|  |
| --- |
|  |
| VIVEHOTEL : SITE HÔTELIER |
| Document de conception technique |

|  |
| --- |
| EL AMRANI Bilel, OUANYOU Djé Arsène, VONIN Cédric  11/04/2023 |

Table des matières

[Description de l’application 2](#_Toc132119129)

[Environnement technique 3](#_Toc132119130)

[Langages de programmation 3](#_Toc132119131)

[Système d’exploitation 3](#_Toc132119132)

[Framework 3](#_Toc132119133)

[Serveur d’application 3](#_Toc132119134)

[Base de données 3](#_Toc132119135)

[Description de la base de données 4](#_Toc132119136)

[Le modèle conceptuel de données (MCD) 4](#_Toc132119137)

[Textuel 4](#_Toc132119138)

[Graphique 7](#_Toc132119139)

[Le modèle logique de données (modèle relationnel) 7](#_Toc132119140)

[Textuel 7](#_Toc132119141)

[Graphique 10](#_Toc132119142)

# Description de l’application

Le groupe ViveHotel possède un réseau comprenant une cinquantaine d’hôtels répartis sur le territoire français.

ViveHotel souhaite disposer d’un système d’information centralisé permettant de gérer l’ensemble des réservations pour tout hôtel de son réseau.

Le système doit fournir les services suivants :

* Gestion des hôtels et des chambres.
* Gestion des réservations.
* Consultation des disponibilités des chambres.
* Calcul du chiffre d’affaire annuel.
* Production de diverses statistiques :
  + Chiffre d’affaire par hôtel, par département.
  + Nombre de réservations par durées.
  + …etc.

ViveHotel dispose d’un Centre d’appels au SRC (Service de Réservation Centrale). Un numéro vert unique (le même pour tous les hôtels) est mis en place pour toutes demandes de réservation par téléphone, mais les clients peuvent également réserver directement auprès des hôtels ou en ligne sur le site internet de ViveHotel.

Que la réservation soit réalisée par un opérateur ou directement sur internet par le client lui-même, un compte client est créé (s’il n’existe pas) avec la création d’une réservation.

# Environnement technique

## Langages de programmation

* JavaScript ES6
* PHP 8.2
* SQL

## Système d’exploitation

* Windows 10

## Framework

* Bootstrap (v5.0.0-beta2)
* Framework PDO MVC BOOTSTRAP (3.0)

## Serveur d’application

* XAMPP Control Panel Version 3.3.0 et Apache 2.4.54

### Base de données

* PhpMyAdmin 5.2.0 (?)

# Description de la base de données

L’utilisateur de la base de données principal est le profil « root » n’ayant aucun mot de passe. Il a le droit de lecture, écriture sur la base de données du projet.

La base de données peut être généré à partir du sous-dossier « \_dataset » (à supprimer en production). Si la base de données existait déjà, elle sera alors réinitialisée par le fichier « index.php » de ce dossier.

La liste des utilisateurs de la base de données avec PhpMyAdmin est donnée par le tableau suivant :

|  |  |
| --- | --- |
| Nom d’utilisateur | Profil |
| Root | SU |

Les sections de ce chapitre vont indiquer l’ensemble des schémas de données créés pour la base de données du projet.

## Le modèle conceptuel de données (MCD)

### Textuel

<MCD>  
 <entités>

\*Client  
- id (AI)  
- nom (varchar(500))  
- identifiant (varchar(500))  
- mot\_de\_passe (varchar(500))  
- email (varchar(500))

\* Personnel  
- id (AI)  
- nom (varchar(500))  
- idenifiant (varchar(500))  
- mot\_de\_passe (varchar(500))  
- email (varchar(500))

\* Hotel  
- id (AI)  
- statut (int)  
- nom (varchar(500))  
- adresse (varchar(500))  
- departement (int)  
- description (text)  
- longitude (float)  
- latitude (float)

\*Chambre  
- id (int)  
- numero (int)  
- statut (varchar(500))  
- surface (int)  
- type\_lit\_un (varchar(500))  
- type\_lit\_deux (varchar(500) or NULL)  
- description (text)  
- jacuzzi (bool)  
- balcon (bool)  
- wifi (bool)  
- minibar (bool)  
- coffre (bool)  
- vue (bool)  
  
\*Service  
- id (AI)  
- nom (varchar(500))

\*ChCategorie  
- id(AI)  
- categorie\_chambre (varchar(500))

\* HoCategorie

- id (AI)  
- categorie\_hotel (varchar(500))

\* Reservation  
- Id(AI)  
- date\_creation (datetime)  
- date\_debut (datetime)  
- date\_maj (datetime)  
- date\_fin (datetime)  
- etat (varchar(500))  
</entités>

<association>  
Prendre  
- Client 1,n  
- Réservation 1,1

Conclure  
- Reservation 1,1  
- Hotel 1,n

Preciser  
- Chambre 1,n  
- Reservation 1,1

Commander (quantité)  
- Service 1,n   
- Reservation 1,n

Travailler (role)  
- Personnel 0,1  
- Hotel 1,n

Proposer (prix)  
- Hotel 0,n  
- service 0,n

Tarifer (prix)  
- HoCategorie 1,n  
- ChCategorie 1,n

HoAppartenir  
- hotel 1,1  
- HoCategorie 1,n

ChAppartenir  
- Chambre 1,1  
- ChCategorie 1,n  
</association>

</MCD>

### Graphique



*MCD graphique*

## Le modèle logique de données (modèle relationnel)

### Textuel

<MLD>  
client  
 - cli\_id (AI)  
 - cli\_nom (varchar(500))  
 - cli\_identifiant (varchar(500))  
 - cli\_mdp (varchar(500))  
 - cli\_email (varchar(500))

personnel  
 - per\_id (AI)  
 - per\_nom (varchar(500))  
 - per\_idenifiant (varchar(500))  
 - per\_mdp (varchar(500))  
 - per\_email (varchar(500))  
 - per\_role (varchar(500))  
 - per\_hotel (FK)

hotel  
 - hot\_id (AI)  
 - hot\_statut (varchar(500))  
 - hot\_nom (varchar(500))  
 - hot\_adresse (varchar(500))  
 - hot\_departement (int)  
 - hot\_description (text)  
 - hot\_longitude (float)  
 - hot\_latitude (float)  
 - hot\_hocategorie (FK)

chambre  
 - cha\_id (AI)  
 - cha\_numero (int)  
 - cha\_statut (varchar(500))  
 - cha\_surface (int)  
 - cha\_typelit1 (varchar(500))  
 - cha\_typelit2 (varchar(500) ou « NULL »)  
 - cha\_description (text)  
 - cha\_jacuzzi (bool)  
 - cha\_balcon(bool)  
 - cha\_wifi (bool)  
 - cha\_minibar (bool)  
 - cha\_coffre (bool)  
 - cha\_vue (bool)  
 - cha\_chcategorie (FK)

service  
 - ser\_id (AI)  
 - ser\_nom (varchar(500))

chcategorie  
 - chc\_id (AI)  
 - chc\_categorie (varchar(500))

hocategorie  
 - hoc\_id (AI)  
 - hoc\_categorie (varchar(500))

reservation  
 - res\_id (AI)  
 - res\_datecreation (datetime)  
 - res\_datedebut (datetime)  
 - res\_datemaj (datetime)  
 - res\_datefin (datetime)  
 - res\_etat (varchar(500))  
 - res\_client (FK)  
 - res\_hotel (FK)  
 - res\_chambre (FK)

commander   
 - com\_id (AI)  
 - com\_quantite (int)  
 - com\_service (FK)  
 - com\_reservation (FK)

proposer  
 - pro\_id (AI)  
 - pro\_prix (float)  
 - pro\_hotel (FK)  
 - pro\_service (FK)

tarifer  
 - tar\_id (AI)  
 - tar\_prix (float)  
 - tar\_hocategorie (FK)  
 - tar\_chcategorie (FK)  
</MLD>

### Graphique



*Modèle logique de données graphique*

**Script SQL :** voir “vivehotel.sql”

# Description des pages

## Contrôleur par défaut

### Page index

La page d’index contient une brève description de l’hôtel ainsi que quelques hôtels suggérés à l’utilisateur.

(Impression écran)

Lorsque l’utilisateur (connecté ou non) clique sur le lien de l’un des hôtels, il va pouvoir choisir d’en réserver un sur les pages suivantes.

### Page index