqcVt2V 3TaN- o1KrAMKcnj_ Nu6zWmNzK xw==
--	--	--	--	--	---

#### AdminDiscipline

IDAdminDisc	DateDattributionDisc	IDContratDisc
ADM001	2024-01-15	CT1-2024-02-01
ADM002	2024-01-20	CT2-2024-05-12

#### AdminEvenement

IDAdminEve	DateDattributionEve	IDContratEve
ADM003	2024-02-10	CT2-2024-10-11
ADM004	2024-02-12	CT9-2024-04-10

#### Discipline

IDDiscipline	NomDiscipline	LieuDiscipline
DISC001	Football	Stade de France
DISC002	Basketball	Accor Arena
DISC003	Tennis	Stade Roland-Garros

#### Responsable de

IDAdminDisc	IDDiscipline
ADM001	DISC001
ADM002	DISC002
ADM002	DISC003

### Evenement

IDEvenement	NomEvenement	DateEvenement	HeureEvenement	HeureFinEvenement	CapaciteMax	IDDiscipline
EVE001	France vs Belgique	2024-11-15	19:00:00	21:00:00	70000	DISC001
EVE002	Chicago Bulls vs Detroit Pistons	2024-10-15	21:00:00	23:00:00	20000	DISC002
EVE003	Djokovic vs Nadal	2024-10-23	19:30:00	23:00:00	15000	DISC003

### Gere

IDAdminEve	IDEvenement
ADM003	EVE001
ADM004	EVE002
ADM004	EVE003

### Zone

IDZone	NomZone	CapaciteZone
Z001	Stade de France Tribune A	20000
Z002	Accor Arena Tribune VIP	3000
Z003	Stade Roland-Garros Tribune C	1000

### Place

IDPlace	NumeroPlace	IDEvenement	IDZone	Typeplace
PLC001	A1	EVE001	Z001	VIP
PLC002	A2	EVE002	Z002	Normale
PLC003	A3	EVE003	Z003	Handicapée

### Spectateur

IDSpectateur	NomSpectateur	PrénomSpectateur	EmailSpectateur
SP001	Durand	Marc	marc.durand@gmail.com
SP002	Leblanc	Sophie	sophie.leblanc@gmail.com
SP003	Owens	Frank	frank.owens@gmail.com

### Ticket

IDTicket	DateAchat	IDEvenement	IDSpectateur	PrixTicket
TCK001	2024-10-30	EVE001	SP001	150
TCK002	2024-09-16	EVE002	SP002	90
TCK003	2024-12-01	EVE003	SP003	120

### Scanner

IDScanner	Emplacement	ModeleScanner
SCN001	Stade de France Tribune A	Zebra DS2208
SCN002	Accor Arena Tribune VIP	Ingenico iPP350
SCN003	Stade Roland-Garros Tribune C	Honeywell Voyager 1202g

### Peut etre scanné

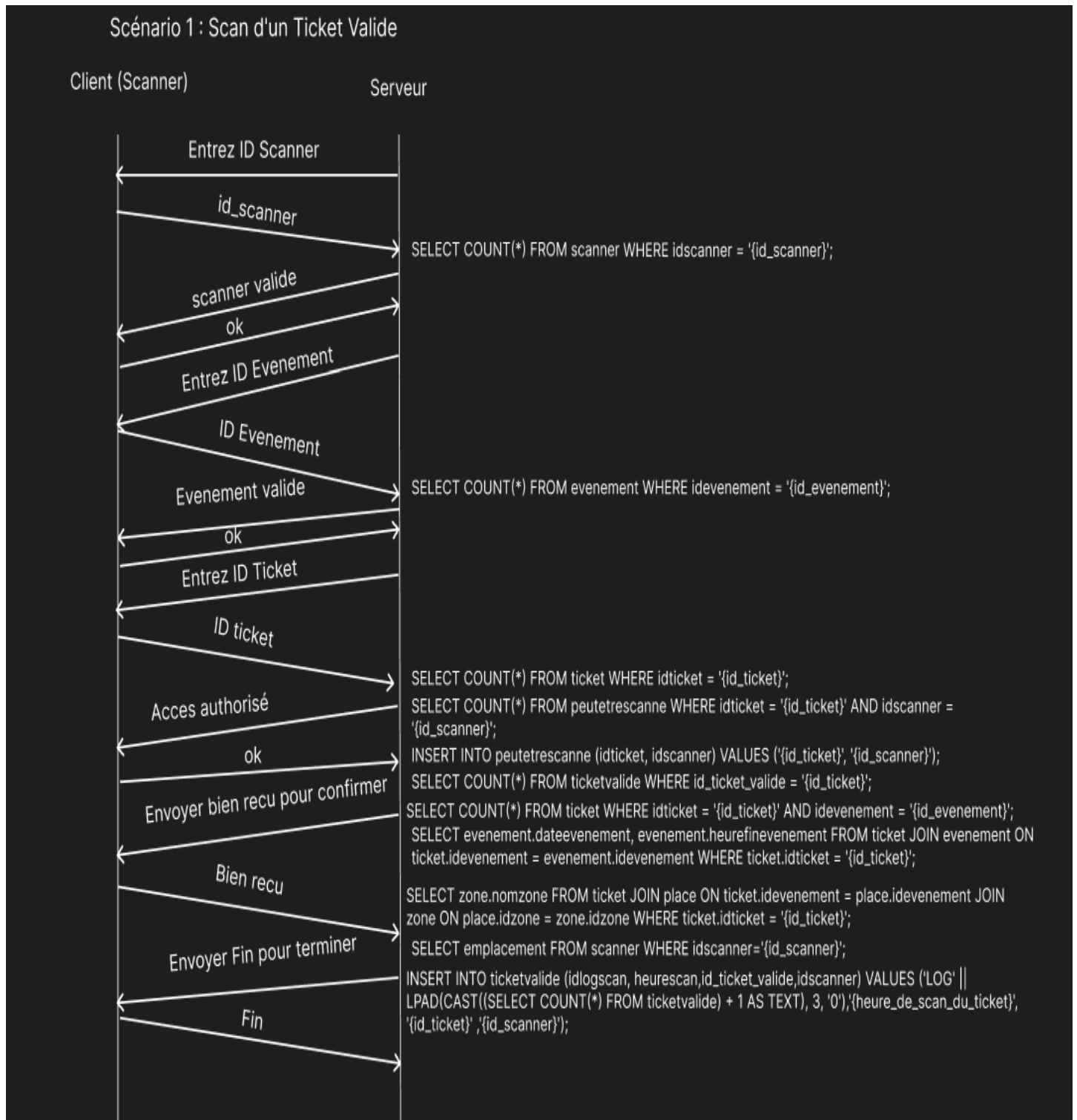
IDTicket	IDScanner
TCK001	SCN001
TCK003	SCN003

### Ticket valide

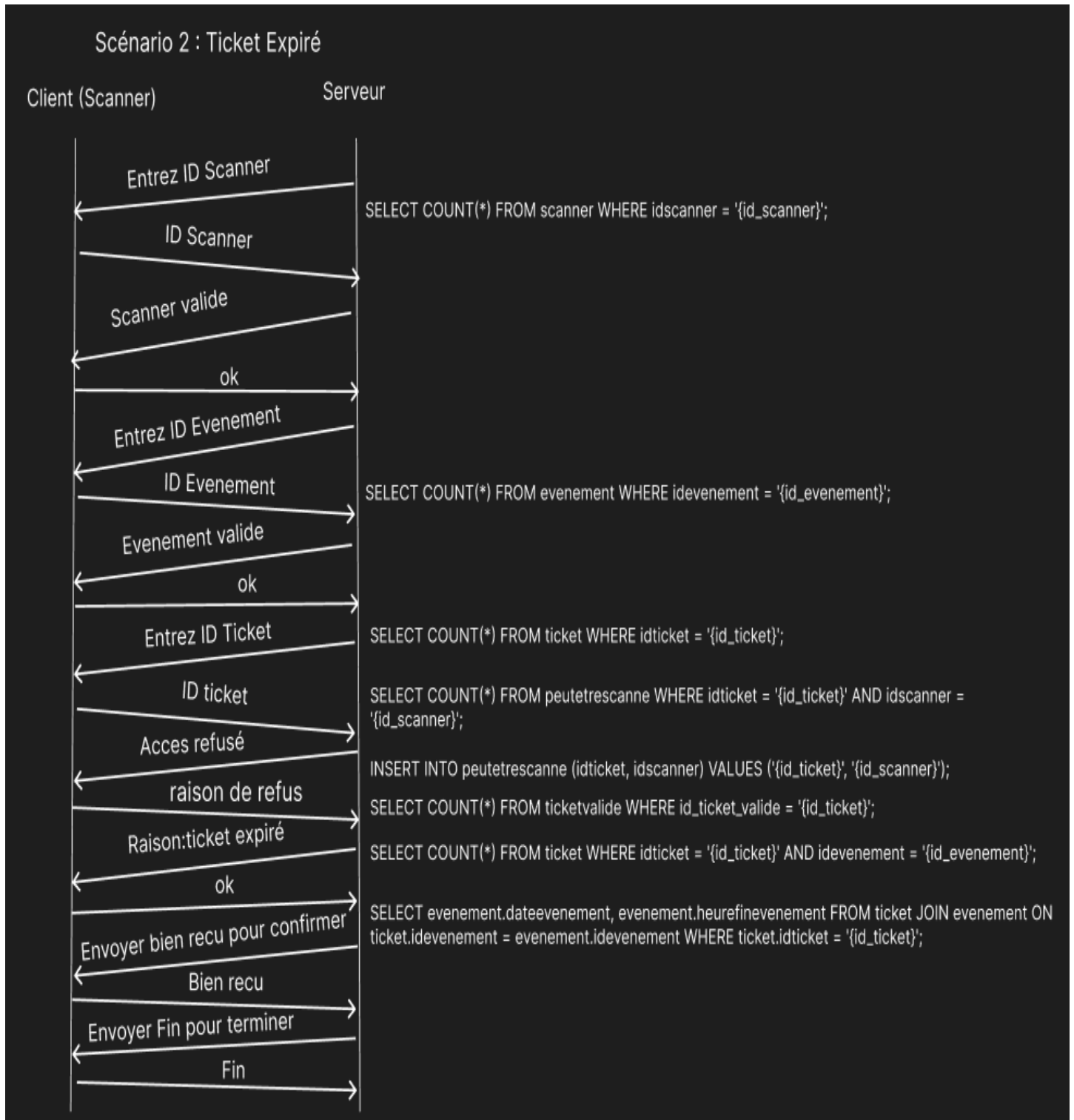
IDLogScan	HeureScan	Id_ticket_valide	IDScanner
LOG001	2024-10-27 14:54:54	TCK003	SCN003=

## 7 Diagrammes applicatifs

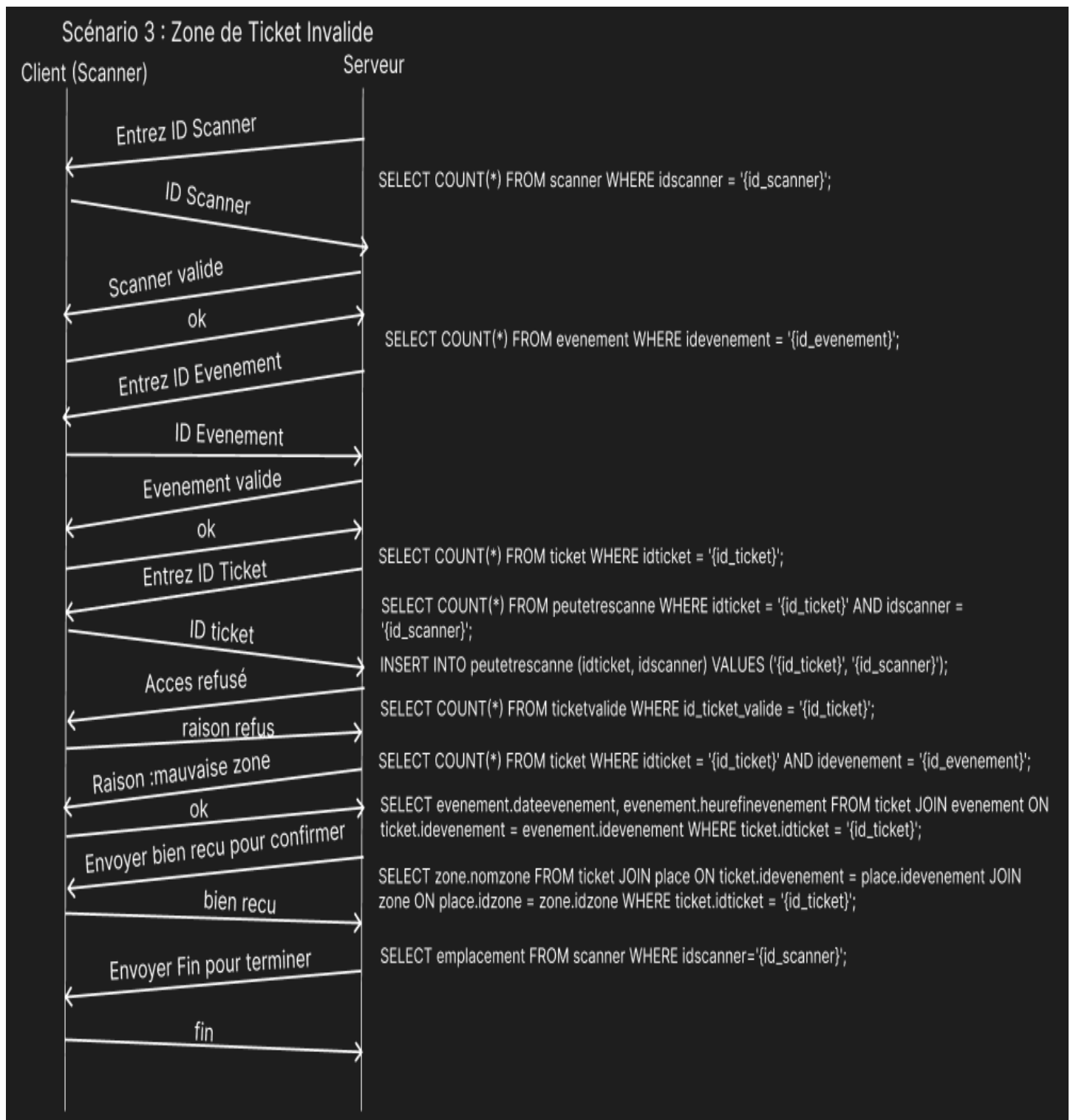
### 7.1 Scénario 1 : Scan d'un ticket valide



## 7.2 Scénario 2 : Scan d'un ticket expiré

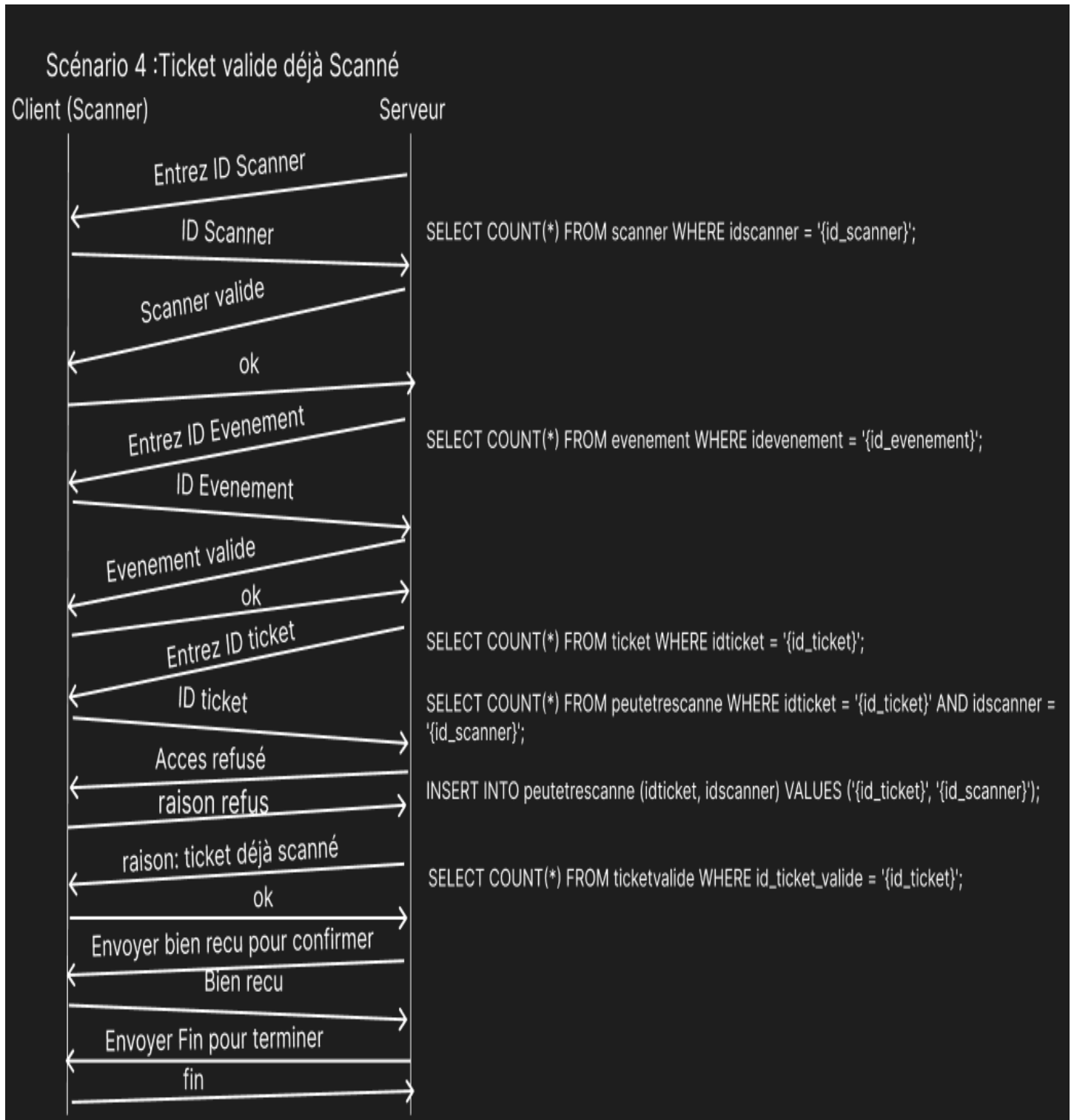


### 7.3 Scénario 3 : Zone de ticket invalide

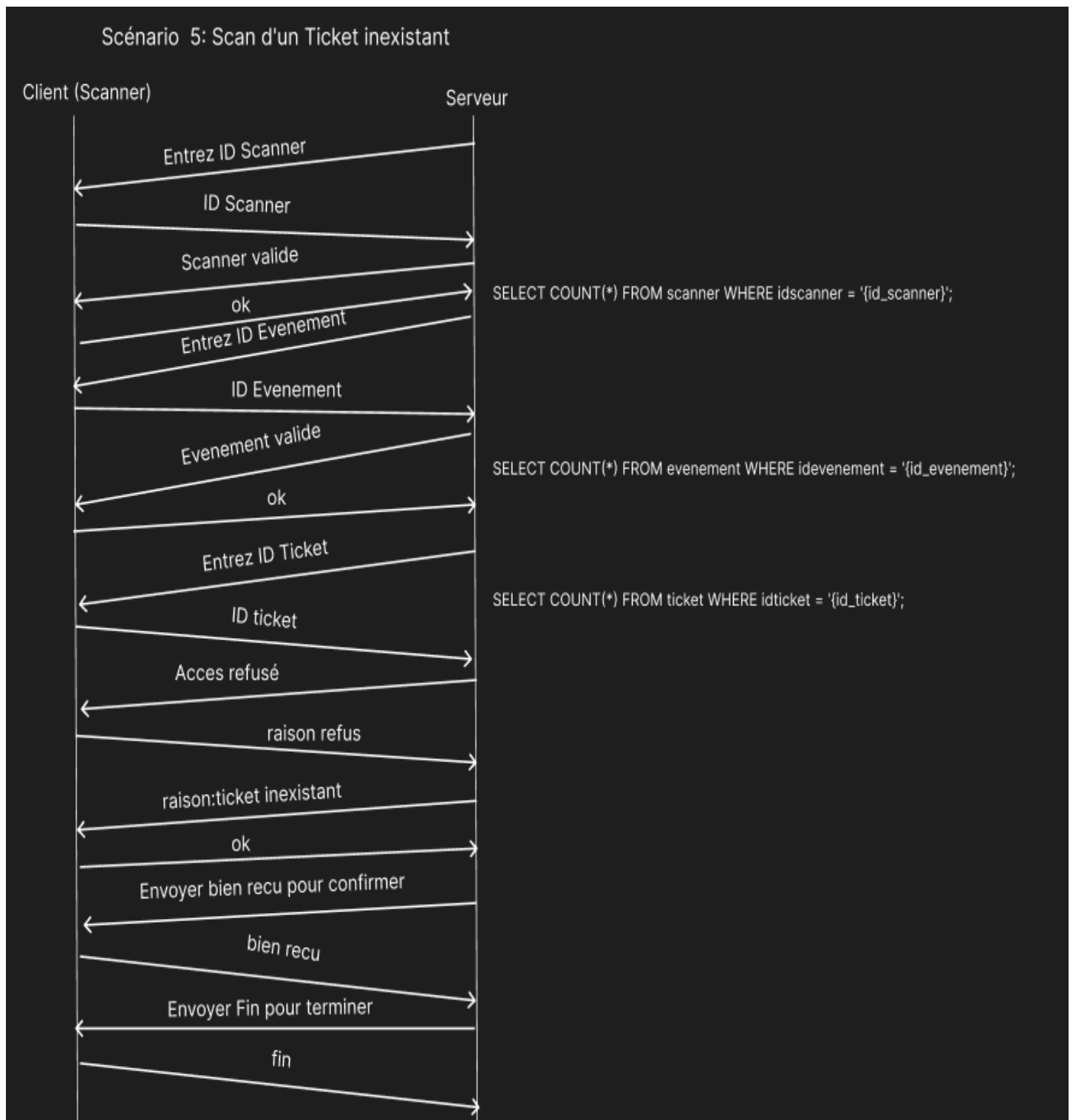




#### 7.4 Scénario 4 : Ticket déjà scanné

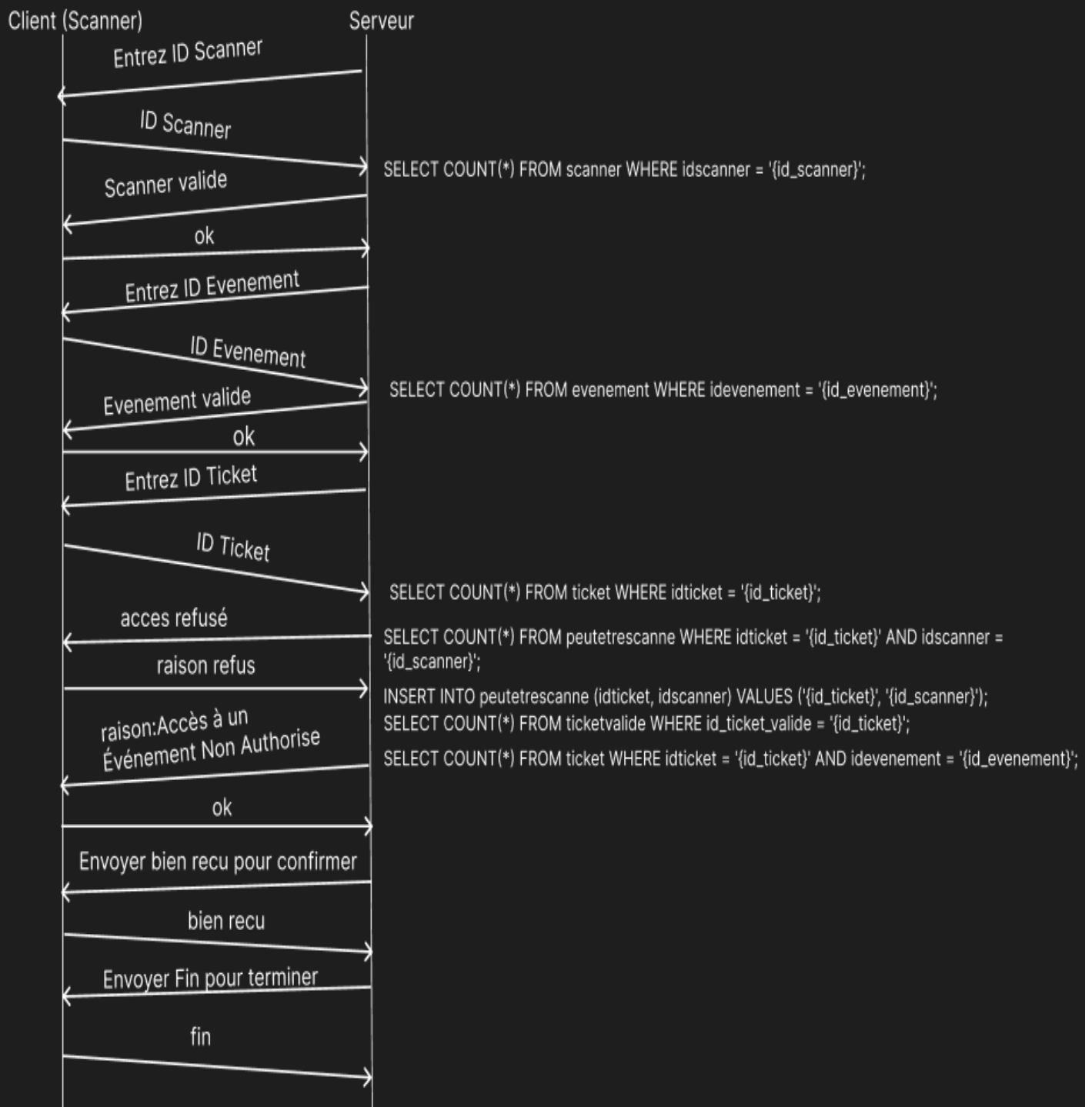


## 7.5 Scénario 5 : Scan d'un ticket inexistant



## 7.6 Scénario 6 : Accès à un événement non autorisé

### Scénario 6: Accès à un Événement Non Autorisé



## 8 Annexe

### 8.1 Code côté client

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 #include <winsock2.h>
5
6 #pragma comment(lib, "ws2_32.lib") // Necessaire pour les
   sockets sous Windows
7
8 int main(int argc, char *argv[]) {
9     WSADATA wsa;
10    SOCKET socket_desc;
11    struct sockaddr_in server;
12    char message[2000], server_reply[2000];
13    int recv_size;
14
15    // Initialisation de Winsock
16    printf("Initialising Winsock...\n");
17    if (WSAStartup(MAKEWORD(2,2), &wsa) != 0) {
18        printf("Failed. Error Code : %d", WSAGetLastError());
19        ;
20        return 1;
21    }
22    printf("Initialised.\n");
23
24    // Creation du socket
25    if ((socket_desc = socket(AF_INET, SOCK_STREAM, 0)) ==
26        INVALID_SOCKET) {
27        printf("Could not create socket : %d",
28            WSAGetLastError());
29        return 1;
30    }
31    printf("Socket created.\n");
32
33    server.sin_addr.s_addr = inet_addr("127.0.0.1"); //
34    Adresse IP du serveur
35    server.sin_family = AF_INET;
36    server.sin_port = htons(12345); // Port du serveur
37
38    // Connexion au serveur distant
39    if (connect(socket_desc, (struct sockaddr *)&server,
40        sizeof(server)) < 0) {
41        printf("Connect error");
42        return 1;
43    }
44    printf("Connected\n");
45
46    // Boucle principale pour les changes avec le serveur
47    while (1) {
48        // Reception des messages du serveur
```

```

44     if ((recv_size = recv(socket_desc, server_reply,
45         sizeof(server_reply), 0)) == SOCKET_ERROR) {
46         printf("recv failed");
47         break;
48     }
49
50     // Ajout de la fin de chaîne la réponse
51     server_reply[recv_size] = '\0';
52     printf("Server reply : %s\n", server_reply);
53
54     // Demande de l'ID scanner
55     if (strstr(server_reply, "Entrez ID scanner") !=
56         NULL) {
57         printf("Client: ID scanner: ");
58         fgets(message, sizeof(message), stdin);
59         message[strcspn(message, "\n")] = '\0'; //
60         // Enlève le saut de ligne de l'entrée
61         send(socket_desc, message, strlen(message), 0);
62     }
63
64     // Demande de l'ID de l'événement
65     if (strstr(server_reply, "Entrez ID evenement") !=
66         NULL) {
67         printf("Client: ID evenement: ");
68         fgets(message, sizeof(message), stdin);
69         message[strcspn(message, "\n")] = '\0'; //
70         // Enlève le saut de ligne de l'entrée
71         send(socket_desc, message, strlen(message), 0);
72     }
73
74     // Demande de l'ID ticket
75     if (strstr(server_reply, "Entrez ID ticket") != NULL
76     ) {
77         printf("Client: ID ticket: ");
78         fgets(message, sizeof(message), stdin);
79         message[strcspn(message, "\n")] = '\0'; //
80         // Enlève le saut de ligne de l'entrée
81         send(socket_desc, message, strlen(message), 0);
82     }
83
84     // Demande de confirmation
85     if (strstr(server_reply, "Envoyez 'bien reçu' pour
86         confirmer") != NULL) {
87         printf("Client: confirmation: ");
88         fgets(message, sizeof(message), stdin);
89         message[strcspn(message, "\n")] = '\0'; //
90         // Enlève le saut de ligne de l'entrée
91         send(socket_desc, message, strlen(message), 0);
92     }
93
94     // Attente de la réponse du serveur après

```

```

85         confirmation
86         if((recv_size=recv(socket_desc, server_reply,
87             sizeof(server_reply), 0)) != SOCKET_ERROR) {
88             printf("recv failed");
89             break;
90         }
91         server_reply[recv_size] = '\0';
92         printf("Server reply : %s\n", server_reply);
93     }
94     // Demande d'envoyer 'fin' pour terminer la session
95     if (strstr(server_reply, "Envoyez 'fin' pour
96         terminer") != NULL) {
97         printf("Client: 'fin' pour terminer: ");
98         fgets(message, sizeof(message), stdin);
99         message[strcspn(message, "\n")] = '\0'; //
100             Enlève le saut de ligne de l'entrée
101         send(socket_desc, message, strlen(message), 0);
102
103         if (strcmp(message, "fin") == 0) {
104             printf("Session terminée.\n");
105             break; // Sortie de la boucle, fin du
106                 programme
107         }
108     }
109 }
110
111 // Fermeture du socket et nettoyage
112 closesocket(socket_desc);
113 WSACleanup();
114
115 return 0;
116 }

```

## 8.2 Fichier de configuration

```
1 [postgresql]
2 host=localhost
3 database=bd_reseau
4 user=postgres
5 password=gAAAAABnHkg9ocUC7TuXKbKJU30f07PwJ0U9_feq
6 0mLS0CWmUdwcgTC97hmMd05ksFSylXwuusYAmlGq6Zn0oRqMVYEUPXkimA==
7 port=5432
```

### 8.3 Code côté serveur

```
1 import socket
2 from datetime import datetime
3 import configparser
4 import psycopg2
5
6 # Lire le fichier de configuration
7 config = configparser.ConfigParser()
8 config.read(r'C:\Users\dell-5320\Desktop\
    codes_client_serveur\codes_client_serveur\serveur\config.
    ini')
9
10 # Obtenir les informations de connexion
11 db_config = config['postgresql']
12
13 # Se connecter la base de données PostgreSQL
14 try:
15     connection = psycopg2.connect(
16         host=db_config['host'],
17         database=db_config['database'],
18         user=db_config['user'],
19         password=db_config['password'],
20         port=db_config['port']
21     )
22
23     # Créer un curseur
24     cursor = connection.cursor()
25
26     # Fonction pour vérifier si un ticket est expiré
27     def est_ticket_expire(id_ticket):
28
29         # Exécuter une requête (exemple)
30         cursor.execute(f"SELECT evenement.dateevenement,
            evenement.heurefin evenement FROM ticket JOIN
            evenement ON ticket.idevenement = evenement.
            idevenement WHERE ticket.idticket = '{id_ticket}';")
31
32         # Récupérer le résultat
33         result_requete6 = cursor.fetchone() # Utilisez
            fetchone() pour un seul résultat
34
35         date_evenement = result_requete6[0] # Stocker la
            date de l'événement dans une variable
36         heure_fin_evenement = result_requete6[1] # Stocker
            l'heure de fin dans une variable
37         print("Date de l'événement :", date_evenement)
38         print("Heure de fin de l'événement :",
            heure_fin_evenement)
39
```



```

40     # Combiner date et heure pour obtenir un datetime
      sans fuseau horaire
41     datetime_fin_evenement = datetime.combine(
      date_evenement, heure_fin_evenement)
42
43     # Obtenir l'heure actuelle
44     now = datetime.now()
45
46     return now > datetime_fin_evenement
47
48     def access_authorized(client_socket):
49         client_socket.sendall("Acces autorise\n\n".encode('
      utf-8'))
50         print("Acces autorise.")
51         client_socket.sendall("Envoyez 'bien reçu' pour
      confirmer".encode('utf-8'))
52         confirmation = client_socket.recv(1024).decode('utf
      -8').strip()
53         if confirmation == "bien reçu":
54             print("Confirmation recue.")
55             client_socket.sendall(b"Envoyez 'fin' pour
      terminer")
56             fin = client_socket.recv(1024).decode('utf-8').
      strip()
57             if fin == "fin":
58                 print("Connexion fermee par le client.")
59                 client_socket.sendall(b"fin, Fermeture
      connexion...\n")
60                 client_socket.close()
61
62     def access_denied(client_socket, reason):
63         client_socket.sendall(f"Acces refuse\nRaison: {
      reason}\n".encode('utf-8'))
64         print(f"Acces refuse: {reason}")
65         client_socket.sendall("Envoyez 'bien reçu' pour
      confirmer".encode('utf-8'))
66         confirmation = client_socket.recv(1024).decode('utf
      -8').strip()
67         if confirmation == "bien reçu":
68             print("Confirmation recue.")
69             client_socket.sendall(b"Envoyez 'fin' pour
      terminer")
70             fin = client_socket.recv(1024).decode('utf-8').
      strip()
71             if fin == "fin":
72                 print("Connexion fermee par le client.")
73                 client_socket.sendall(b"fin, Fermeture
      connexion...\n")
74                 client_socket.close()
75

```

```

76 def handle_client(client_socket):
77     client_socket.sendall("Entrez ID scanner".encode('
       utf-8'))
78     id_scanner = client_socket.recv(1024).decode('utf-8'
       ).strip()
79     print(f"ID scanner reçu : {id_scanner}")
80
81     cursor.execute(f"SELECT COUNT(*) FROM scanner WHERE
       idscanner = '{id_scanner}';")
82     # Récupérer le résultat
83     result_requete1 = cursor.fetchone() # Utilisez
       fetchone() pour un seul résultat
84     # Afficher le résultat
85     print("Nombre de scanners avec cette id :",
       result_requete1[0]) # Le premier élément du
       tuple
86
87     if result_requete1[0]== 0:
88         access_denied(client_socket, "Scanner invalide")
89         return
90     else:
91         client_socket.sendall(b"Scanner valide\n")
92
93     client_socket.sendall("Entrez ID evenement".encode('
       utf-8'))
94     id_evenement = client_socket.recv(1024).decode('utf
       -8').strip()
95     print(f"ID evenement reçu : {id_evenement}")
96
97     cursor.execute(f"SELECT COUNT(*) FROM evenement
       WHERE idevenement = '{id_evenement}';")
98     # Récupérer le résultat
99     result_requete2 = cursor.fetchone() # Utilisez
       fetchone() pour un seul résultat
100    # Afficher le résultat
101    print("Nombre d'événement avec cette id :",
       result_requete2[0]) # Le premier élément du
       tuple
102    if result_requete2[0]== 0:
103        access_denied(client_socket, "Evenement invalide
       ")
104        return
105    else:
106        client_socket.sendall(b"Evenement valide\n")
107
108    client_socket.sendall("Entrez ID ticket".encode('utf
       -8'))
109    id_ticket = client_socket.recv(1024).decode('utf-8')
       .strip()
110    print(f"ID ticket reçu : {id_ticket}")

```

```

111
112
113
114         cursor.execute(f"SELECT COUNT(*) FROM ticket WHERE
            idticket = '{id_ticket}';")
115         # Récupérer le résultat
116         result_requete3 = cursor.fetchone() # Utilisez
            fetchone() pour un seul résultat
117         # Afficher le résultat
118         print("Nombre de ticket avec cette id :",
            result_requete3[0]) # Le premier élément du
            tuple
119         if result_requete3[0] == 0:
120             access_denied(client_socket, "Ticket inexistant
                ")
121             return
122
123         # insérer idticket et idscanner à la table peut être
            scannée
124
125         # Vérifier si l'idticket et idscanner existent
            déjà dans la table
126         cursor.execute(f"SELECT COUNT(*) FROM peutêtrescannée
            WHERE idticket = '{id_ticket}' AND idscanner =
            '{id_scanner}';")
127         result = cursor.fetchone()
128
129         if result[0] == 0: # Si le résultat est 0, cela
            signifie que la combinaison n'existe pas
130             # Insérer idticket et idscanner dans la table
            peutêtrescannée
131             cursor.execute(f"INSERT INTO peutêtrescannée (
                idticket, idscanner) VALUES ('{id_ticket}',
                '{id_scanner}');")
132             connection.commit() # Valider l'insertion
133             print("Insertion réussie.")
134         else:
135             print("La combinaison idticket et idscanner
                existe déjà.")
136
137         # Exécuter une requête (exemple)
138         cursor.execute(f"SELECT COUNT(*) FROM ticketvalide
            WHERE id_ticket_valide = '{id_ticket}';")
139
140         # Récupérer le résultat
141         result_requete4 = cursor.fetchone() # Utilisez
            fetchone() pour un seul résultat
142
143         # Afficher le résultat
144

```

```

145     print("Nombre de ticket valide scanne avec cette id
146           :", result_requete4[0]) # Le premier lment
147           du tuple
148     if result_requete4[0] > 0:
149         access_denied(client_socket, "Ticket valide deja
150             scanne")
151         return
152
153     cursor.execute(f"SELECT COUNT(*) FROM ticket WHERE
154         idticket = '{id_ticket}' AND idevenement = '{
155             id_evenement}';")
156
157     # R cup rer le r sultat
158     result_requete5 = cursor.fetchone() # Utilisez
159         fetchone() pour un seul r sultat
160
161     # Afficher le r sultat
162     print("Nombre devenement associ s au ticket:",
163         result_requete5[0]) # Le premier lment du
164         tuple
165     if result_requete5[0] == 0:
166         access_denied(client_socket, "Evenement non
167             autorise")
168         return
169
170     if est_ticket_expire(id_ticket):
171         access_denied(client_socket, "Ticket expire")
172         return
173
174     # Ex cuter une requ te (exemple)
175     cursor.execute(f"SELECT zone.nomzone FROM ticket
176         JOIN place ON ticket.idevenement = place.
177         idevenement JOIN zone ON place.idzone = zone.
178         idzone WHERE ticket.idticket = '{id_ticket}';")
179
180     # R cup rer le r sultat
181     result_requete7 = cursor.fetchone() # Utilisez
182         fetchone() pour un seul r sultat
183
184     Nom_de_la_zone=result_requete7[0]
185
186     # Afficher le r sultat
187     print("Nom de la zone associe au ticket:",
188         Nom_de_la_zone)
189
190     # Ex cuter une requ te (exemple)
191     cursor.execute(f"SELECT emplacement FROM scanner

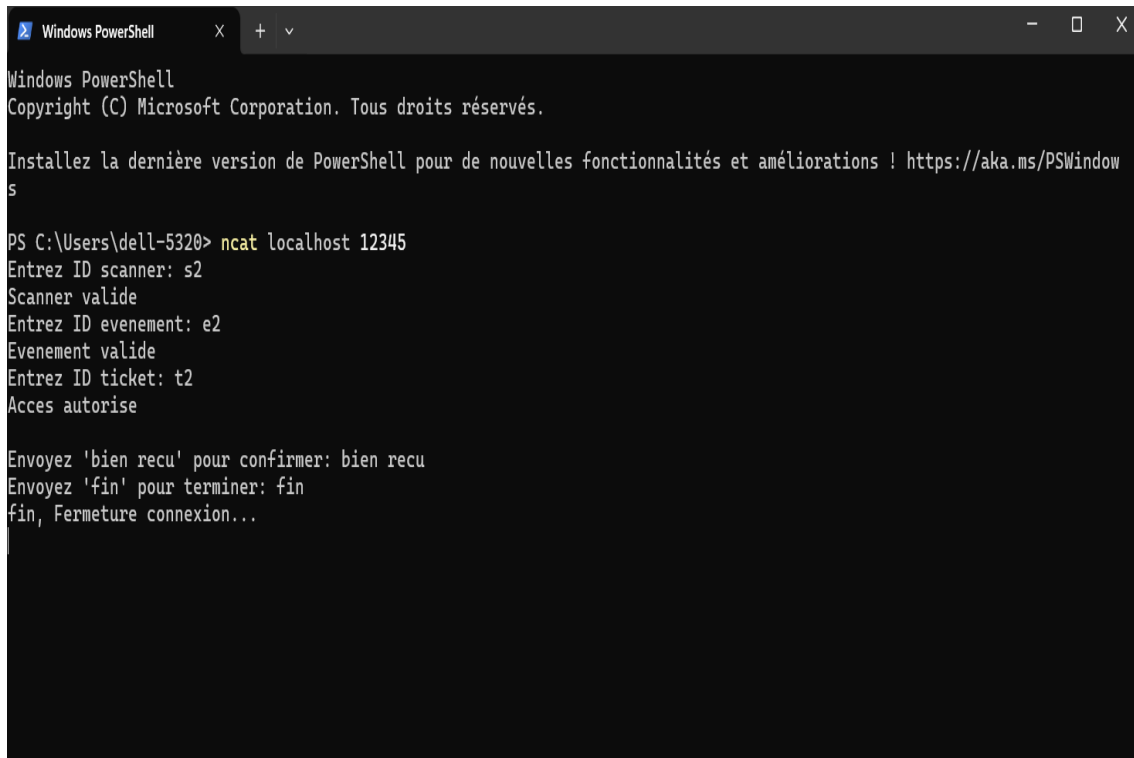
```

```

WHERE idscanner='{id_scanner}';")
181
182 # Récupérer le résultat
183 result_requete8= cursor.fetchone() # Utilisez
    fetchone() pour un seul résultat
184
185 Emplacement_du_scanner=result_requete8[0]
186
187 # Afficher le résultat
188 print("Emplacement du scanner:",
    Emplacement_du_scanner)
189
190 if Nom_de_la_zone == Emplacement_du_scanner :
191
192     heure_de_scan_du_ticket = datetime.now().
        strftime('%Y-%m-%d %H:%M:%S')
193     cursor.execute(f"INSERT INTO ticketvalide (
        idlogscan, heurescan,id_ticket_valide,
        idscanner) VALUES ('LOG' || LPAD(CAST((SELECT
        COUNT(*) FROM ticketvalide) + 1 AS TEXT), 3,
        '0'), '{heure_de_scan_du_ticket}', '{
        id_ticket}', '{id_scanner}');"
194     connection.commit() # Assurez-vous de valider l
        'insertion
195
196
197     access_authorized(client_socket)
198 else:
199     access_denied(client_socket, "Mauvaise zone")
200
201 except Exception as error:
202     print("Erreur lors de la connexion à la base de
        données :", error)
203
204 def main():
205     server_socket = socket.socket(socket.AF_INET, socket.
        SOCK_STREAM)
206     server_socket.bind(('127.0.0.1', 12345))
207     server_socket.listen(5)
208     print("Serveur écoute sur port 12345.")
209
210     while True:
211         client_socket, addr = server_socket.accept()
212         print(f"Connexion avec {addr}")
213         handle_client(client_socket)
214
215 if __name__ == "__main__":
216     main()

```

#### 8.4 Les principaux tests réseaux mis en oeuvre NCAT avec le code du serveur en python (comme client)



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\Users\dell-5320> ncat localhost 12345
Entrez ID scanner: s2
Scanner valide
Entrez ID evenement: e2
Evenement valide
Entrez ID ticket: t2
Acces autorise

Envoyez 'bien reçu' pour confirmer: bien reçu
Envoyez 'fin' pour terminer: fin
fin, Fermeture connexion...
```

FIGURE 4 – Scénario 1 : Scan d'un ticket valide

```
Invite de commandes - ncat | X + v
Microsoft Windows [version 10.0.22631.4317]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\dell-5320>ncat localhost 12345
Entrez ID scanner: s1
Scanner valide
Entrez ID evenement: e1
Evenement valide
Entrez ID ticket: t1
Acces refuse
Raison: Ticket expire
Envoyez 'bien reçu' pour confirmer: bien reçu
Envoyez 'fin' pour terminer: fin
fin, Fermeture connexion...
|
```

FIGURE 5 – Scénario 2 : Scan d'un ticket expiré

```
Windows PowerShell X + v
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\Users\dell-5320> ncat localhost 12345
Entrez ID scanner: s3
Scanner valide
Entrez ID evenement: e3
Evenement valide
Entrez ID ticket: t3
Acces refuse
Raison: Mauvaise zone
Envoyez 'bien reçu' pour confirmer: bien reçu
Envoyez 'fin' pour terminer: fin
fin, Fermeture connexion...
```

FIGURE 6 – Scénario 3 : Mauvaise zone de ticket

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\Users\dell-5320> ncat localhost 12345
Entrez ID scanner: s2
Scanner valide
Entrez ID evenement: e2
Evenement valide
Entrez ID ticket: t2
Acces refuse
Raison: Ticket deja scanne
Envoyez 'bien reçu' pour confirmer: bien reçu
Envoyez 'fin' pour terminer: fin
fin, Fermeture connexion...
```

FIGURE 7 – Scénario 4 : Ticket valide déjà scanné

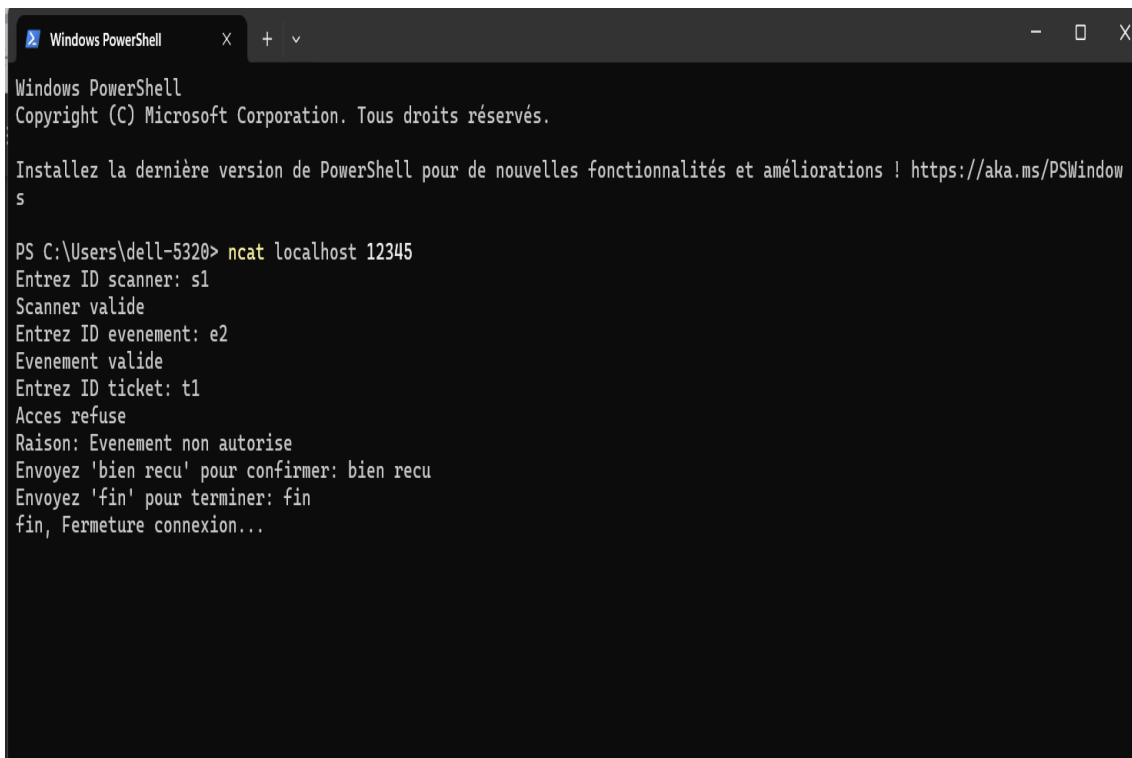
```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\Users\dell-5320> ncat localhost 12345
Entrez ID scanner: s1
Scanner valide
Entrez ID evenement: e1
Evenement valide
Entrez ID ticket: t9
Acces refuse
Raison: Ticket inexistant
Envoyez 'bien reçu' pour confirmer: bien reçu
Envoyez 'fin' pour terminer: fin
fin, Fermeture connexion...
|
```

FIGURE 8 – Scénario 5 : Scan d'un ticket inexistant





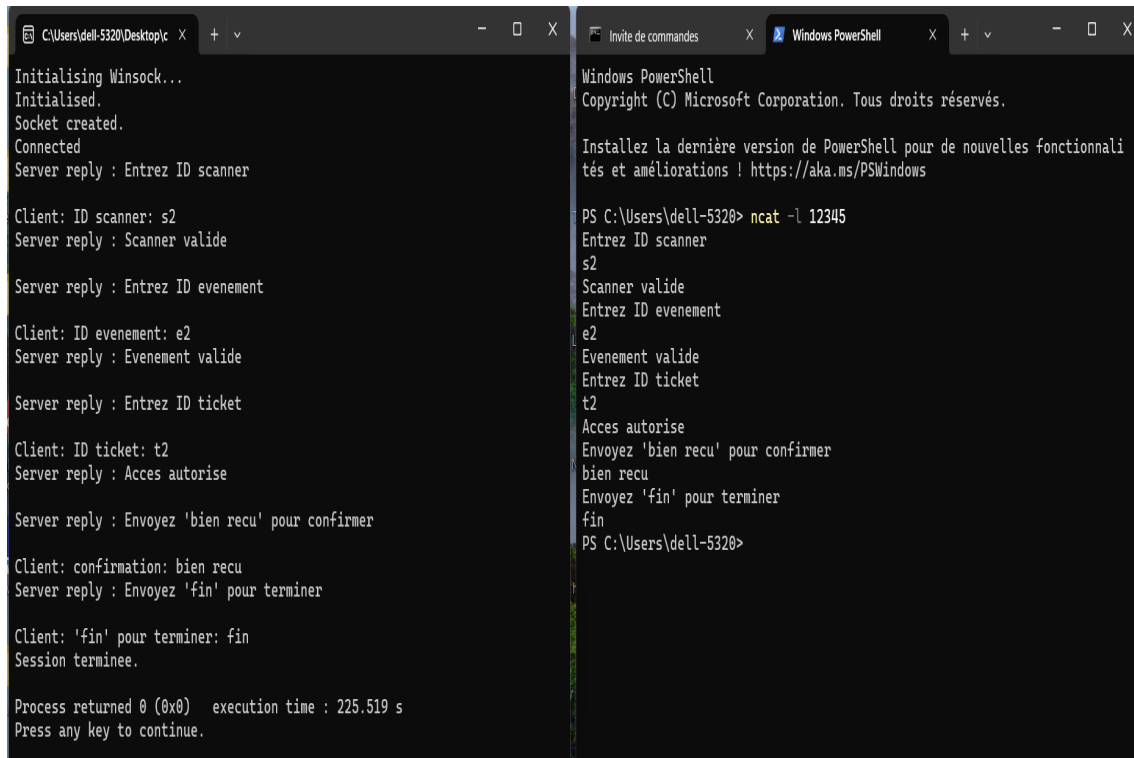
```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\Users\dell-5320> ncat localhost 12345
Entrez ID scanner: s1
Scanner valide
Entrez ID evenement: e2
Evenement valide
Entrez ID ticket: t1
Accès refuse
Raison: Evenement non autorise
Envoyez 'bien reçu' pour confirmer: bien reçu
Envoyez 'fin' pour terminer: fin
fin, Fermeture connexion...
```

FIGURE 9 – Scénario 6 : Accès à un événement non autorisé

## 8.5 Les principaux tests réseaux mis en oeuvre NCAT avec le code du client en C (comme serveur)



```
C:\Users\dell-5320\Desktop\c x + v - □ X
Initialising Winsock...
Initialised.
Socket created.
Connected
Server reply : Entrez ID scanner

Client: ID scanner: s2
Server reply : Scanner valide

Server reply : Entrez ID evenement

Client: ID evenement: e2
Server reply : Evenement valide

Server reply : Entrez ID ticket

Client: ID ticket: t2
Server reply : Acces autorise

Server reply : Envoyez 'bien reçu' pour confirmer

Client: confirmation: bien reçu
Server reply : Envoyez 'fin' pour terminer

Client: 'fin' pour terminer: fin
Session terminée.

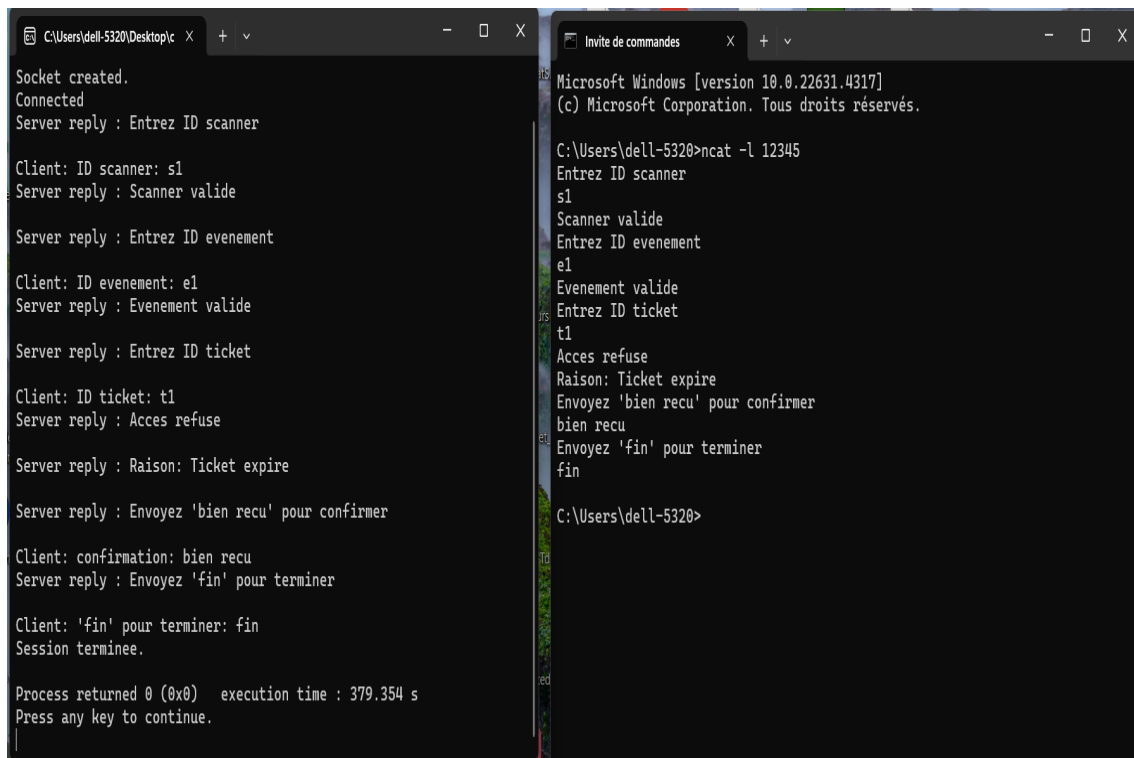
Process returned 0 (0x0) execution time : 225.519 s
Press any key to continue.

Invite de commandes x Windows PowerShell x + v - □ X
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\Users\dell-5320> ncat -l 12345
Entrez ID scanner
s2
Scanner valide
Entrez ID evenement
e2
Evenement valide
Entrez ID ticket
t2
Acces autorise
Envoyez 'bien reçu' pour confirmer
bien reçu
Envoyez 'fin' pour terminer
fin
PS C:\Users\dell-5320>
```

FIGURE 10 – Scénario 1 : Scan d'un ticket valide



```
C:\Users\dell-5320\Desktop>c
Socket created.
Connected
Server reply : Entrez ID scanner

Client: ID scanner: s1
Server reply : Scanner valide

Server reply : Entrez ID evenement

Client: ID evenement: e1
Server reply : Evenement valide

Server reply : Entrez ID ticket

Client: ID ticket: t1
Server reply : Acces refuse

Server reply : Raison: Ticket expire

Server reply : Envoyez 'bien reçu' pour confirmer

Client: confirmation: bien reçu
Server reply : Envoyez 'fin' pour terminer

Client: 'fin' pour terminer: fin
Session terminée.

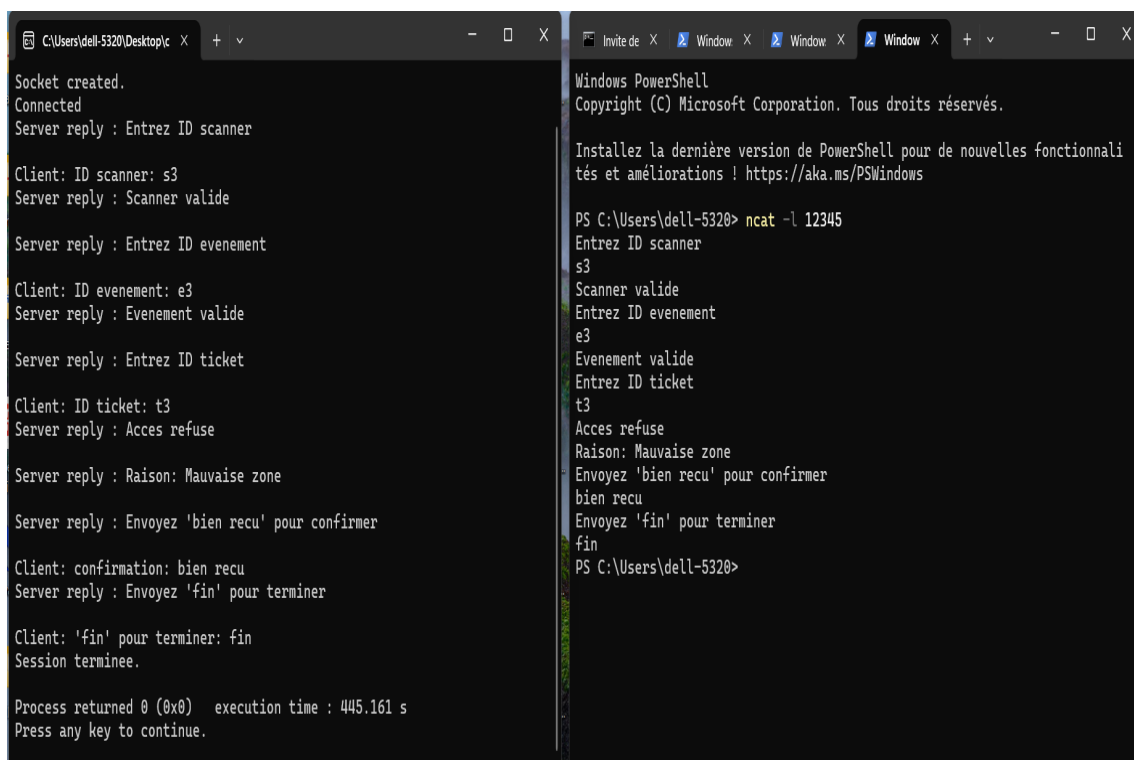
Process returned 0 (0x0)   execution time : 379.354 s
Press any key to continue.
```

```
Microsoft Windows [version 10.0.22631.4317]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\dell-5320>ncat -l 12345
Entrez ID scanner
s1
Scanner valide
Entrez ID evenement
e1
Evenement valide
Entrez ID ticket
t1
Acces refuse
Raison: Ticket expire
Envoyez 'bien reçu' pour confirmer
bien reçu
Envoyez 'fin' pour terminer
fin

C:\Users\dell-5320>
```

FIGURE 11 – Scénario 2 : Scan d'un ticket expiré



```
C:\Users\dell-5320\Desktop>c
Socket created.
Connected
Server reply : Entrez ID scanner

Client: ID scanner: s3
Server reply : Scanner valide

Server reply : Entrez ID evenement

Client: ID evenement: e3
Server reply : Evenement valide

Server reply : Entrez ID ticket

Client: ID ticket: t3
Server reply : Acces refuse

Server reply : Raison: Mauvaise zone

Server reply : Envoyez 'bien reçu' pour confirmer

Client: confirmation: bien reçu
Server reply : Envoyez 'fin' pour terminer

Client: 'fin' pour terminer: fin
Session terminée.

Process returned 0 (0x0)   execution time : 445.161 s
Press any key to continue.
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnali
tés et améliorations ! https://aka.ms/PSWindows

PS C:\Users\dell-5320> ncat -l 12345
Entrez ID scanner
s3
Scanner valide
Entrez ID evenement
e3
Evenement valide
Entrez ID ticket
t3
Acces refuse
Raison: Mauvaise zone
Envoyez 'bien reçu' pour confirmer
bien reçu
Envoyez 'fin' pour terminer
fin

PS C:\Users\dell-5320>
```

FIGURE 12 – Scénario 3 : Mauvaise zone de ticket

```
C:\Users\dell-5320\Desktop\c x + v - □ X
Socket created.
Connected
Server reply : Entrez ID scanner

Client: ID scanner: s2
Server reply : Scanner valide

Server reply : Entrez ID evenement

Client: ID evenement: e2
Server reply : Evenement valide

Server reply : Entrez ID ticket

Client: ID ticket: t2
Server reply : Acces refuse

Server reply : Raison: Ticket deja scanne

Server reply : Envoyez 'bien reçu' pour confirmer

Client: confirmation: bien reçu
Server reply : Envoyez 'fin' pour terminer

Client: 'fin' pour terminer: fin
Session terminee.

Process returned 0 (0x0) execution time : 156.739 s
Press any key to continue.
```

```
Invite de commi x Windows Power X Windows Power X + v - □ X
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnali
tés et améliorations ! https://aka.ms/PSWindows

PS C:\Users\dell-5320> ncat -l 12345
Entrez ID scanner
s2
Scanner valide
Entrez ID evenement
e2
Evenement valide
Entrez ID ticket
t2
Acces refuse
Raison: Ticket deja scanne
Envoyez 'bien reçu' pour confirmer
bien reçu
Envoyez 'fin' pour terminer
fin
PS C:\Users\dell-5320>
```

FIGURE 13 – Scénario 4 : Ticket valide déjà scanné

```
C:\Users\dell-5320\Desktop\c x + v - □ X
Socket created.
Connected
Server reply : Entrez ID scanner

Client: ID scanner: s1
Server reply : Scanner valide

Server reply : Entrez ID evenement

Client: ID evenement: e1
Server reply : Evenement valide

Server reply : Entrez ID ticket

Client: ID ticket: t9
Server reply : Acces refuse

Server reply : Raison: Ticket inexistant

Server reply : Envoyez 'bien reçu' pour confirmer

Client: confirmation: bien reçu
Server reply : Envoyez 'fin' pour terminer

Client: 'fin' pour terminer: fin
Session terminee.

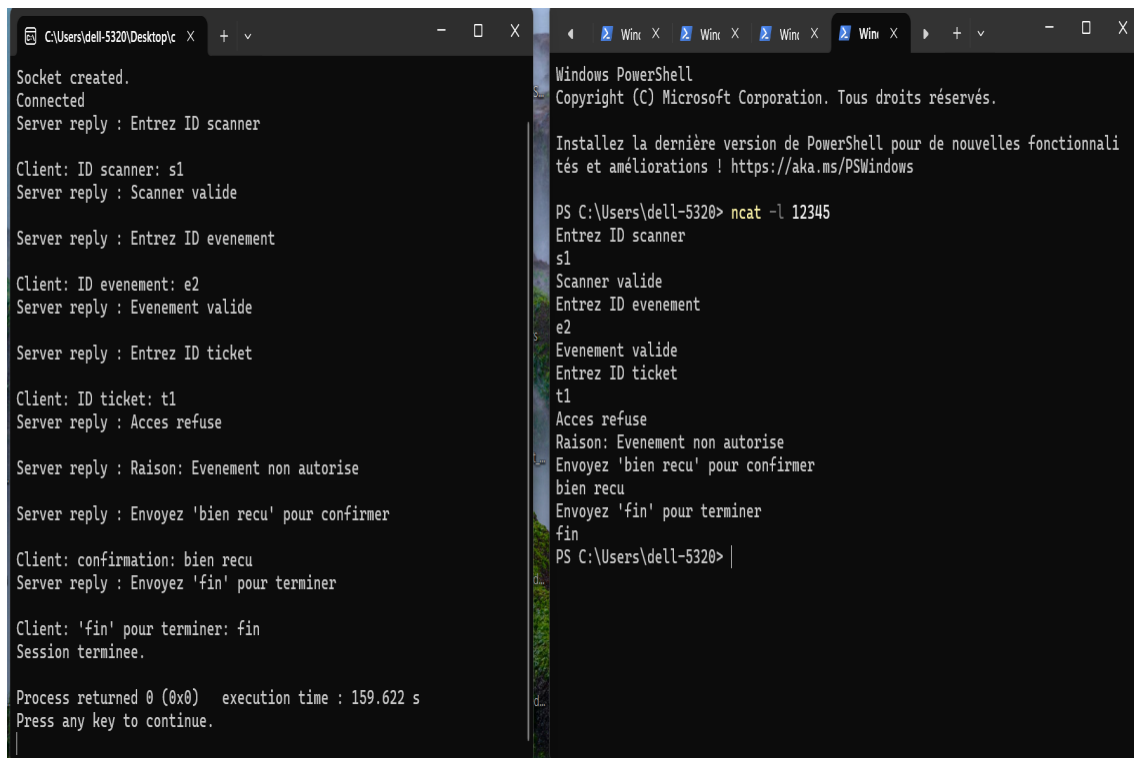
Process returned 0 (0x0) execution time : 236.961 s
Press any key to continue.
```

```
Winn x Winn x Winn x Winn x + v - □ X
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnali
tés et améliorations ! https://aka.ms/PSWindows

PS C:\Users\dell-5320> ncat -l 12345
Entrez ID scanner
s1
Scanner valide
Entrez ID evenement
e1
Evenement valide
Entrez ID ticket
t9
Acces refuse
Raison: Ticket inexistant
Envoyez 'bien reçu' pour confirmer
bien reçu
Envoyez 'fin' pour terminer
fin
PS C:\Users\dell-5320> |
```

FIGURE 14 – Scénario 5 : Scan d'un ticket inexistant



The image shows two terminal windows side-by-side. The left window is a netcat listener on port 12345, and the right window is a Windows PowerShell session running the netcat listener. The netcat listener shows the following sequence of events:

```
Socket created.
Connected
Server reply : Entrez ID scanner
Client: ID scanner: s1
Server reply : Scanner valide
Server reply : Entrez ID evenement
Client: ID evenement: e2
Server reply : Evenement valide
Server reply : Entrez ID ticket
Client: ID ticket: t1
Server reply : Acces refuse
Server reply : Raison: Evenement non autorise
Server reply : Envoyez 'bien reçu' pour confirmer
Client: confirmation: bien reçu
Server reply : Envoyez 'fin' pour terminer
Client: 'fin' pour terminer: fin
Session terminée.
Process returned 0 (0x0)   execution time : 159.622 s
Press any key to continue.
```

The Windows PowerShell session shows the following sequence of events:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\Users\dell-5320> ncat -l 12345
Entrez ID scanner
s1
Scanner valide
Entrez ID evenement
e2
Evenement valide
Entrez ID ticket
t1
Acces refuse
Raison: Evenement non autorise
Envoyez 'bien reçu' pour confirmer
bien reçu
Envoyez 'fin' pour terminer
fin
PS C:\Users\dell-5320> |
```

FIGURE 15 – Scénario 6 : Accès à un événement non autorisé

## 8.6 URL et captures d'écran du site web

URL du projet : <https://eventticket-manager.alwaysdata.net/>

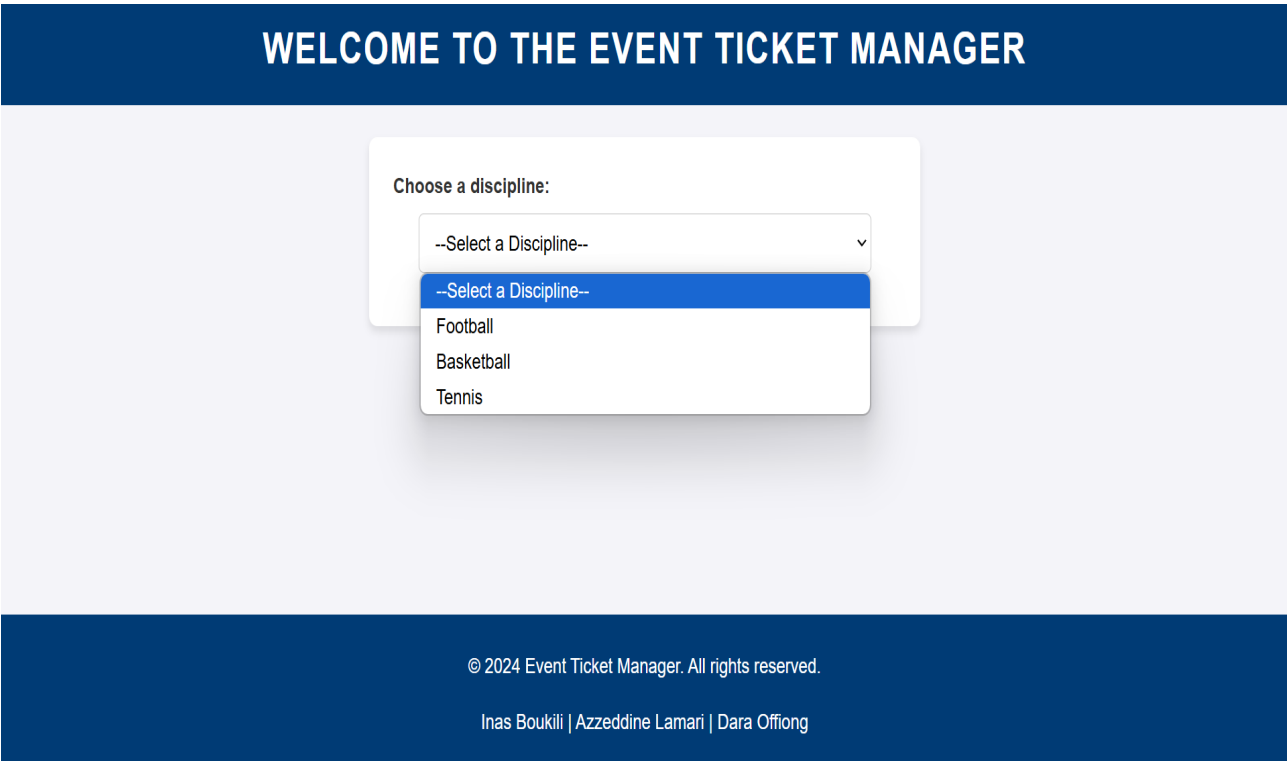


FIGURE 16 – Selectionner une discipline

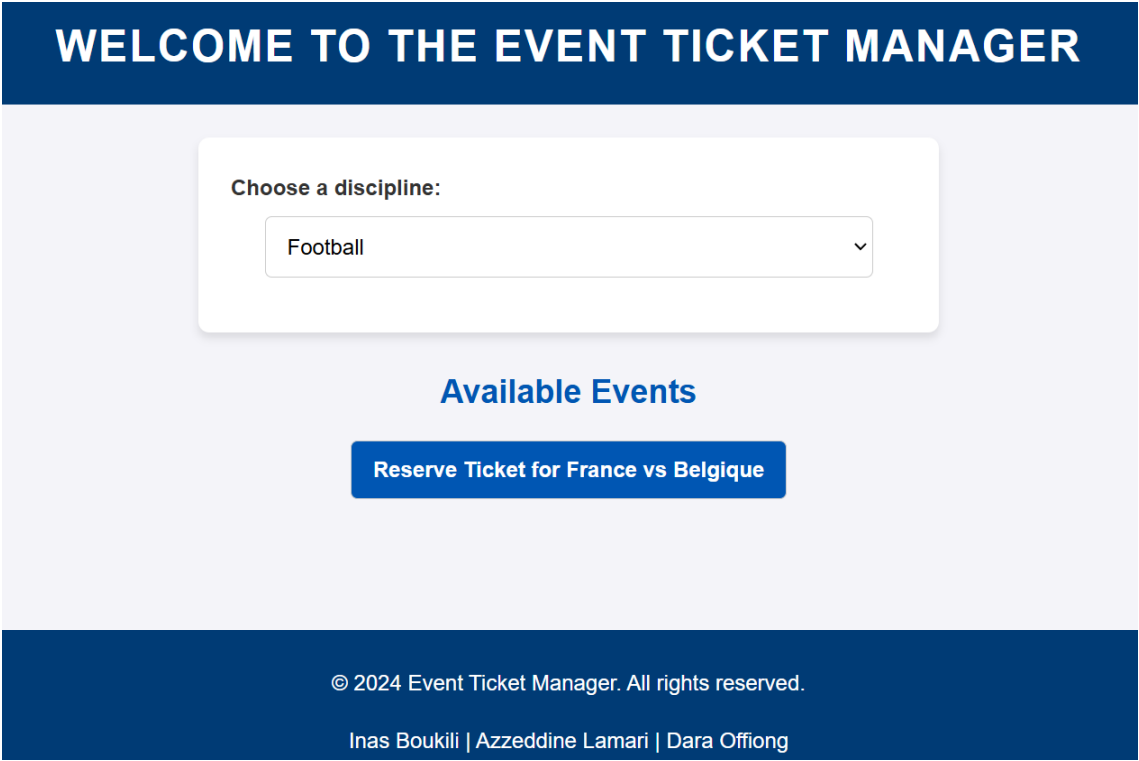


FIGURE 17 – Selectionner un evenement

Select a Place:

VIP

VIP

Normale

Handicapee

details:

Last Name:

First Name:

Email:

© 2024 Event Ticket Manager. All rights reserved.

FIGURE 18 – Choisir l’emplacement

# Payment Gateway

## Reservation Details

**Match:** France vs Belgique

**Zone:** someZone

**Name:** edogawa conan

**Email:** bklannousa@gmail.com

## Enter Payment Details

**Card Number:**

**Expiry Date:**

**CVV:**

© 2024 Event Ticket Manager. All rights reserved.

FIGURE 19 – Entrer les détails de paiement



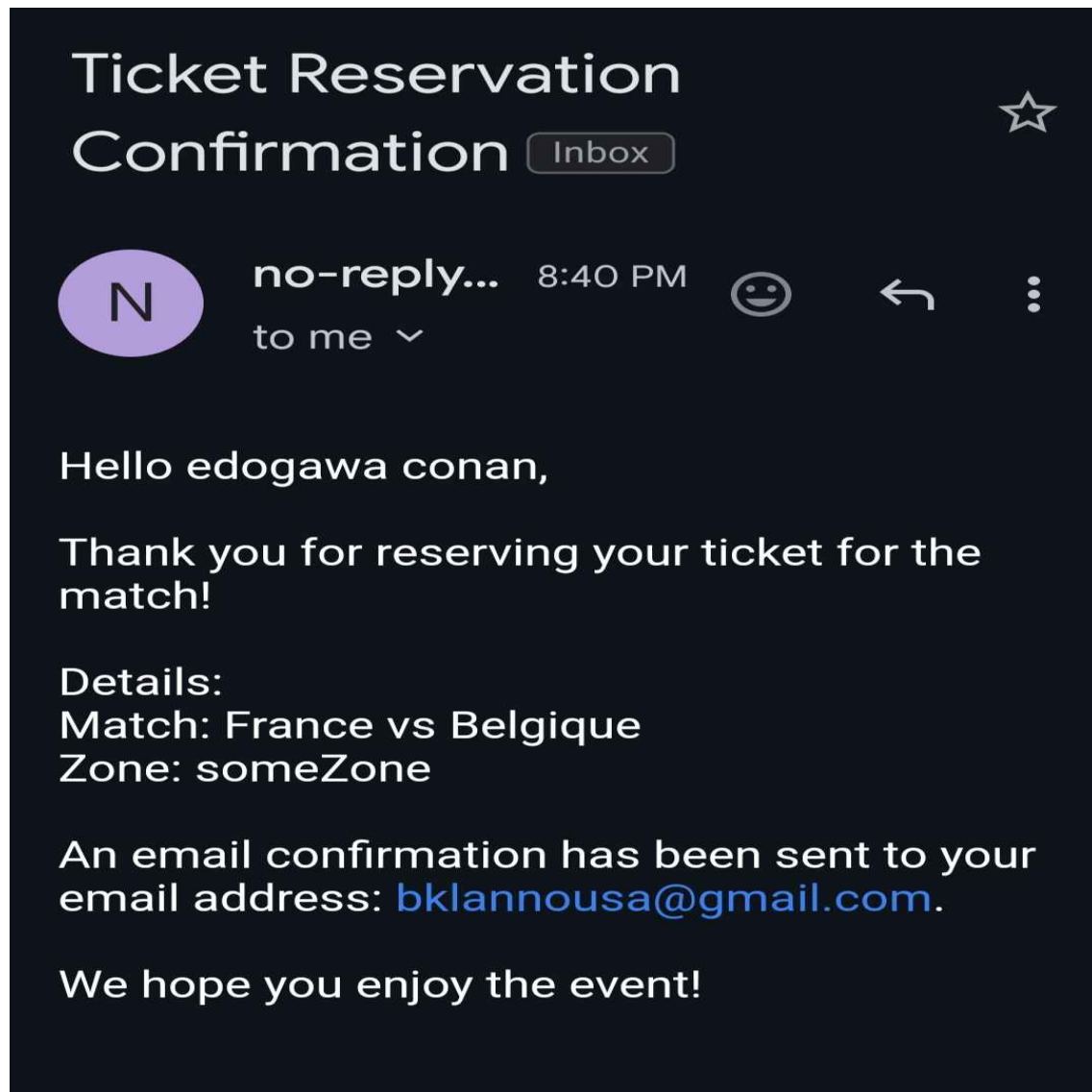


FIGURE 20 – Mail de confirmation

## 8.7 Les 9 requêtes SQL SELECT utilisées

```
— SELECT evenement.dateevenement, evenement.heurefineevenement
FROM ticket
JOIN evenement ON ticket.idevenement = evenement.idevenement
WHERE ticket.idticket = '{id_ticket}';
— SELECT COUNT(*)
FROM scanner
WHERE idscanner = '{id_scanner}';
— SELECT COUNT(*)
FROM evenement
WHERE idevenement = '{id_evenement}';
— SELECT COUNT(*)
FROM ticket
WHERE idticket = '{id_ticket}';
— SELECT COUNT(*)
FROM peutetrescanne
```

```

WHERE idticket = '{id_ticket}'
AND idscanner = '{id_scanner}';
— SELECT COUNT(*)
FROM ticketvalide
WHERE id_ticket_valide = '{id_ticket}';
— SELECT COUNT(*)
FROM ticket
WHERE idticket = '{id_ticket}'
AND idevenement = '{id_evenement}';
— SELECT zone.nomzone
FROM ticket
JOIN place ON ticket.idevenement = place.idevenement
JOIN zone ON place.idzone = zone.idzone
WHERE ticket.idticket = '{id_ticket}';
— SELECT emplacement
FROM scanner
WHERE idscanner = '{id_scanner}';

```