

CY CERGY PARIS UNIVERSITE

RAPPORT

Projet BASES DE DONNEES ET RESEAU

Hôtel TAJ

LICENCE 3 INFORMATIQUE

Rédigé par

**DAMODARANE JEAN-BAPTISTE, ELUMALAI
SRIGURU et SANTOKI NIHAR**



Octobre 2022

Table des matières

I.	INTRODUCTION	4
A.	Contexte du Projet	4
II.	CONCEPTION ET REALISATION DU PROJET	4
A.	Dictionnaire de données	4
B.	Modélisation Conceptuelle des données (MCD)	5
C.	Modélisation Logique des données (MLD)	6
D.	Quelques exemples de requêtes SQL et leurs résultats en affichage	6
E.	Schémas Applicatif	8
F.	Echanges Réseaux	10
G.	Programmation Réseau (Sockets)	10
Socket côté Serveur	10	
Socket côté Client	11	
III.	DEROULEMENT DU PROJET	11
A.	Diagramme de Gantt	11
B.	Répartitions des tâches	12
VI.	CONCLUSION ET PERSPECTIVES	13
A.	Résumé du travail réalisé	13
Résumé du projet	13	
Défis rencontrés	13	
B.	Améliorations possibles	14

Remerciements

Nous tenons à remercier très sincèrement et à témoigner toute notre reconnaissance à nos professeurs Madame Dang Ngoc Tuyet tram, Monsieur Marc Lemaire et Monsieur Jen Tao Yuan, qui nous ont enseigné lors des CM et des TD, le concept des bases de données, le réseau, le langage SQL et ainsi que le langage PHP lors du semestre 5, et cela nous a permis de nous introduire dans une dimension différente du monde de l'informatique. Nous remercions encore une fois, nos professeurs, qui nous ont guidé et conseillé lorsque nous avons rencontré des problèmes pendant que nous développons notre projet.

I. INTRODUCTION

Dans le cadre de notre troisième année de Licence en Informatique à CY Cergy Paris Université, nous avons eu à réaliser, un projet de 4 mois au sein du module Projet Bases de données et Réseaux.

A. Contexte du Projet

Nous avons fait le choix de réaliser un site web permettant d'effectuer la réservation des chambres d'un hôtel en ligne. L'hôtel propose différents types de chambres, telles que : chambre simple, chambre double, chambre économique et chambre de luxe. Le prix de réservation varie en fonction des chambres choisies et de leurs durées d'occupation par les clients.

Différents mode de paiement sont mis en place : possibilité de payer en ligne ou bien sur place. L'hôtel fournit un badge avec lequel, le client peut accéder à sa chambre en validant son badge par un capteur se trouvant près de la porte d'entrée de sa chambre. Le client peut, en effet, augmenter la validité de son badge au sein de l'hôtel, dans le cas où il souhaiterait prolonger sa durée d'occupation de la chambre. Toutefois, le paiement augmentera également. Si le paiement n'est pas effectué, il perdra l'accès à sa chambre et devra en réserver une autre.

L'hôtel a choisi d'utiliser les badges afin d'éviter tout types de délits et notamment pour son haut niveau de fiabilité et de sécurité.

Au cas où il y a un dysfonctionnement du badge ou de la porte, alors l'hôtel peut, exceptionnellement, fournir des clés aux clients. Mais avant cela, un contrôle devra être fait au niveau du fonctionnement du badge et de la porte par une équipe de maintenance.

Il y a également un second type de badge, réservé seuls aux personnels de l'hôtel. Ces badges donnent accès à l'ensemble des chambres mais aussi aux autres pièces de l'hôtel. Par exemple, le badge de manager ou celui de l'équipe de surveillance peut accéder à l'espace de surveillance avec caméras et d'autres pièces non-autorisées aux clients.

II. CONCEPTION ET REALISATION DU PROJET

A. Dictionnaire de données

Entités	Attribut	Type	domaine/contraintes	NULL / NOT NULL	description/exemple
CLIENT	num_client	CHAR(10)	alphanumérique	NOT NULL	cl12345678
	nom_client	VARCHAR(25)	alphabétique	NOT NULL	Robinson
	prenom_client	VARCHAR(25)	alphabétique	NOT NULL	Jhon
	mail_client	VARCHAR(50)	alphanumérique	NOT NULL	robinson16@gmail.com
	tel_client	CHAR(10)	numérique	NOT NULL	1234567899 et > 0 et 10 char obligatoirement
	addr_client	VARCHAR(100)	alphanumérique	NOT NULL	7 rue de la paix, 92000 Nanterre

Entités	Attribut	Type	domaine/contraintes	NULL / NOT NULL	description/exemple
CHAMBRE	num_chambre	CHAR(5)	numérique	NOT NULL	10001
	capacité	INTEGER	numérique	NULL	2 et > 0, null = chambre utilise par hotel
	typeChambre	VARCHAR(50)	alphabétique	NULL	chambre double
	occupation	BOOL	true/false	NOT NULL	occupé : true / disponible : false
	tel_chambre	CHAR(10)	numérique	NOT NULL	1234567899
	num_etage	INTEGER	numérique	NOT NULL	1
	prix	FLOAT	numérique	NULL	100,10, null = si chambre utilise par hotel
	description	TEXT	alphanumérique	NULL	chambre avec vue extérieur et une balcon
RESERVATION	num_reservation	CHAR(10)	numérique	NOT NULL	1234567899
	date_reservation	DATE	date	NOT NULL	2022-10-02
	prixPaiement	FLOAT	numérique	NOT NULL	200,50 et >= 0
	date_debut	DATE	date	NOT NULL	2022-10-02
	date_fin	DATE	date	NOT NULL	2022-10-05
FACTURE	num_facture	CHAR(8)	alphanumérique	NOT NULL	F1234567
	somme_totale	FLOAT	numérique	NOT NULL	200,50 et >= 0
	date_facture	DATE	date	NOT NULL	2022-10-02
BADGE	num_badge	CHAR(5)	alphanumérique	NOT NULL	B1101
	droitAcces	BOOL	true/false	NOT NULL	autorisé : true / interdit : false
	tentative_access	INT	numérique	NOT NULL	1 et >= 0
EMPLOYE	num_employe	CHAR(8)	numérique	NOT NULL	ht123456
	nom_employe	VARCHAR(25)	alphabétique	NOT NULL	TOTO
	prenom_employe	VARCHAR(30)	alphabétique	NOT NULL	TATA
	mail_employe	VARCHAR(50)	alphanumérique	NOT NULL	toto.tata@gmail.com
	tel_employe	CHAR(10)	numérique	NOT NULL	1234567899
	type_contrat	VARCHAR(25)	alphabétique	NOT NULL	CDD
	salaires	FLOAT	numérique	NOT NULL	1600 et >= 1539.42 (salaire brut)
	poste	VARCHAR(50)	alphabétique	NOT NULL	Manager
DEPARTEMENTS	num_departement	CHAR(2)	numérique	NOT NULL	1
	nom_departement	VARCHAR(50)	alphabétique	NOT NULL	Service Accueil
	responsable_departement	VARCHAR(24)	alphabétique	NULL	M.Dupont
	nbre_personnels	INTEGER	numérique	NOT NULL	20 et > 0
	type_departement	VARCHAR(25)	alphabétique	NULL	
	description	TEXT	alphanumérique	NULL	

B. Modélisation Conceptuelle des données (MCD)

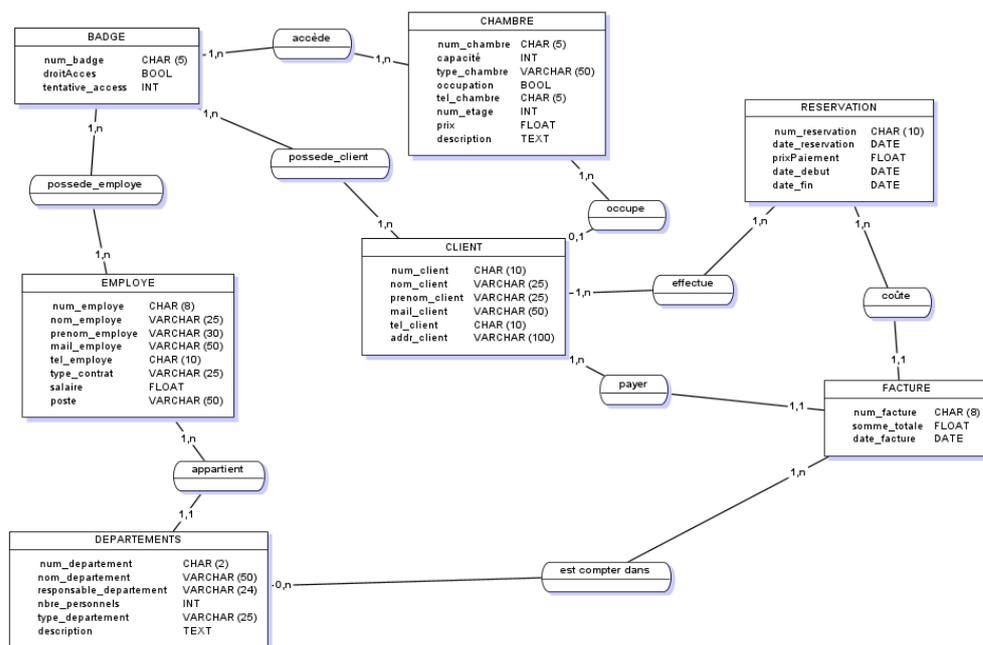


Figure 1: Image illustrant la modélisation conceptuelle des données

C. Modélisation Logique des données (MLD)

EMPLOYE (num_employe, nom_employe, prenom_employe, mail_employe, tel_employe, type_contrat, salaire, poste) ;

BADGE (num_badge, droitAcces, tentative_access) ;

CHAMBRE (num_chambre, capacite, type_chambre, occupation, tel_chambre, num_etage, prix, description) ;

RESERVATION (num_reservation, date_reservation, prixPaiement, date_debut, date_fin) ;

CLIENT (num_client, nom_client, prenom_client, mail_client, tel_client, addr_client, #num_chambre) ;

EFFECTUER (#num_client, #num_reservation) ;

ACCEDER(#num_badge, #num_chambre) ;

POSSEDE_CLIENT(#num_badge, #num_client) ;

POSSEDE_EMPLOYE(#num_badge, #num_employe) ;

DEPARTEMENTS (num_departement, nom_departement, responsable_departement, nbre_personnels, type_departement, description, #num_employe) ;

FACTURE (num_facture, somme_totale, date_facture, #num_client, #num_reservation) ;

EST_COMPTER_DANS (#num_departement, #num_facture) ;

D. Quelques exemples de requêtes SQL et leurs résultats en affichage

1- Affichage du nombre de chambres qui sont occupées dans l'hôtel :

```
SELECT COUNT(num_chambre) FROM chambre WHERE occupation = TRUE;
tjaj_hotel=# SELECT COUNT(num_chambre) FROM chambre WHERE occupation = TRUE;
count
-----
      4
(1 ligne)
```

2- Affichage de toutes les informations sur le Manager qui reçoit le plus grand salaire :

```
SELECT * FROM employe WHERE salaire=(SELECT MAX(salaire) FROM employe
WHERE poste Like '%Manager%');
```

```
tjaj_hotel=# SELECT * FROM employe WHERE salaire=(SELECT MAX(salaire) FROM employe WHERE poste Like '%Manager%');
 num_employe | nom_employe | prenom_employe | mail_employe | tel_employe | type_contrat | salaire | poste
-----+-----+-----+-----+-----+-----+-----+-----
 ht193121   | Zdenek     | Mercure       | ZdenekMercure@teleworm.us | 0715502305 | CDD          | 5246.35 | Manager
(1 ligne)
```

3- Affichage des noms de départements et du salaire max par départements dans lequel le salaire maximal est supérieur ou égal au salaire moyen dans l'ordre décroissant :

```
SELECT MAX(salaire), nom_departement FROM employe INNER JOIN departments
ON employe.num_employe=departments.num_employe GROUP BY nom_departement
HAVING MAX(salaire) >= AVG(salaire) ORDER BY MAX(salaire) DESC;
```

```
taj_hotel=# SELECT MAX(salaire), nom_department FROM employe INNER JOIN departments ON employe.num_employe=departments.num_employe GROUP BY nom_department HAVING MAX(salaire) >= AVG(salaire)
) ORDER BY MAX(salaire) DESC;
max | nom_department
-----+-----
5246.35 | Service de gestion hotel
3684.01 | Service restauration
2536.26 | Securite et surveillance
(3 lignes)
```

- 4- Affichage du numéro des chambres de l'hôtel n'ayant pas de prix avec une capacité définie et de leurs descriptions :

```
SELECT num_chambre AS Chambre_utiliser_par_Hotel, description from
chambre WHERE capacite IS NULL AND prix IS NULL;
```

```
taj_hotel=# SELECT num_chambre AS Chambre_utiliser_par_Hotel, description from chambre WHERE capacite IS NULL AND prix IS NULL;
chambre_utiliser_par_hotel | description
-----+-----
10002 | chambre de surveillance
10003 | Les serveur de données
(2 lignes)
```

- 5- Affichage des informations sur le client qui payent une facture minimale parmi tous les clients :

```
SELECT * FROM client JOIN facture ON client.num_client=facture.num_client
WHERE somme_totale =(SELECT MIN(somme_totale) FROM facture);
```

```
taj_hotel=# SELECT * FROM client JOIN facture ON client.num_client=facture.num_client WHERE somme_totale =(SELECT MIN(somme_totale) FROM facture);
num_client | nom_client | prenom_client | mail_client | tel_client | addr_client | num_chambre | num_facture | somme_totale | date_facture | num_reservation | num_client
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
c187654321 | Christopher | Milton | milton@gmail.com | 0676233212 | 5 Rue du Général, 93000 | 20001 | F0123456 | 2800 | 2022-12-20 | 3123456789 | c187654321
(1 ligne)
```

- 6- Affichage des réservations en nombre de jours avec le nom et le prénom du client :

```
SELECT date_fin - date_debut AS jour_reserve, nom_client, prenom_client
AS jours_reserver FROM reservation JOIN effectuer ON reservation.num_re-
servation=effectuer.num_reservation JOIN client ON effec-
tuer.num_client=client.num_client;
```

```
taj_hotel=# SELECT date_fin - date_debut AS jour_reserve, nom_client, prenom_client AS jours_reserver FROM reservation JOIN effectuer ON reservation.num_reservation=effectuer.num_reservatio
n JOIN client ON effectuer.num_client=client.num_client;
jour_reserve | nom_client | prenom_client | jours_reserver
-----+-----+-----+-----
24 | Bernard | Legrand | 24
24 | Christopher | Milton | 24
24 | Lemoulin | Lerat | 24
(3 lignes)
```

- 7- Affichage du salaire, du poste, du nom et du prénom des employés qui ne sont pas manager :

```
SELECT salaire, nom_employe, prenom_employe, poste FROM employe WHERE
poste Not Like '%Manager%';
```

```
taj_hotel=# SELECT salaire, nom_employe, prenom_employe, poste FROM employe WHERE poste Not Like '%Manager%';
salaire | nom_employe | prenom_employe | poste
-----+-----+-----+-----
2536.26 | Gilbert | Levasseur | Surveiller
3684.01 | Ferragus | Caya | Responsable restauration
1956.66 | Sébastien | Courcelle | Receptionniste
1754.84 | Varden | Chalut | Serveur
(4 lignes)
```

- 8- Affichage de toutes les informations sur les chambres qui ont un prix supérieur à 150 et qui ne sont pas occupées par des clients :

```
SELECT * FROM chambre GROUP BY chambre.num_chambre,prix HAVING prix >= 150
AND occupation = FALSE;
```

```
taj_hotel=# SELECT * FROM chambre GROUP BY chambre.num_chambre ,prix HAVING prix >= 150 AND occupation = FALSE;
num_chambre | capacite | type_chambre | occupation | tel_chambre | num_etage | prix | description
-----
30004 | 1 | chambre luxe | f | 0963636368 | 3 | 150 | permet de avoir une vue globale de la cité et nottament accés à une télescope pour visualiser
la ciel pendant la nuit
30001 | 1 | chambre luxe | f | 0963636369 | 3 | 150 | permet de avoir une vue globale de la cité et nottament accés à une télescope pour visualiser
la ciel pendant la nuit
10005 | 2 | double deluxe | f | 0952525253 | 1 | 200 | chambre avec une balcon et possiblité de voir la plage de lexterieur
40001 | 2 | double deluxe | f | 0952525254 | 4 | 200 | chambre avec une balcon et possiblité de voir la plage de lexterieur
40002 | 2 | double deluxe | f | 0952525255 | 4 | 200 | chambre avec une balcon et possiblité de voir la plage de lexterieur
10004 | 2 | double deluxe | f | 0952525251 | 1 | 200 | chambre avec une balcon et possiblité de voir la plage de lexterieur
30002 | 1 | chambre luxe | f | 0963636366 | 3 | 150 | permet de avoir une vue globale de la cité et nottament accés à une télescope pour visualiser
la ciel pendant la nuit
50001 | 40 | Spéciale pour party | f | 0972727272 | 15 | 10000 | piscine, terrasse avec une petite jardin, cuisine extérieur, des tables et bancs etc...
40003 | 2 | double deluxe | f | 0952525256 | 4 | 200 | chambre avec une balcon et possiblité de voir la plage de lexterieur
10001 | 2 | double deluxe | f | 0952525252 | 1 | 200 | chambre avec une balcon et possiblité de voir la plage de lexterieur
40005 | 2 | double deluxe | f | 0952525258 | 4 | 200 | chambre avec une balcon et possiblité de voir la plage de lexterieur
40004 | 2 | double deluxe | f | 0952525257 | 4 | 200 | chambre avec une balcon et possiblité de voir la plage de lexterieur
30003 | 1 | chambre luxe | f | 0963636367 | 3 | 150 | permet de avoir une vue globale de la cité et nottament accés à une télescope pour visualiser
la ciel pendant la nuit
(13 lignes)
```

9- Affichage de différents types de chambre avec leur prix dans l'hôtel :

```
SELECT DISTINCT type_chambre, prix FROM chambre;
```

```
taj_hotel=# SELECT DISTINCT type_chambre, prix FROM chambre;
type_chambre | prix
-----
Spéciale pour party | 10000
chambre luxe | 150
Sérveur des données
Chambre Surveillance camera
double deluxe | 200
simple | 50
(6 lignes)
```

10- Affichage des informations sur le dernier client ajouté dans la base de données :

```
SELECT * FROM client ORDER BY num_client, nom_client, prenom_client,
mail_client, tel_client, num_chambre ASC LIMIT 1;
```

```
taj_hotel=# SELECT * FROM client ORDER BY num_client, nom_client, prenom_client, mail_client, tel_client, num_chambre ASC LIMIT 1;
num_client | nom_client | prenom_client | mail_client | tel_client | addr_client | num_chambre
-----
cl11223344 | Lemoulin | Lerat | lerat@gmail.com | 0676932212 | 5 Rue de Maitre, 93000 | 30005
(1 ligne)
```

E. Schémas Applicatif

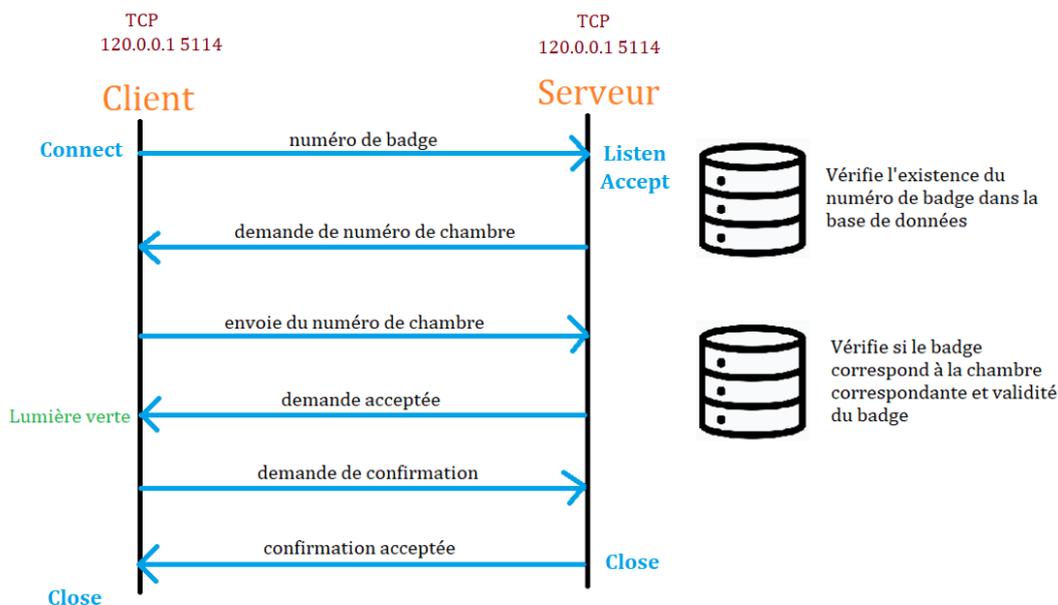


Figure 2: Image illustrant le cas normal de fonctionnement d'échanges entre le client et le serveur

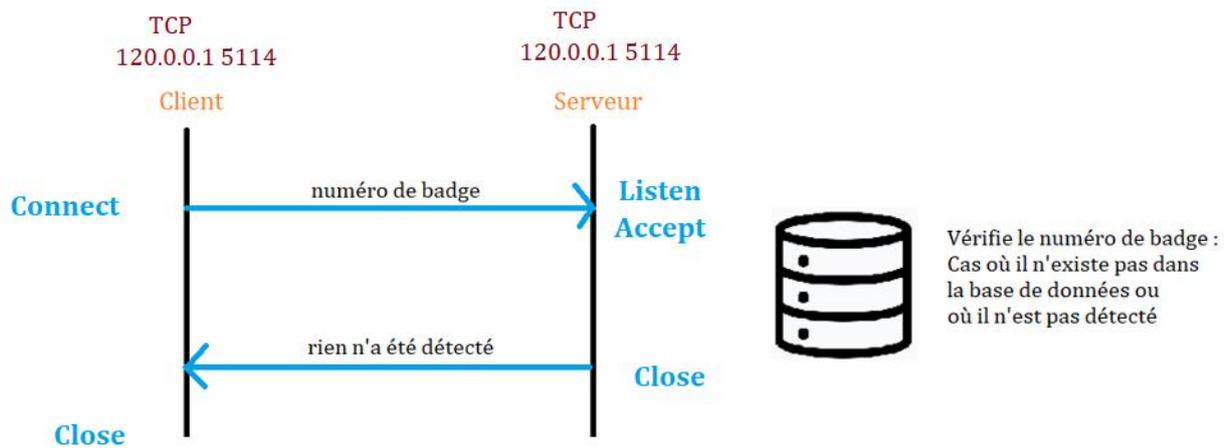


Figure3: Image illustrant le cas où le serveur ne détecte pas le badge

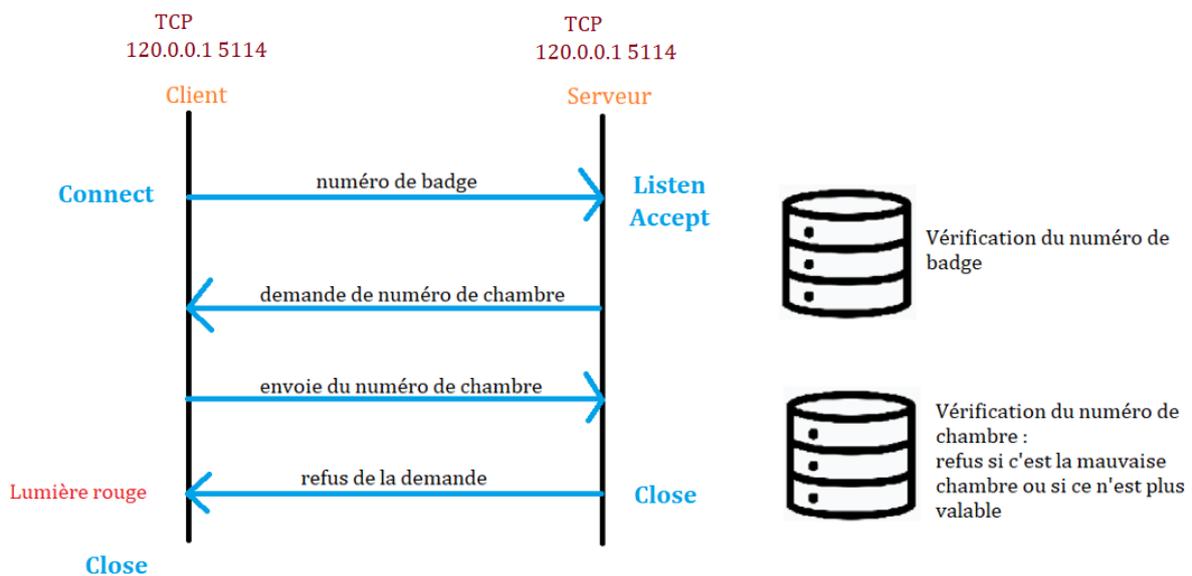


Figure 4: Image illustrant le cas où le serveur refuse l'accès au badge

Dans la base de données, nous stockons l'ensemble des comptes salariés et clients.

Le client pourra accéder à sa chambre avec le badge. Cependant, le badge peut ne pas être détecté par le serveur et ne renvoie aucune indication et le client pourra réessayer.

Dans le cas où, le client présente un mauvais badge, le serveur lui donnera en total 3 essais, après lesquels, la carte sera bloquée et ne fonctionnera plus.

Le dysfonctionnement des badges peut être également causés par la perte du badge (dans ce cas, une clé sera proposée comme alternative), la coupure d'électricité (absence de validation de badge), la coupure d'internet (vérification impossible) et la validité du badge : date limite (dépend de la réservation du client).

F. Echanges Réseaux

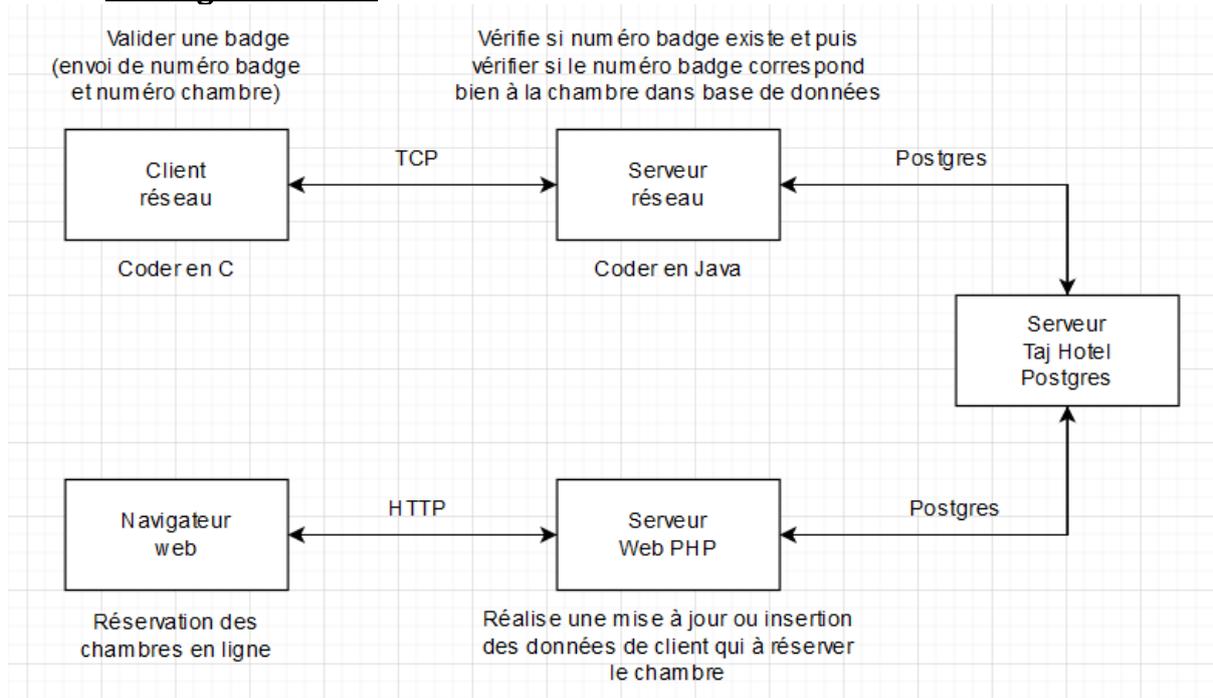


Figure 5: Image illustrant les échanges réseaux

G. Programmation Réseau (Sockets)

Les sockets permettent l'établissement des connexions entre une application exécutée sur une machine et une autre lancée sur une deuxième machine. La conceptualisation de socket est composée de la structure de données contenant des informations requises pour la communication et des appels manipulant la structure du socket.

Après la création du socket, il attend une connexion de la part d'un client. Ainsi, les échanges pourront se faire entre le serveur et le client.

Socket côté Serveur

Nous avons codé le côté serveur en langage **JAVA** suivant le protocole TCP car il est le plus sécurisé.

Une fois que le lancement de serveur est fait, il établit la connexion et attend, de la part d'un client, que les requêtes arrivent sur le réseau.

Une fois connecté à la base de données, nous la transmettons les requêtes.

Le rôle primordial du serveur est de procéder à la vérification de l'existence du numéro de badge et il contrôle également s'il correspond au numéro de chambre de la pièce que le client souhaite accéder. Cependant, trois cas sont possibles :

- La carte du client n'est pas détectée et le serveur ne répond rien.

- Si le badge est faux ou si le numéro de chambre est faux, alors nous renvoyons une lumière rouge. Au bout de trois tentatives, la carte du client sera bloquée et n'aura pas accès à la chambre.
- Si l'ensemble des informations sont correctes, alors nous renvoyons une lumière et le client aura l'accès à la chambre correspondante.

Socket côté Client

Nous avons codé le côté client en langage C suivant le protocole TCP car il est le plus sécurisé.

Une fois le lancement du client, il entre en connexion avec le serveur. Le client envoie alors un numéro de badge au serveur qui va, à son tour, vérifier et ensuite, le client envoie le numéro de chambre qui sera également vérifié par le serveur.

Si l'ensemble des données envoyées par le client est valide, alors un message indiquant « Lumière vert » apparaît pour le client, ce qui signifie que le client a bien été autorisé pour accéder à la chambre. Si « Lumière rouge » est affiché, alors le client n'y est pas autorisé et au bout de 3 essais, la carte sera bloquée.

III. DEROULEMENT DU PROJET

A. Diagramme de Gantt

Nom	Date de début	Date de fin
• contexte du sujet	12/09/2022	16/09/2022
• début de modélisation	16/09/2022	19/09/2022
• dictionnaire de donnée	19/09/2022	30/09/2022
• schema E/A	19/09/2022	30/09/2022
• partie DDL du projet	30/09/2022	14/10/2022
• connexion BDD	30/09/2022	14/10/2022
• partie DML du projet	14/10/2022	28/10/2022
• client et serveur reseau	14/10/2022	28/10/2022
• finalisation et tests	28/10/2022	04/11/2022
• rapport du projet	04/11/2022	18/11/2022
• 2 videos de demonstration	18/11/2022	25/11/2022
• préparation oral	25/11/2022	02/12/2022

Figure 6 : schéma représentant les tâches à effectuer

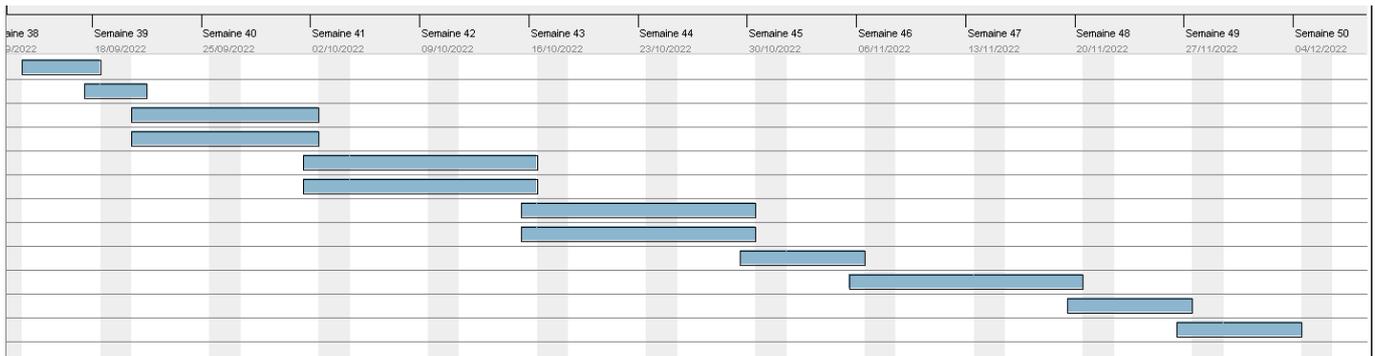


Figure 7 : schéma représentant les dates limites correspondantes aux tâches

Le diagramme de Gantt nous permet d'obtenir une vision plus globale et claire des objectifs du projet, leurs responsabilités et leurs niveaux d'importance. Nous avons fait un diagramme de Gantt afin d'étudier les niveaux de faisabilités.

B. Répartitions des tâches

Répartitions des tâches

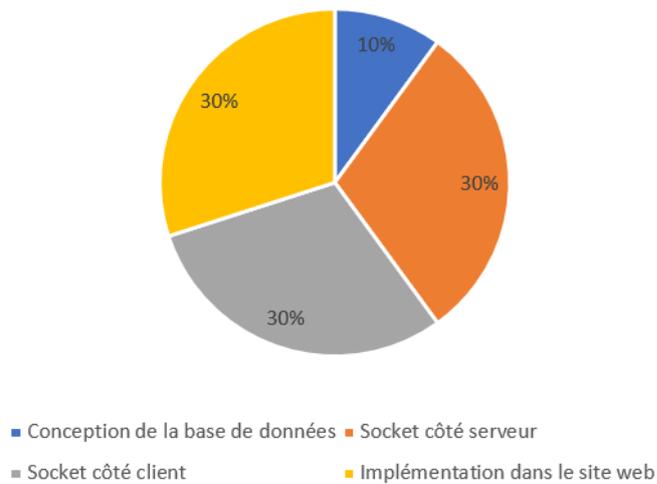


Figure 8 : Schéma montrant la répartition des tâches

Cette figure représente la répartition des tâches de chacune des grandes étapes du projet. Il s'agit d'une conclusion graphique résultant de notre gestion de projet.

Répartitions des tâches entre membres de l'équipe

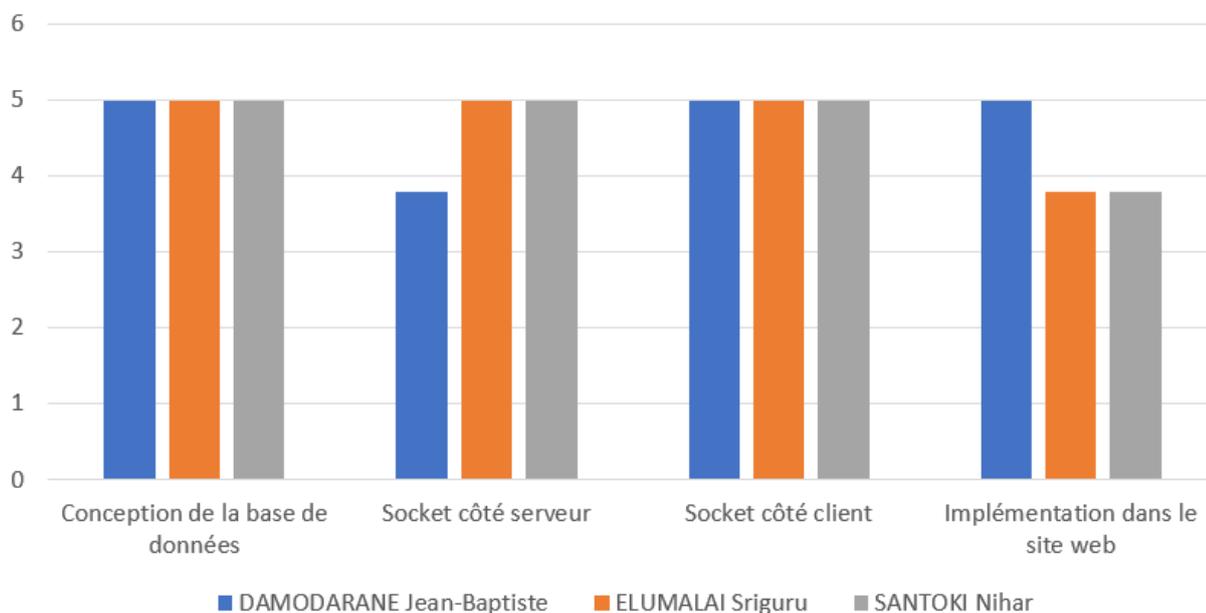


Figure 9 : Schéma illustrant la répartition des tâches entre les membres de l'équipe

Cette figure représente les taux travaux effectués par les membres du groupe.

VI. CONCLUSION ET PERSPECTIVES

A. Résumé du travail réalisé

Résumé du projet

Notre projet consiste en la création d'un échange entre le client réseau et le serveur réseau suivant le protocole TCP.

En fait, l'échange repose sur le serveur qui attend la connexion d'un client sur le même port. Une fois connecté, les échanges se déroulent : le client envoie le numéro et puis le numéro de chambre au serveur qui les traitera et renvoie un message d'acceptation ou de refus en fonction des données de la base de données.

Notre [site web](#) présente les chambres disponibles dans l'hôtel et une liste des chambres de luxe. Nous avons également implémenté un système permettant aux clients d'en réserver en entrant les données nécessaires.

Défis rencontrés

Nous avons dû faire face à des problèmes d'échanges entre le client et le serveur, du fait que le serveur n'a pas été en position de recevoir des messages émis par le client. Mais après de nombreux essais, nous avons réussi à procéder aux échanges.

B. Améliorations possibles

Selon nous, nous devons améliorer l'activation et la désactivation des badges. En effet, dans le contexte de notre projet, le badge est activé ou désactivé lorsqu'un employé de l'hôtel modifie la table badge en changeant la valeur des droits d'accès en FALSE ou en TRUE. Alors cela pourrait être amélioré en faisant de tel sorte que les badges soient activés ou désactivés sans l'intervention sur la table. Pour cela, nous pourrions activer et désactiver le badge par rapport à la période de réservation du client, autrement dit le badge sera automatiquement activé dès le premier jour de la réservation du client et sera désactivé après son départ. L'intervention sur la table sera faite seulement dans le cas où la carte est bloquée, volée ou perdue.

D'autre part, une autre amélioration concernant les badges peuvent être apportée. En fonction des besoins de l'hôtel, nous pouvons modifier l'attribution des badges. Quant à nous, nous avons conçu les badges de tel sorte que chaque chambre ne peut être accédée que par un badge unique correspondant à cette chambre. De plus, plusieurs badges peuvent être accordés à un groupe de clients réservant ensemble la même chambre. Cependant, les badges utilisés par les employés ont accès à la majorité des chambres.