

## Création de la base de données

```
CREATE DATABASE IF NOT EXISTS hypnos_bdd;
```

## Création des différentes tables

```
CREATE TABLE IF NOT EXISTS User_Ref(  
    ID INTEGER NOT NULL AUTO_INCREMENT,  
    Label VARCHAR(250) NOT NULL,  
    PRIMARY KEY(ID)  
);
```

```
CREATE TABLE IF NOT EXISTS User(  
    ID INTEGER NOT NULL AUTO_INCREMENT,  
    Name VARCHAR(250) NOT NULL,  
    Firstname VARCHAR(250) NOT NULL,  
    Mail VARCHAR(250) NOT NULL,  
    Password VARCHAR(250) NOT NULL,  
    Role INTEGER NOT NULL,  
    FOREIGN KEY(Role) REFERENCES User_Ref(ID),  
    PRIMARY KEY(ID)  
);
```

```
CREATE TABLE IF NOT EXISTS Establishment(  
    ID INTEGER NOT NULL AUTO_INCREMENT,  
    Address VARCHAR(250) NOT NULL,  
    City VARCHAR(250) NOT NULL,  
    Name VARCHAR(250) NOT NULL,  
    Manager INTEGER NOT NULL,  
    FOREIGN KEY(Manager) REFERENCES User(ID),  
    PRIMARY KEY(ID)  
);
```

```
CREATE TABLE IF NOT EXISTS Suite(  
    ID INTEGER NOT NULL AUTO_INCREMENT,  
    Title VARCHAR(250) NOT NULL,  
    Establishment INTEGER NOT NULL,  
    Price DECIMAL NOT NULL,  
    IsOccupied BOOLEAN NOT NULL,  
    BookingLink VARCHAR(250) NOT NULL,  
    Description VARCHAR(1000) NOT NULL,  
    MainPic VARCHAR(250) NOT NULL,  
    Gallery VARCHAR(250) NOT NULL,  
    Gallery2 VARCHAR(250) NOT NULL,  
    Gallery3 VARCHAR(250) NOT NULL,  
    FOREIGN KEY(Establishment) REFERENCES Establishment(ID),  
    PRIMARY KEY(ID)  
);
```

```
CREATE TABLE IF NOT EXISTS SubjectMessage_Ref(  
    ID INTEGER NOT NULL AUTO_INCREMENT,  
    Label VARCHAR(250) NOT NULL,  
    PRIMARY KEY(ID)  
);
```

```
CREATE TABLE IF NOT EXISTS Message(  
    ID INTEGER NOT NULL AUTO_INCREMENT,  
    Name VARCHAR(250) NOT NULL,  
    Firstname VARCHAR(250) NOT NULL,  
    Mail VARCHAR(250) NOT NULL,  
    MessageBody VARCHAR(1000) NOT NULL,  
    Recipient INTEGER NOT NULL,  
    Subject INTEGER NOT NULL,
```

```
FOREIGN KEY(Recipient) REFERENCES User(ID),  
FOREIGN KEY(Subject) REFERENCES SubjectMessage_Ref(ID),  
PRIMARY KEY(ID)  
);
```

```
CREATE TABLE IF NOT EXISTS Booking(  
ID INTEGER NOT NULL AUTO_INCREMENT,  
StartingDate DATE NOT NULL,  
EndingDate DATE NOT NULL,  
Suite INTEGER NOT NULL,  
Client INTEGER NOT NULL,  
FOREIGN KEY(Suite) REFERENCES Suite(ID),  
FOREIGN KEY(Client) REFERENCES User(ID),  
PRIMARY KEY(ID)  
);
```

### **Création des roles Utilisateurs.**

```
INSERT INTO User_Ref (Label) VALUES('Admin');  
INSERT INTO User_Ref (Label) VALUES('Manager');  
INSERT INTO User_Ref (Label) VALUES('Client');
```

### **Création de l'administrateur**

```
INSERT INTO User(Name, Firstname, Mail, Password, Role) VALUES('Doerr', 'Zoran',  
'doerrzoran@gmail.com', 'motdepasse', 1);
```

### **Création des gérants**

```
INSERT INTO User(Name, FirstName, Mail, Password, Role) VALUES('Duclain', 'Bob',  
'bob.duclain@mail.com', 'motdepasseBD', 2);  
INSERT INTO User(Name, FirstName, Mail, Password, Role) VALUES('Vison', 'George',  
'george.vison@mail.com', 'motdepasseGV', 2);  
INSERT INTO User(Name, FirstName, Mail, Password, Role) VALUES('Lemaitre', 'Pierre',  
'pierre.lemaitre@mail.com', 'motdepassePL', 2);
```

```
INSERT INTO User(Name, FirstName, Mail, Password, Role) VALUES('Boutella', 'Salim',  
'salim.boutella@mail.com', 'motdepasseSB', 2);
```

```
INSERT INTO User(Name, FirstName, Mail, Password, Role) VALUES('Wise', 'John',  
'john.wise@mail.com', 'motdepasseJW', 2);
```

```
INSERT INTO User(Name, FirstName, Mail, Password, Role) VALUES('Da Costa', 'Ryan',  
'ryan.dacosta@mail.com', 'motdepasseRD', 2);
```

```
INSERT INTO User(Name, FirstName, Mail, Password, Role) VALUES('Vigneux', 'Armand',  
'armand.vigneux@mail.com', 'motdepasseAV', 2)
```

## **Création des établissements**

```
INSERT INTO Establishment(Address, City, Name, Manager) VALUES('31 rue de la voie', 'Paris 5em',  
'Hotel Hypnos Paris', 2);
```

```
INSERT INTO Establishment(Address, City, Name, Manager) VALUES('25 rue des mouettes',  
'Bordeaux', 'Hotel Hypnos Bordeaux', 3);
```

```
INSERT INTO Establishment(Address, City, Name, Manager) VALUES('17 rue du phare', 'saint-malo',  
'Hotel Hypnos Saint-Malo', 4);
```

```
INSERT INTO Establishment(Address, City, Name, Manager) VALUES('29 rue du fort', 'La rochelle',  
'Hotel Hypnos La rochelle', 5);
```

```
INSERT INTO Establishment(Address, City, Name, Manager) VALUES('60 rue des cygales', 'Nice', 'Hotel  
Hypnos Nice', 6);
```

```
INSERT INTO Establishment(Address, City, Name, Manager) VALUES('40 rue Bonchard', 'Cherbourg',  
'Hotel Hypnos Cherbourg', 7);
```

```
INSERT INTO Establishment(Address, City, Name, Manager) VALUES('5 allée de la rue', 'Bayonne',  
'Hotel Hypnos Bayonne', 8);
```

## **Création des suites**

```
INSERT INTO User_Ref (Label) VALUES('Admin');
```

```
INSERT INTO User_Ref (Label) VALUES('Manager');
```

```
INSERT INTO User_Ref (Label) VALUES('Client');
```

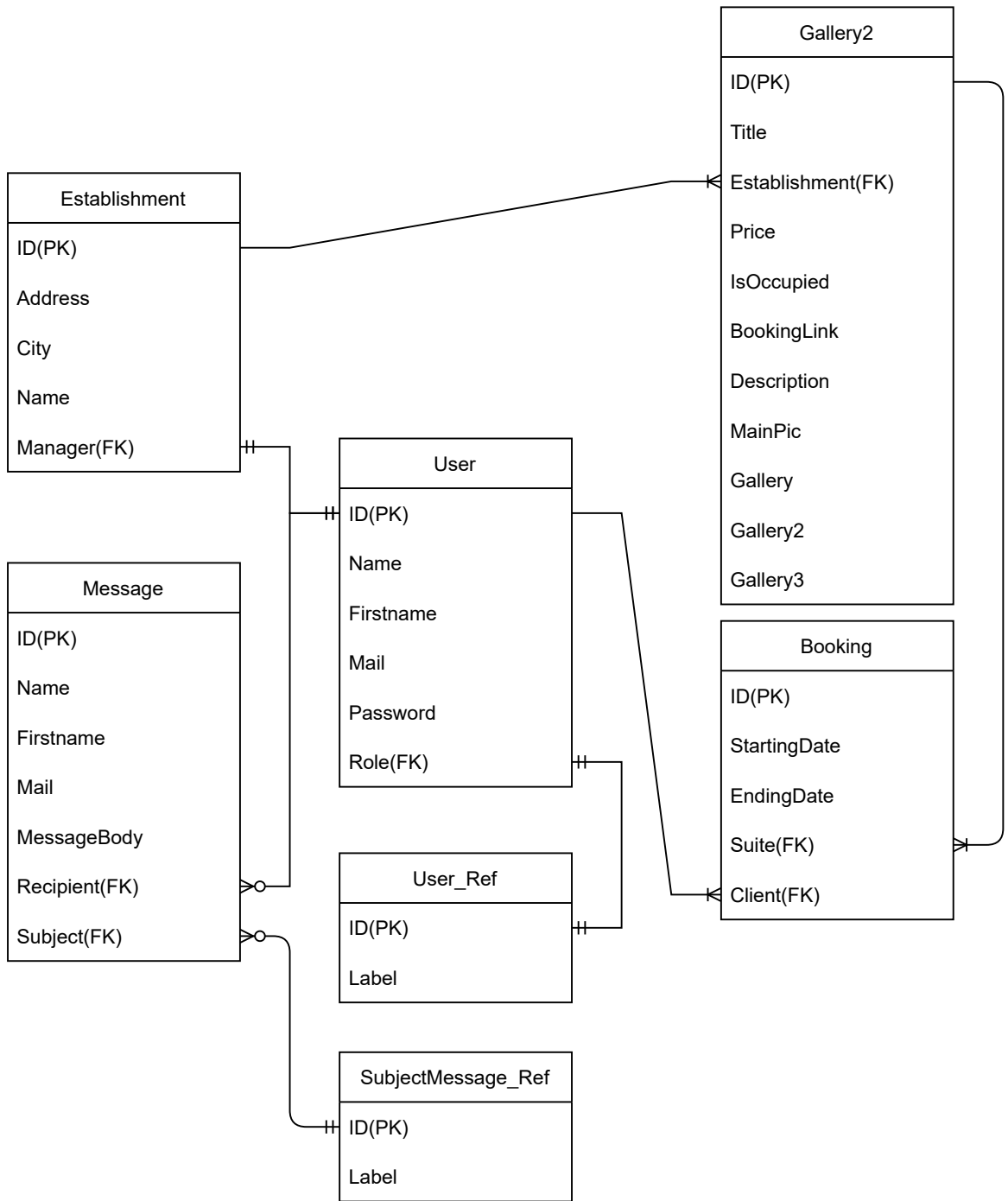
## **Création des sujets de message**

```
INSERT INTO subjectmessage_ref (Label) VALUES('Je souhaite poser une reclamation');
```

```
INSERT INTO subjectmessage_ref (Label) VALUES('Je souhaite commander un service  
supplémentaire');
```

```
INSERT INTO subjectmessage_ref (Label) VALUES('Je souhaite en savoir plus sur une suite');
```

```
INSERT INTO subjectmessage_ref (Label) VALUES("J'ai un soucis avec cette application");
```



## **Documentation technique**

Editeur de Texte : Visual Studio Code ; recommandé sur internet et gratuit, utilisé en milieu professionnel.

Serveur MAMP : Utilisé pour Apache, PHP et MySQL : expérience avec cet outil due à son utilisation lors de mes cours. Beaucoup de documentation en ligne.

Editeur de maquette : Balsamique Wireframe car recommandé par Studi.

Utilisation du Framework Bootstrap car largement recommandé sur les forums spécialisés.

## Modèle de copie : Créer une application web



**GDWFSCAWEXAIII1A**

**Ceci est un modèle de copie. N'oubliez pas de renseigner vos prénom/nom, ainsi que le nom et le lien vers le projet.**

**Vous pouvez bien sûr agrandir les cadres pour répondre aux questions sur la description du projet si nécessaire.**

**Prénom :** Zoran

**Nom :** Doerr

**ATTENTION ! PENSEZ À RENSEIGNER VOS NOM ET PRÉNOM DANS LE TITRE DE VOS FICHIERS / PROJETS !**

Nom du projet : ProjetHypnos Zoran Doerr

Lien Github du projet : <https://github.com/doerrzoran/ProjetHypnos.git>

URL du site (si vous avez mis votre projet en ligne) :  
<https://projethypnos.herokuapp.com>

### Description du projet

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions. Dans cette rubrique, le jury cherche à voir comment vous procédez : comment vous organisez votre travail, comment vous réalisez concrètement la tâche ou l'opération pas à pas. Utiliser un langage professionnel. Employez le « je », car vous parlez en votre nom. Vous pouvez écrire au temps présent.



2. Précisez les moyens utilisés. Expliquez tout ce dont vous avez eu besoin pour réaliser vos tâches : langages de programmation, frameworks, outils, logiciels, documentations techniques, etc...

Editeur de Texte : Visual Studio Code ; recommandé sur internet et gratuit, utilisé en milieu professionnel. Serveur MAMP : Utilisé pour Apache, PHP et MySQL : expérience avec cet outil due à son utilisation lors de mes cours. Beaucoup de documentation en ligne. Editeur de maquette : Balsamique Wireframe car recommandé par Studi. Utilisation du Framework Bootstrap car largement recommandé sur les forums spécialisés.

3. Contexte. Les noms des organismes, entreprises ou associations, dans lesquels vous avez exercé vos pratiques

NB: Pour le cas des exercices et évaluations demandées sur la plateforme Studi, il s'agit de...Studi.

4. Informations complémentaires (*facultatif*)

<https://trello.com/invite/b/Vh7mdGaK/a245014ebccf01cb092cc1c0a73170e4/project-jethypos>

# Manuel d'utilisation

## **Arrivée sur le site.**

Le visiteur se connectant à l'application Hypnos arrive sur la page d'accueil. S'il n'est pas connecté, il a comme options de s'identifier grâce à un formulaire demandant son identifiant et son mot de passe ou de créer un compte en cliquant sur un lien vers une page contenant un formulaire de création de compte.

Une fois connecté, l'utilisateur sera redirigé vers une page utilisateur. S'il retourne sur la page d'accueil en cliquant sur le titre HYPNOS en haut de chaque page, son nom et son prénom seront affichés ainsi qu'un bouton de déconnexion et un lien vers sa page d'utilisateur, à la place du formulaire d'identification et du lien pour créer un compte.

Sur la page d'accueil, l'utilisateur trouve également un lien pour écrire un message, et un lien vers un formulaire de réservation, ainsi qu'une liste déroulante des établissements qu'il peut consulter.

## **Page établissement**

Sur la page établissement, l'utilisateur peut voir la liste des suites avec le prix et les photos. Si la suite est disponible, en bouton « réserver » le renvoie sur la page de réservation.

## **Création du compte**

Sur la page d'inscription, l'utilisateur trouve un formulaire où lui sont demandés son nom, son prénom, son adresse mail, ainsi qu'un mot de passe à confirmer. Après avoir confirmé son inscription, il est redirigé vers la page d'accueil.

## **La page client**

Si l'utilisateur crée un compte manuellement, il est enregistré en temps que client. De ce fait, il est redirigé vers la page client où sont affichés, ses réservations. Si la date de début du séjour est dans plus de trois jours. Un bouton pour supprimer la réservation est présent.

Il y trouve également un lien vers la page de réservation et un lien vers la page de message.

## **La page gérant**

Si l'utilisateur a le statut de gérant, il peut voir sur la page utilisateur sur sa page utilisateur, la liste des suites de l'établissement dont il s'occupe. Pour chacune il peut changer leur statut (les rendre disponibles ou indisponibles).

Il peut ajouter ou supprimer des suites grâce à des formulaires dédiés.

## **La page administrateur**

Sur sa page, l'administrateur a accès à la liste de tous les établissements.

Il peut, ajouter, supprimer ou modifier un établissement, ainsi que des gérants, grâce à des formulaires dédiés.

## **Réserver**

Si l'utilisateur accède à la page de réservation directement depuis la page d'accueil ou la page client, il devra renseigner un établissement, ceci lui permettra de choisir une suite parmi celles de l'établissement. S'il accède à la page réservation depuis la page Etablissement, l'établissement et la suite seront déjà renseignés.

S'il n'est pas identifié, il sera redirigé vers une page avec un formulaire d'identification et un lien vers la page de création de compte.

### **Ecrire un message**

L'utilisateur peut se rendre sur la page de message ou il peut remplir un formulaire lui demandant son nom, son prénom et son adresse mail. Il lui sera demandé de choisir un sujet parmi ceux proposés avant d'écrire son message et de l'envoyer. Après quoi il sera redirigé vers la page client.

J'ai créé 7 tables :

La table *User* permet de stocker toutes les données relatives à chaque utilisateur de l'application. Cette table reçoit pour une clé étrangère la table *User\_Réf*. Cette dernière est la table qui liste les différents rôles possibles sur l'application. Cette relation est de type « 1 to 1 » car chaque utilisateur ne peut avoir qu'un rôle.

Dans les rôles enregistrés il n'y a que *Client*, *Admin* et *Gérant*. Le rôle *Visiteur* n'étant pas traité car aucune donnée n'est enregistrée pour cet utilisateur.

La Table *User* constitue également une clé étrangère pour les tables *Establishment*, *Booking* et *Message*. Mais pas pour la table *Suite*, car celle-ci sert déjà comme clé étrangère pour la table *Booking*.

La table *Booking* ne prend d'ailleurs pas comme clé étrangère *Establishment* car celle-ci sert déjà de clé étrangère pour la table *Suite* à travers une relation de type : « 1 to many », car un Etablissement peut contenir plusieurs Suites.

De même, *Suite* est liée à la table *Booking* par une relation de type : « 1 to many », car une suite peut faire l'objet de plusieurs réservations.

Il en va de même pour *User* et *Booking* car un utilisateur peut effectuer plusieurs réservations.

Cependant, *User* n'est lié à *Establishment* que par une relation « 1 to 1 » car un établissement ne comporte qu'un seul gérant.

La table *Suite* comporte un champ *isOccupied* lequel indique si la suite est disponible.

La table *Establishment* a également pour clé étrangère la table *Service*. Cette table détermine le type de service fournis par l'établissement.

On retrouve *Service* comme clé étrangère dans la table *Booking*. Ces services étant optionnels et leur quantité étant variable selon les établissements, la table *Service* est liée aux tables *Establishment* et *Booking* par des liens de type « many optional to many optional ».

La table *Message* Comporte les champs : *Name*, *Firstname* et *Mail* au lieu d'un éventuel champ *expéditeur* qui serait une clé étrangère *User*, car un message doit pouvoir être envoyé par un simple visiteur dont les données ne seraient pas préalablement enregistrées.

A l'inverse, le champ *Recipient* prend comme valeur une clé étrangère *User* car chaque message est adressé à l'administrateur de l'application. Ce lien est de type « 1 mandatory to many optional » car seul l'administrateur peut recevoir des messages, et il peut en recevoir plusieurs, un seul ou aucun.

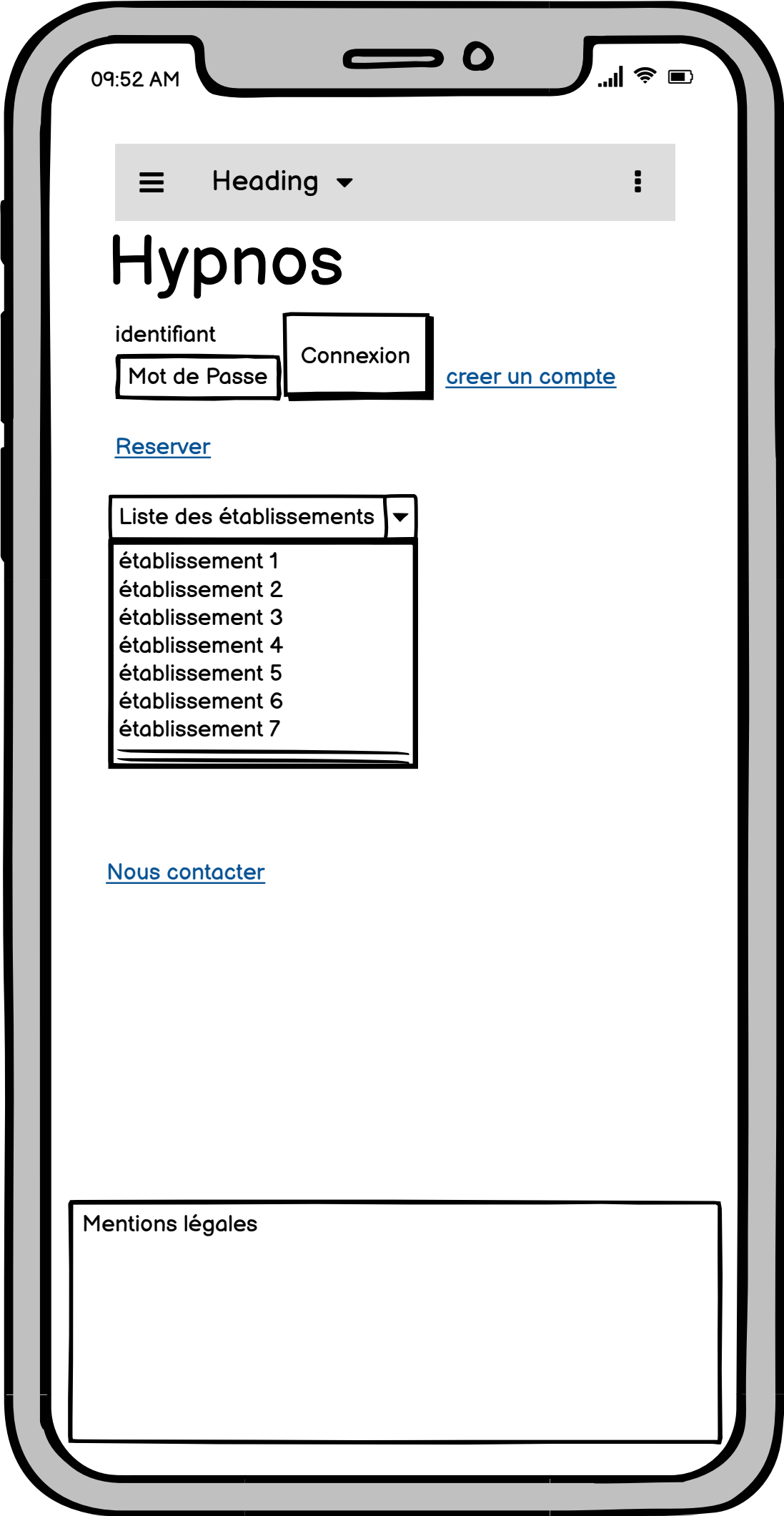
L'autre clé étrangère de la table *Message*, est *Subject*, celle-ci est définie par la table *SubjectMessage\_Ref*, laquelle détermine le sujet du message parmi cinq possibilités. Ce lien est de type « 1 to 1 » car un message n'a qu'un seul sujet.

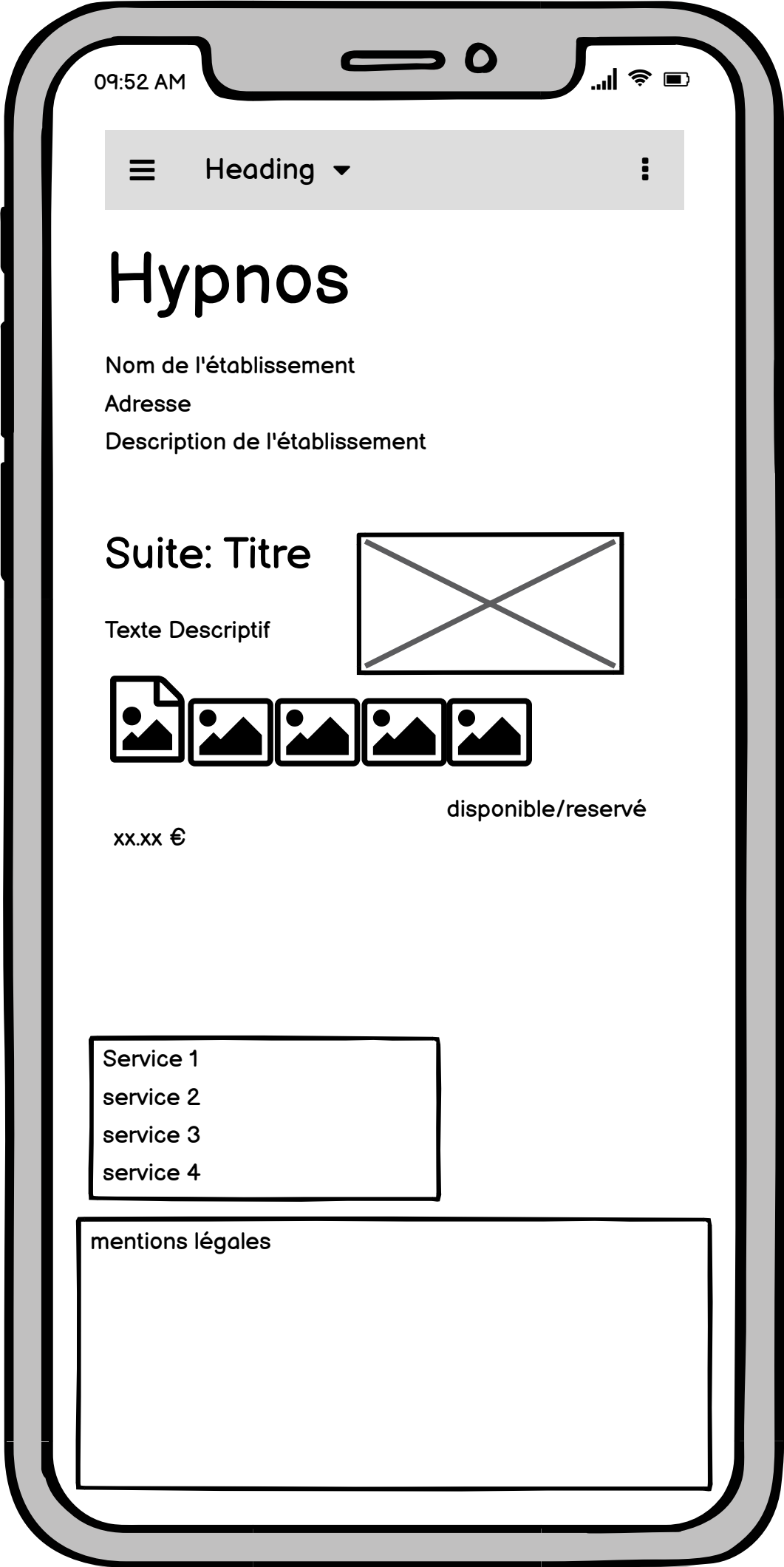
Palette graphique :

\_background: "bg-warning";

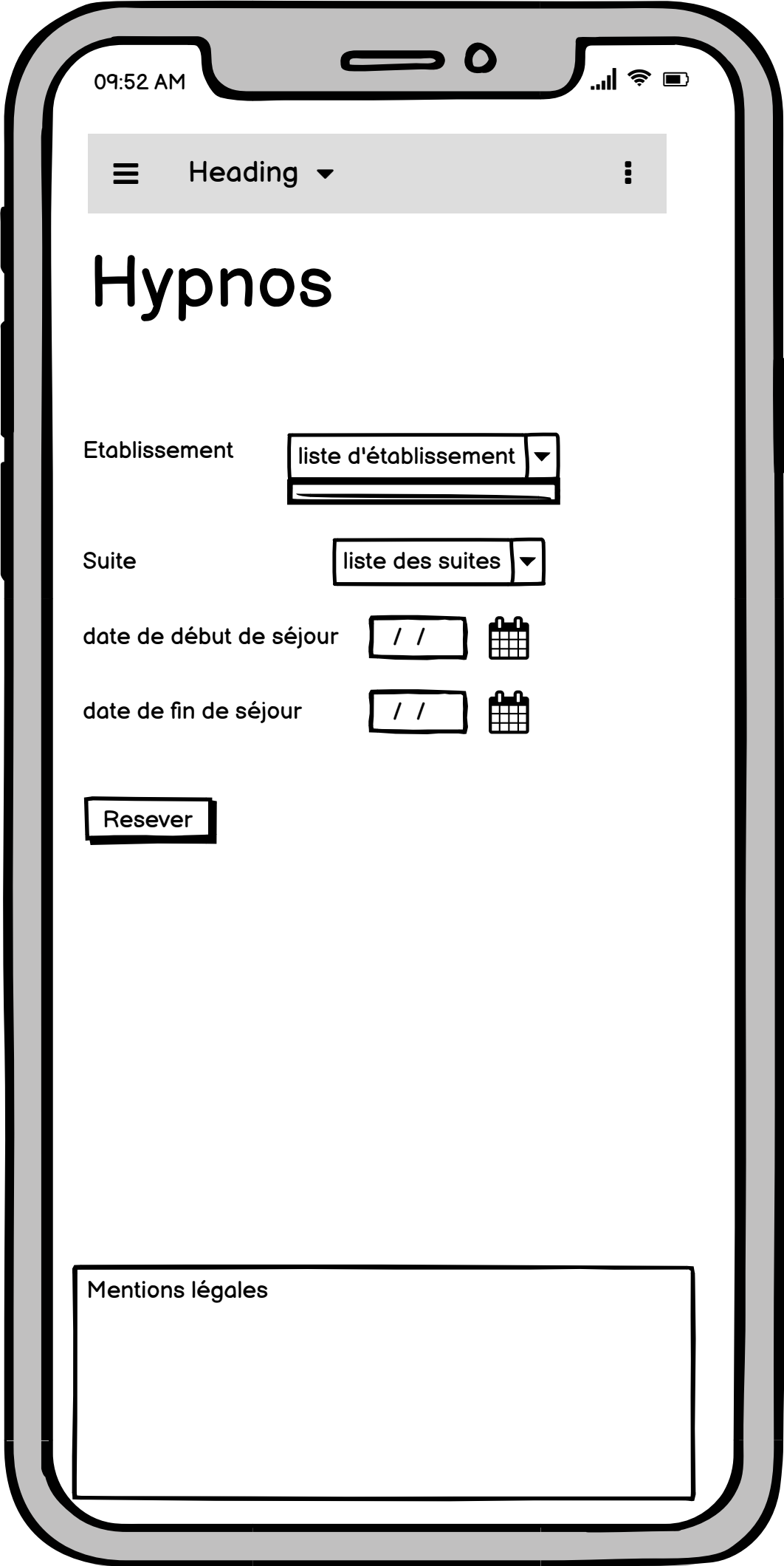
\_header: "bg-info";

\_footer "bg-primary";









09:52 AM

☰

Heading ▾

⋮

# Hypnos

Nom

nom

Prenom

prenom

Adresse email

adresse email

Mot de passe

mot de passe

confirmer mot de passe

mot de passe

Créer un compte

mentions légales



09:52 AM

Heading

Hypnos

Nom

Prenom

adresse email

objet

choisissez un objet

écrivez votre message

Envoyer

mentions légales