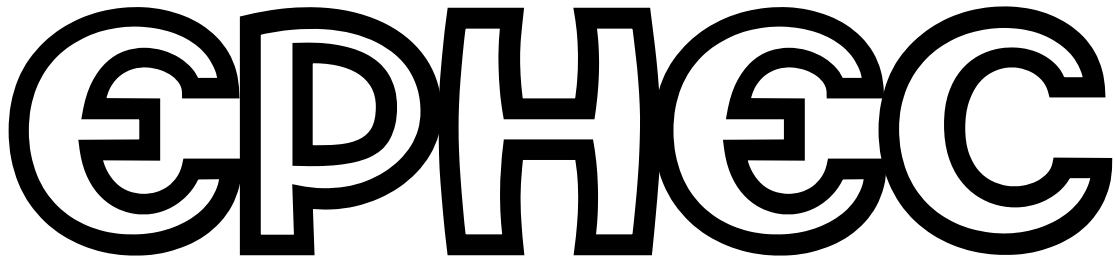


**Ecole Pratique des Hautes Etudes Commerciales**



**Institut d'Enseignement Supérieur Economique de Type Court**

**« STARFISH HOTELS »,  
APPLICATION WEB MOBILE  
DE RÉSERVATION DE SÉJOURS TOURISTIQUES**

**Farid ID BOURHIM**

**Rapporteur(s)  
J. HECQUET**

Travail de Fin d'Études présenté  
en vue de l'obtention du diplôme de  
**BACHELIER EN INFORMATIQUE DE GESTION**

**Année académique 2017-2018**

# Remerciements

---

J'adresse mes remerciements à toutes les personnes qui m'ont aidé de près ou de loin dans la réalisation de ce travail.

En premier lieu, je remercie Monsieur Hecquet, professeur à l'Ephec, pour le temps et l'aide précieuse qu'il m'a consacré.

J'adresse un remerciement particulier à Madame Brancart pour avoir accepté de lire et corriger mon travail de fin d'études.

Je remercie également toute l'équipe pédagogique et administrative de l'Ephec pour leur disponibilité et collaboration.

Je tiens à remercier particulièrement mon épouse pour son soutien et ses encouragements durant tout mon cursus.

# Table des matières

---

Introduction .....	3
Première partie : Aspects théoriques .....	4
Introduction .....	5
Chapitre I : Contexte .....	5
Section I : Le client .....	5
Section II : Synthèse des besoins .....	5
Section III : Le résultat attendu .....	6
Chapitre II : Cahier de charge .....	7
Section I : Elaboration .....	7
Section II : Fonctionnalités retenues .....	7
Section III : Architecture de l'application .....	11
Section IV : Infrastructure technique .....	12
Synthèse .....	13
Seconde Partie : Méthodologie et analyse .....	14
Introduction .....	15
Chapitre I : Méthodologie .....	15
Chapitre II : Analyse .....	15
Section I : Diagramme des cas d'utilisation .....	16
Section II : Règles métier .....	17
Section III : Notion de rôle et Autorisations .....	17
Section IV : Scénarios des cas d'utilisation .....	18
Section V : Diagrammes de séquences .....	22
Section VI : Analyse des processus .....	27
Section VII : Structure des données .....	32
Synthèse .....	39
Troisième Partie : Réalisation .....	40
Introduction .....	41
Chapitre I : Base de données .....	41
Section I : Conception .....	41
Section II : Création .....	42
Section III : Gestion des flux de données .....	42
Chapitre II : Développement applicatif .....	45
Section I : Application web .....	45
Section II : Application mobile .....	46

Chapitre III : Le déploiement .....	47
Section I : Microsoft Azure.....	48
Section II : Microsoft app center.....	49
Synthèse.....	50
Conclusions et recommandations.....	51
Bibliographie.....	52
Annexes.....	53
Annexe II : Détail d'un hôtel .....	54
Annexe III : Page de réservation d'un hôtel .....	55
Annexe IV : Première page de paiement.....	56
Annexe V : Deuxième page de Paiement.....	56
Annexe VI : Aperçu des vues de l'application mobile .....	57

# Introduction

---

Dans le cadre de mon travail de fin d'études, je vais vous présenter les différentes étapes que mon projet informatique traverse avant d'être utilisé par les différents utilisateurs.

La réalisation des projets informatiques passe par plusieurs étapes majeures. La première se base sur l'analyse des besoins, cette phase est marquée par une série de réunions organisées avec le client afin de déterminer exactement l'objectif et le périmètre du projet.

Le rôle de cette phase est d'interpréter les différents besoins du client et de les traduire sous forme de fonctionnalités.

L'analyse des besoins effectuée, le projet entame sa deuxième phase qui porte sur la modélisation des fonctionnalités retenues, ce qui permettra aux développeurs de réaliser exactement ce que le client attend comme produit fini.

La phase finale concerne le développement et le déploiement des différentes solutions ainsi que les différents tests à réaliser.

# Première partie : Aspects théoriques

---

# Introduction

---

Cette partie est consacrée à une phase très importante dans le cycle d'un projet. Cette phase consiste à étudier l'entière des besoins du client car il est indispensable de bien comprendre les besoins et attentes du client avant de commencer l'analyse.

L'analyse des besoins se fait à travers l'organisation d'une ou plusieurs rencontres avec le client afin de garantir que la solution proposée traitera et répondra au maximum aux attentes des utilisateurs.

Les discussions tourneront autour des fonctionnalités de l'application. Celles-ci permettront à l'analyste d'établir le périmètre du projet et l'ensemble des fonctionnalités qui seront intégrées au projet.

A l'issue de ces réunions, l'analyste dresse un rapport de synthèse en concertation avec le client. Ainsi, le client prend connaissance de ce qui va être analysé et fait part de ses remarques éventuelles sur l'objet d'analyse.

Lorsque cette première phase est validée par le client, l'analyste passe à la phase suivante qui porte sur l'analyse fonctionnelle du projet.

## Chapitre I : Contexte

---

### Section I : Le client

« StarFish Hotels » est une société spécialisée dans la gestion des grands hôtels dans plusieurs pays à travers le monde.

Dans le cadre d'une démarche marketing visant à rapprocher le produit du client, la société « StarFish Hotels » a décidé d'intégrer le monde digital en proposant ses produits à ses clients sur une plateforme de réservation qui regroupe les différents établissements hôteliers gérés par celle-ci.

A travers la plateforme de réservation, « StarFish Hotels » vise à permettre à ses clients d'effectuer des réservations de chambres et autres services qu'un hôtel, géré par la société, propose.

La plateforme de réservation regroupe l'ensemble des établissements hôteliers que la société gère. Ceci permet de mieux gérer les flux de réservations et de garder en vue l'activité globale ou spécifique de chaque établissement.

### Section II : Synthèse des besoins

Après une analyse de l'activité de la société, ont été retenus les éléments suivants :

- « StarFish Hotels » propose divers services tels que l'hébergement, la restauration, l'animation, des activités sportives et aquatiques. Les tarifs des chambres sont influencés par les saisons qui varient entre « Haute » et « Basse ». Le type de saison est défini par la variation de la demande en matière de réservations. Le tarif de la

haute saison est appliqué quand un hôtel connaît une forte demande et celui de la basse saison est appliqué lors de la baisse de celle-ci. Les dates de début et de fin de chaque saison varient d'un pays à un autre.

- Des combinaisons de services sont parfois proposés en promotion sous forme de packages afin d'encourager les ventes.
- La société propose différentes formules culinaires, ces offres varient selon les établissements et les saisons.
- Les pays qui connaissent une grande fréquentation en matière de réservation sont présentés sous forme de meilleurs destinations.
- Une catégorisation des établissements est appliquée selon des thèmes auxquels appartiennent ces derniers. Proposer ces thèmes permet aux clients d'avoir une idée sur les catégories de clients qui fréquentent chaque établissement.  
Exemple : un hôtel qui a pour thème « famille » reçoit beaucoup de familles, ce qui indique la forte présence de clients « bas-âge », ceci est un point important pour les clients qui recherchent la tranquillité.
- La société désire mettre en place une plateforme qui permet de centraliser la gestion des réservations des différents hôtels qu'elle propose.
- La société désire mettre en place une application mobile qui pourrait effectuer des réservations à travers les appareils mobiles.

## Section III : Le résultat attendu

À travers ce projet, la société « StarFish Hotels » s'attend à disposer des différents livrables suivant :

### 1. Application web

L'application web fera office de vitrine en ligne. Celle-ci permettra aux clients potentiels de réserver des séjours touristiques dans les différents hôtels gérés par la société. A la demande du celle-ci, un client peut annuler sa réservation au plus tard 7 jours avant la date de son arrivée à l'hôtel réservé. Cette application disposera d'une console d'administration qui permet aux différents administrateurs et selon les droits qui leur seront octroyés, de gérer les éléments suivants :

- **Hôtels** : la gestion des hôtels consiste à pouvoir ajouter, modifier ou désactiver un hôtel ainsi que ce dernier propose.
- **Réservations** : la gestion des réservations consiste aussi à créer, modifier ou annuler une réservation.
- **Utilisateurs** : représenté par la création et l'attribution des droits au différents utilisateurs ainsi que la mise à jours des profils clients si nécessaire.
- **Promotions** : l'administrateur doit pouvoir créer des promotions et les lier aux établissements.



- **Notifications** : à travers la console d'administration, il doit être possible aux administrateurs d'envoyer directement des notifications vers les appareils équipés de l'application mobile dans le cadre des promotions ou informations.

## 2. Application mobile

Afin de rapprocher ses produits aux clients, la société « StarFish Hotels » s'attend aussi à disposer d'une application mobile pour permettre à ces clients de réserver leurs séjours depuis leurs appareils mobiles. Cette application doit assurer les fonctionnalités suivantes auprès des utilisateurs :

- Parcourir les différents hôtels proposés par la société.
- Effectuer des recherches d'hôtels selon les dates et le nombre de personnes désirées.
- Effectuer des réservations portant sur des chambres d'hôtels.
- Consulter les réservations effectuées et annuler celles qui rentrent dans les délais permettant d'effectuer cette action.
- Consulter leurs profils et pouvoir les mettre à jour.

# Chapitre II : Cahier de charge

---

## Section I : Elaboration

Vu le rapprochement entre les deux applications attendues, j'ai trouvé très intéressant de diviser mon projet selon les phases suivantes :

- **Phase de conception** : Une fois définies, les fonctionnalités retenues seront analysées les unes après les autres. Pendant cette phase, l'analyse des fonctionnalités de l'application web sera exécutée en parallèle à celles de l'application mobile vue les similitudes considérables entre les deux.
- **Phase de réalisation** : Cette phase sera consacrée au développement des solutions proposées. Contrairement à la première phase qui porte sur la conception, cette phase sera réalisée en premier lieu pour l'application web. Ce choix se justifie par le fait que l'exécution des tests nécessaires pour l'application mobile nécessite la présence d'une source de données. Les données requises par l'application seront communiquées par l'application web qui doit disposer d'une interface de programmation applicative (api).

## Section II : Fonctionnalités retenues

Sur base des besoins retenus et exprimés par le client, j'ai pu extraire les fonctionnalités documentées ci-dessous. Certaines fonctionnalités seront similaires pour l'application mobile et web alors que d'autres seront spécifiques à l'une ou à l'autre.

### 1. Authentification de l'utilisateur

Cette fonctionnalité sera implémentée pour les deux applications sauf qu'elle sera gérée différemment selon le type d'application. Le fonctionnement de cette fonctionnalité reste le même et permettra à l'utilisateur de se connecter à son compte personnel soit à l'aide de

compte interne (login interne) préalablement créé ou via le login externe selon les réseaux sociaux implémentés ;

- Authentification interne : ceci concerne l'implémentation de la fonctionnalité de connexion via un module qui permet de gérer les connexions des utilisateurs.
- Authentification externe : connexion avec le compte habituel du réseau social de l'utilisateur s'il est implémenté au niveau de l'application back-end. Ce mode de connexion facilite la vie de l'utilisateur final en lui évitant de créer un compte supplémentaire. Dans le cadre de cette fonctionnalité, j'ai proposé au client d'implémenter dans un premier temps la connexion pour trois des réseaux sociaux le plus utilisés tel que Google, Facebook et Microsoft. À sa première tentative de connexion, un compte sera créé au niveau de la base de données des utilisateurs via un module qui permet la gestion des connexions via les réseaux sociaux.

## 2. Gestion du profil utilisateur

Cette fonctionnalité sera identique pour les deux applications, sauf que l'application mobile passe par l'interface de programmation applicative pour ce qui concerne le traitement du profil utilisateur. Cette fonctionnalité consiste à offrir à l'utilisateur, une fois connecté, la possibilité de charger ou d'éditer l'ensemble des informations relative à son profil tel que :

- Informations du profil (nom, prénom, adresse e-mail...etc.)
- Changement de mot de passe

## 3. Gestion des hôtels

Ce groupe de fonctionnalités permet la gestion de l'ensemble des informations relatives aux hôtels. Une partie de cet ensemble sera dédié à l'utilisateur final alors que l'autre partie sera dédiée aux administrateurs.

### 3.1 Créer, modifier ou désactiver un hôtel

Ces fonctionnalités sont réservées aux administrateurs qui pourront ajouter un hôtel, le modifier ou le désactiver. Elles seront par conséquent disponibles dans la console d'administration.

### 3.2 Rechercher un hôtel

La recherche des hôtels est une fonctionnalité que nous rencontrerons dans les deux applications, elle permet d'afficher les hôtels selon les différents critères suivants :

- **Le pays** : représente le pays sélectionné et concerné par la recherche
- **La ville** : représente la ville sélectionnée et concernée par la recherche
- **Dates du séjour** : représentées par la date d'arrivée et de départ
- **Le nombre de personnes** : le nombre de personnes qui compte réserver des chambres peut impacter le résultat de la recherche.

Prenons le cas d'un hôtel qui dispose de 3 lits disponibles (1 chambre triple ou 3 chambres singles ou 1 double + 1 single) et qui ne sera pas affiché dans le résultat de recherche si le nombre d'adultes dépasse 3 personnes.

### *3.3 Afficher les hôtels*

Cette fonctionnalité gère le chargement et l’affichage de l’ensemble des hôtels depuis la base de données vers l’application mobile ou web. Les hôtels seront présentés sous forme de liste avec certains éléments qui représentent chacun des établissements présentés à l’utilisateur tels que :

- Le nom de l’hôtel
- Image
- Localisation (ville -Pays)
- Le tarif minimum

### *3.4 Afficher le détail d’un hôtel*

Chaque hôtel peut être affiché sous forme de page de détail. Cette page contient l’ensemble des informations relatives à cet établissement tel que :

- Nom de l’hôtel
- Coordonnées
- Descriptif complet de l’hôtel
- Les différents service gratuits/payants
- Galerie photos

## **4. Gestion des réservations**

Ce groupe de fonctionnalités concerne les traitements nécessaires pour les réservations.

### *4.1 Chargement des réservations*

Pour l’utilisateur, cette fonctionnalité sera implémentée dans les deux applications et concerne le chargement de toutes les réservations effectuées par celui-ci. Un utilisateur ne peut visualiser que ses propres réservations.

Le profil « administrateur » dispose d’un accès lui permettant de charger et visualiser l’ensemble des réservations. L’implémentation de cette fonctionnalité pour l’administrateur sera réalisée au niveau de l’application web seulement.

L’affichage des réservations au niveau de l’application web sera fait sous forme de liste avec les colonnes les plus importantes. Il sera toujours possible d’afficher le détail complet d’une réservation grâce à un simple clic.

Cependant, les résultats affichés au niveau de l’application mobile seront présentés sous forme de deux groupes de réservations :

- **Historique** : sous ce groupe seront regroupées toutes les réservations pour lesquelles la date d’arrivée est antérieure à la date du jour.
- **Réservations à venir** : ce groupe de réservations représente toutes les réservations pour lesquelles la date d’arrivée est postérieure à la date du jour.

## 4.2 Création de réservations

La création des réservations fait partie des fonctionnalités les plus importantes. Elle sera présente dans les deux applications (web et mobile) et permet à l'utilisateur final d'effectuer de nouvelles réservations.

Le processus de réservation commence par une vérification de disponibilité sur base de l'hôtel choisi, les dates et le nombre de personnes introduits. Un formulaire de réservation sera affiché, celui-ci comprendra les différentes catégories de chambres dont dispose l'hôtel sélectionné. Les quantités disponibles pour chaque catégorie seront le résultat de la vérification de la disponibilité.

Selon la quantité choisie pour une catégorie de chambre, un formulaire par chambre sera affiché pour pouvoir renseigner d'autres informations telles que le nom de l'occupant de la chambre, le nombre d'adultes et d'enfants pour cette chambre, l'arrangement ou le plan culinaire choisi ainsi que le besoin d'un lit supplémentaire pour enfants.

Une fois validée par l'utilisateur, la réservation sera enregistrée et les chambre seront affectées sous forme de transaction. Après l'affectation des chambres réservées, une vérification des chambres affectées sera réalisée, ceci consiste à s'assurer que chaque chambre ne serait pas réservée deux fois pour la même période. Le résultat de cette vérification décidera de la validation de la transaction si une ou plusieurs chambres n'ont pas été déjà réservées pour la même période, ou l'annulation de celle-ci si une ou plusieurs chambres ont déjà été prises. La suite du processus de réservation dépend du statut de la transaction. Si la transaction est validée, la réservation obtiendra le statut « Enregistrée » et le processus continuera, sinon toutes les opérations relatives à la création de la réservation et l'affectation des chambres seront annulées.

Si la transaction est validée, et que l'utilisateur est un client une redirection est effectué vers une plateforme de paiements afin de réaliser le paiement relatif à la réservation précédemment effectuée. La réussite ou l'échec de cette opération décideront de la confirmation ou de l'annulation de la réservation.

## 5. Gestion des paiements

Dans le cadre de ce projet, la gestion des paiements pour l'application web se fera à travers l'intégration d'un module de gestion de paiements fourni par la société Ingenico spécialisée dans les solutions de paiements en ligne. Ce module permet le traitement des paiements relatifs aux réservations via une page de paiement lorsque les clients y sont redirigés.

Pour le projet « StarFish Hotels », un compte sera créé dans l'environnement de test d'Ingenico afin de permettre le traitement et le suivi des paiements exécutés. Ce compte permet aussi de configurer les différents modes de paiements qui seront autorisé ainsi les liens de redirection en cas de succès ou d'échec du paiement.

Le processus de gestion des paiements s'exécute après la création de la réservation et se présente sous forme de page web contenant un résumé de la transaction à exécuter ainsi que les différents modes de paiements autorisés. Une fois que l'utilisateur sélectionne le mode souhaité, un formulaire, où l'utilisateur peut renseigner les détails de paiements, sera généré.

Au niveau de l'application mobile, le traitement sera effectué sur base des échanges serveur-à-serveur dès que l'utilisateur introduit les informations relatives au moyen de paiement.

## 6. Gestion des notifications

Cette fonctionnalité sera implémentée pour communiquer avec les appareils qui embarquent l'application mobile de la société. Le principe consiste à diffuser des messages publicitaires ainsi que des informations visant à ramener l'utilisateur vers l'application. Une plateforme sera paramétrée pour permettre l'envoi des notifications. Elle permet aussi de remonter les éventuels bugs ou erreurs générés par l'application mobile. Cette plateforme fournit aussi un outil performant qui permet d'analyser l'utilisation de l'application mobile.

Il est aussi envisageable d'implémenter l'envoi des notifications au niveau de l'application web. Le but est d'envoyer des notifications sans devoir s'authentifier sur la plateforme de gestion de l'application mobile.

## Section III : Architecture de l'application

Après l'analyse des différentes fonctionnalités retenues, il est temps de les structurer sous forme d'arborescence de pages ou de vues. L'avantage de ce type de structure est que cela permet de visualiser les liens entre les pages qui traitent les fonctionnalités.

### 1. Application web

Dans le schéma 1, on trouvera la disposition des différentes pages qui composent l'application web ainsi que les liens entre ces pages. On constate aussi que la navigation est mise du haut vers le bas puisqu'on suit l'arborescence du site web.

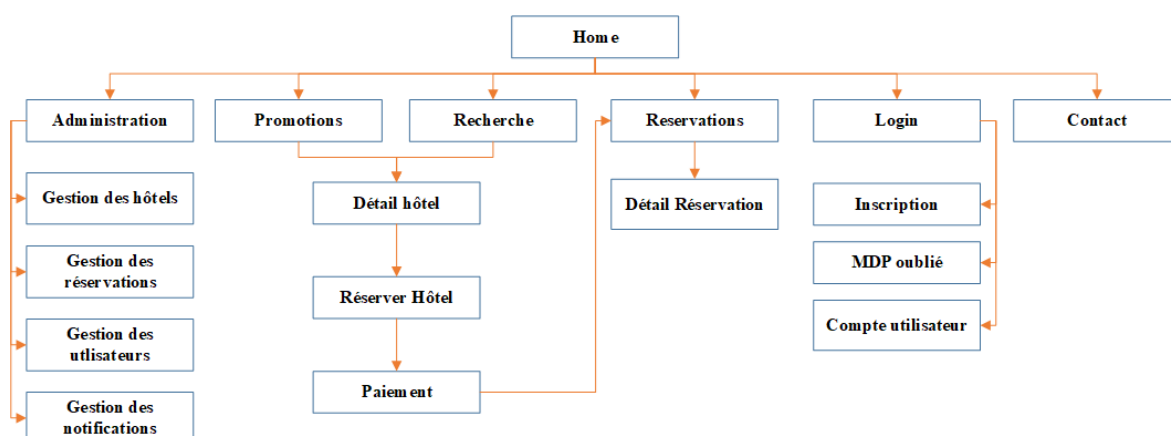


Schéma 1 : Architecture de l'application web

### 2. Application mobile

Dans le schémas 2, on trouve les différentes vues qui seront disponible au niveau de l'application mobile ainsi que les liens entre les différentes pages. La hiérarchie est mise du bas vers le haut, contrairement au schéma précédent, car les pages seront affichées selon un flux de navigation sous forme de pile (Stack).

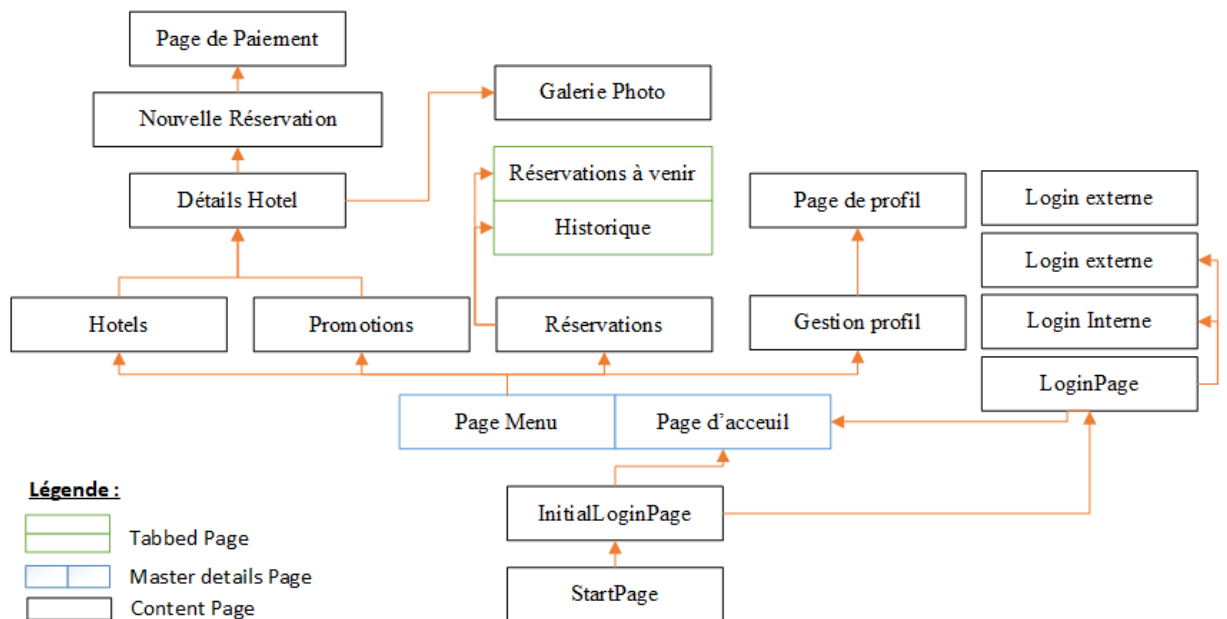


Schéma 2 : Architecture de l'application mobile

## Section IV : Infrastructure technique

Le projet sera construit autour de l'infrastructure illustrée dans le schéma 3. On constate la présence d'une base de données SQL server, cette dernière sera liée à une application Asp.Net web api qui pourra gérer les volets web MVC traditionnels avec les fonctionnalités intégrées d'une interface de programmation applicative (api). Les échanges client-serveur sont représentés par un navigateur web pour l'application web MVC et une application mobile pour l'api.

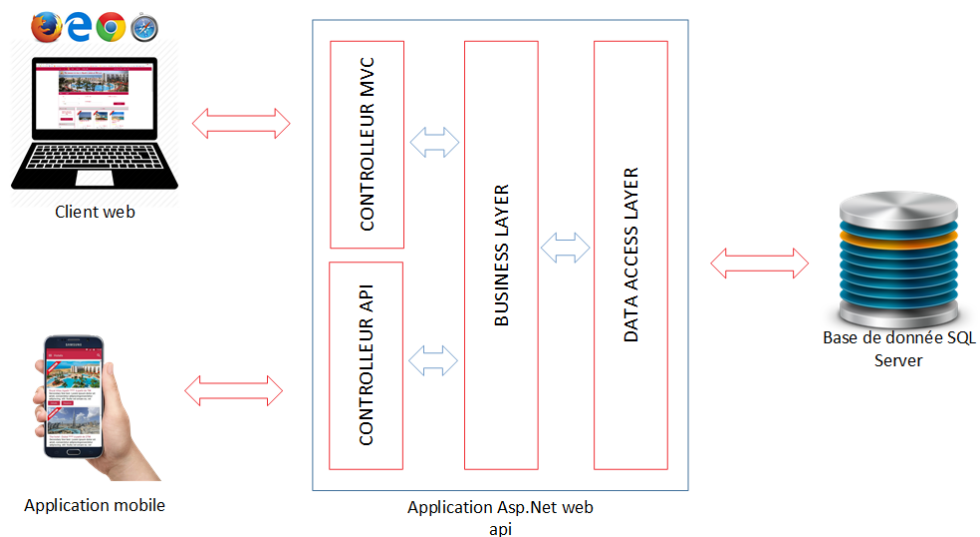


Schéma 3 : Infrastructure technique

# Synthèse

---

Nous avons vu dans cette partie comment nous sommes parvenus à déterminer les besoins du client en matière de fonctionnalités.

Après validation de cette première partie du projet, je vais démontrer le cheminement des prochaines parties de ce travail. Je vais passer, dans un premier temps, par l'analyse des fonctionnalités retenues avant de continuer vers le volet qui porte sur la réalisation effective des deux applications en matière de développement et de conception de base de données.

## Seconde Partie : Méthodologie et analyse

---



# Introduction

---

Poursuivant cette première phase, j'entame cette seconde partie qui consiste à cerner les différents besoins du client à travers une analyse fonctionnelle du projet. Cette analyse permet d'avoir une idée sur la méthodologie qui sera employée ainsi que les aspects techniques à respecter lors de l'exécution des fonctionnalités.

## Chapitre I : Méthodologie

---

Pour l'élaboration de cette partie, deux approches seront utilisées pour exposer l'ensemble des fonctionnalités. La méthode utilisée dans l'analyse est de classer les cas d'utilisation selon leur importance et leurs dépendances.

À titre d'exemple, la fonctionnalité « Créer une réservation » est une fonctionnalité très importante, car c'est la fonctionnalité majeure du projet. L'analyse de celle-ci impliquera l'analyse de la fonctionnalité d'authentification vu l'interdépendance existante entre les deux fonctionnalités.

L'analyse des cas d'utilisation sera réalisée pour la plupart par des diagrammes UML<sup>1</sup> sauf pour les diagrammes d'activité qui seront réalisés selon une approche de modélisation BPMN<sup>2</sup> étant donné que les processus BPMN sont, pour moi, plus parlants que les diagrammes d'activité.

## Chapitre II : Analyse

---

Dans ce chapitre, je vais traiter des fonctionnalités que j'estime être les plus importantes. Ces fonctionnalités seront sélectionnées sur base de leur importance ainsi que de leur pertinence.

Je commencerai l'analyse en présentant l'ensemble des cas d'utilisation pour chacune des applications afin de visualiser les différents acteurs qui agissent sur chaque fonctionnalité. Ensuite, nous passerons aux différents diagrammes qui dévoilent le traitement de chaque fonctionnalité.

Enfin, nous aboutirons à la fin de ce chapitre sur une analyse de la structure de données qui recevra et stockera les informations générées par l'ensemble des applications.

---

<sup>1</sup> UML : Le langage de modélisation unifié (Unified Modeling Language) est la modélisation normalisée permettant de visualiser la conception d'un système sous forme de pictogrammes.

<sup>2</sup> BPMN : Le Business Process Model and Notation est une modélisation des processus métier planifiés afin de permettre une vue détaillée sur l'exécution de ce dernier.

## Section I : Diagramme des cas d'utilisation

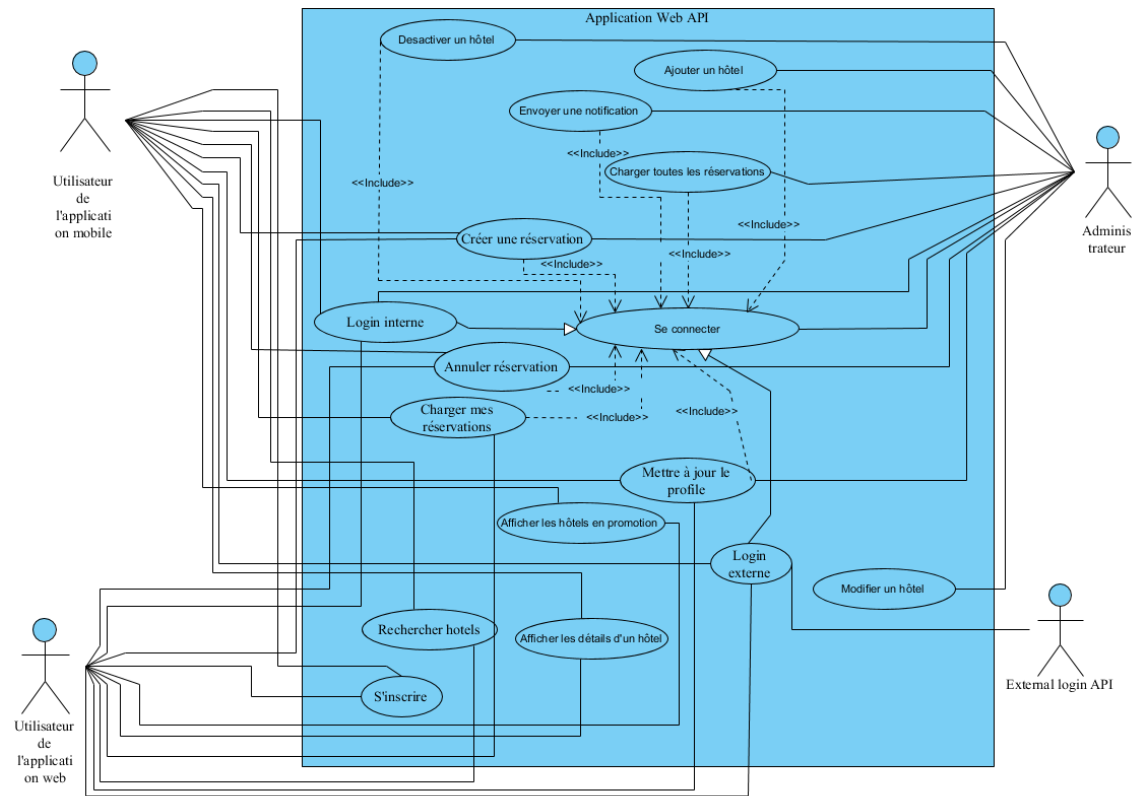


Schéma 4 : Diagramme des cas d'utilisation de l'application web

Dans la schéma 4 ci-dessus, nous allons voir les différentes fonctionnalités prévues pour le projet ainsi que les différents acteurs qui sont liés à chacune des fonctionnalités.

L'utilisateur accèdera à une variété de fonctionnalités tel que :

- Chercher des hôtels
- Afficher le détail d'un hôtel
- Créer une réservation
- Mettre à jour son profil
- S'inscrire et se connecter
- Afficher ses réservations et pouvoir les annuler

Certaines des fonctionnalités nécessitent l'authentification des utilisateurs, celle-ci sera implémentée pour assurer une certaine sécurité et permettre de restreindre l'accès aux utilisateurs uniquement aux différentes informations qui les concernent.

L'administrateur à son tour peut accéder, en plus des fonctionnalités autorisées à l'utilisateur normal, à d'autres fonctionnalités relatives à la gestion des hôtels, des réservations et des utilisateurs.

La fonctionnalité de connexion via le réseau social « Authentification externe » nécessite l'interpellation d'un acteur externe à nos applications. On parle ici de l'interface de

programmation applicative des différents réseaux sociaux qui autorisera l'obtention des informations personnelles des utilisateurs.

## Section II : Règles métier

Dans cette partie j'aborderai certaines des règles métier qu'il faut prendre en considération :

- Pour créer, modifier, désactiver un hôtel, l'utilisateur doit avoir le rôle « administrateur ».
- Effectuer une réservation nécessite l'authentification de l'utilisateur.
- Les logins externes implémentés sont : Facebook, Google+ et Microsoft.
- L'annulation d'une réservation nécessite l'authentification de l'utilisateur et doit s'effectuer au plus tard 7 jours avant la date d'arrivée du client.
- L'authentification interne nécessite une adresse email comme nom d'utilisateur.
- L'utilisateur dans le rôle « client » ne peut afficher que les réservations qu'il a effectuées et ne peut pas visualiser les réservations d'un autre utilisateur.
- Visualiser l'ensemble des réservations nécessite disposer du rôle « administrateur ».
- Chaque hôtel dispose d'un coefficient par lequel le tarif de chaque catégorie est multiplié pour obtenir le tarif appliqué par l'hôtel.
- Une réservation ne peut pas concerner des chambres appartenant à des hôtels différents.

## Section III : Notion de rôle et Autorisations

Dans l'optique de gérer et sécuriser l'accès à certaines fonctionnalités de ce projet, il est nécessaire de distinguer les différents type d'utilisateurs et de leur assigner les rôles qui réglementent les droits d'accès dont ils auront besoin. Nos applications seront utilisées selon les rôles suivants :

- Anonyme** : Ce rôle est attribué à tous les utilisateurs non authentifiés et autorise l'accès aux différentes fonctionnalités publiques comme :
- Affichage des hôtels
  - Affichage des promotions
  - Recherche d'hôtels
- Client** : Ce rôle est réservé aux utilisateurs de type « client » et qui sont authentifiés. En plus des fonctionnalités publiques, ce rôle donne accès à d'autres fonctionnalités telle que :
- Effectuer une réservation
  - Afficher ses réservations
  - Annuler une réservation (au plus tard 7 jours avant la date d'arrivée)
- Administrateur** : Le rôle administrateur est le plus privilégié car il dispose de l'accès à l'entièreté des fonctionnalités. Les utilisateurs de type « Administrateur » peuvent exécuter, en plus des fonctionnalités susmentionnées, les différentes actions relatives à :
- La gestion des hôtels (création, modification et désactivation)
  - La gestion des réservations (création, modification et annulation sans contrainte de délai)

- La gestion des utilisateurs (création, modification, désactivations et attribution de rôles)

## Section IV : Scénarios des cas d'utilisation

### 1. Authentification

#### 1.1 Inscription d'un nouvel utilisateur (client)

Cette fonctionnalité vise la création de nouveaux comptes utilisateur. L'accès à cette fonctionnalité est possible à travers la page de connexion et dans le cas où l'utilisateur désire créer un nouveau compte interne. Le tableau suivant illustre le déroulement du processus.

Cas d'utilisation	Créer un nouveau compte utilisateur	
Etat	<ul style="list-style-type: none"><li>- La page d'inscription a déjà été générée</li><li>- L'utilisateur a déjà introduit les informations nécessaires</li></ul>	
Scénario principal		
Navigateur / App mobile		Serveur web
Utilisateur		Application web api
1- L'utilisateur lance l'inscription.		2- L'application valide les informations introduites.
4- L'utilisateur est prêt à se connecter avec son compte utilisateur.		3- L'application enregistre le nouvel utilisateur et le redirige vers la page de connexion.

#### 1.2 Authentification interne

Le principe de l'authentification interne est de permettre aux utilisateurs de se connecter grâce à leurs identifiants personnels. Dans le cadre de notre projet les identifiants seront composés d'adresse mail et de mot de passe. L'interaction durant le processus d'authentification se fera entre l'utilisateur et l'application web api.

Cas d'utilisation	Authentification interne	
Etat	<ul style="list-style-type: none"><li>- La page d'authentification a déjà été générée</li><li>- L'utilisateur a déjà introduit ses identifiants</li></ul>	
Scénario principal		
Navigateur / App mobile		Serveur web
Utilisateur		Application web api
1- L'utilisateur lance l'authentification.		2- L'application valide les identifiants et génère un token pour l'utilisateur. 3- L'application renvoie le token vers le client.
4- L'utilisateur est prêt à effectuer des requêtes vers les méthodes nécessitant l'authentification.		

### 1.3 Authentification externe

Dans le cas où il ne dispose pas de compte interne au niveau de l'application, et désire simplement utiliser l'authentification externe, l'utilisateur peut se connecter avec son compte d'un des réseaux sociaux implémentés dans le cadre de ce projet. Le processus se déroule comme suit :

Cas d'utilisation	Authentification externe		
Etat	- La page d'authentification a déjà été générée		
Scénario principal			
Navigateur / App mobile	Serveur web	Serveur externe	
Utilisateur	Application web api	Api du réseau social	
1- L'utilisateur lance l'authentification externe.	2- L'application redirige vers la page de connexion et d'autorisation.		
3- L'utilisateur introduit ses identifiants puis autorise ou refuse le chargement de son profil de base.		4- Vérification et génération d'un code d'autorisation.	
		5- Renvoi du code d'autorisation.	
7- L'utilisateur est prêt d'accéder au action nécessitant l'authentification.	6- Vérification si ce compte utilisateur a déjà été créé; si ce n'est pas le cas, l'application charge le profil de base de l'utilisateur et crée un compte utilisateur.		

## 2. Créer une réservation

Cette fonctionnalité s'exécute suite à la demande de l'utilisateur et se déroule en trois phases :

<b>Chargement des disponibilités et des prix</b>	Cette opération consiste à vérifier et charger les quantités de chambres disponible pour chaque catégorie. Sur base des dates introduites, le tarif de chaque catégorie/nuit est aussi récupéré.
<b>Création de la réservation</b>	Ceci concerne la création de la réservation, l'affectation des différentes chambres qui lui sont attribuées et la vérification qui vise à s'assurer que les chambre ne sont pas affectées à une autre réservation pour la même période. Cette étape est gérée sous forme de transaction, ce qui veut dire qu'elle est validée si toutes ses étapes ont connu le succès ou annulées en cas d'échec.
<b>Gestion du paiement</b>	Cette étape ne s'exécute qu'en cas du succès de celle qui la précède. Cette opération porte sur la réalisation du paiement relatif à la réservation créée.

Cas d'utilisation	Créer une réservation	
Etat	<ul style="list-style-type: none"><li>- L'utilisateur s'est authentifié</li><li>- L'utilisateur a déjà choisi l'hôtel qu'il désire réserver</li></ul>	
Scénario principal		
Navigateur / App mobile	Serveur web	
Utilisateur	Application web api	
1- L'utilisateur lance le processus de réservation.	2- L'application vérifie et charge les quantités de chambres disponibles, le prix pour chaque catégorie/nuit et les formules de restauration possibles pour l'hôtel choisi.	
4- L'utilisateur choisit les quantités des chambres à réserver, les dates de séjour, spécifie le nombre d'adultes et enfants et sélectionne la formule désirée pour chaque chambre.	3- L'application renvoie les résultats.	
5- L'utilisateur valide la réservation.	6- L'application crée la réservation validée par l'utilisateur.	
	7- Redirection de l'utilisateur vers la page de paiement en ligne.	
8- L'utilisateur introduit les informations relatives au moyen de paiement choisi.	9- L'application exécute le paiement et confirme la réservation en cas de	

	réussite de l'opération ou l'annule en cas d'échec. 10- L'application redirige l'utilisateur vers la page qui reprend l'ensemble de ses réservations.
--	--

### 3. Rechercher des hôtels

Sur base des différents critères introduits au niveau du formulaire de recherche, cette fonctionnalité se charge de récupérer tous les hôtels qui répondent à ces critères. L'exécution de la fonctionnalité commence par déterminer la zone géographique concernée par la recherche ainsi que la période de séjour choisie. Sur base de ces éléments, l'application va chercher les hôtels qui auront des chambres disponibles pendant ces dates et se trouvant dans la zone (ville / pays) sélectionnée. Les hôtels chargés subiront un filtrage qui porte sur l'analyse de la capacité d'accueil de chacun d'eux. Seuls les hôtels qui disposent d'assez de lits seront gardés et affichés à l'utilisateur.

Cas d'utilisation	Rechercher des hôtels
Etat	- L'utilisateur a déjà remplis le formulaire de recherche.
Scénario principal	
Navigateur / App mobile	Serveur web
Utilisateur	Application web api
1- L'utilisateur lance le processus de recherche.	2- L'application charge les hôtels disponibles pour les dates de séjour choisies et qui se trouvent dans le pays et/ou la ville choisie.
	3- L'application filtre les hôtels obtenus et garde seulement ceux qui disposent d'une capacité d'accueil suffisante pour le nombre de personnes introduit.
	4- L'application renvoiE les résultats.
5- L'utilisateur dispose des hôtels qui répondent au critères de recherche qu'il a introduits.	

### 4. Afficher mes réservations

L'exécution de cette fonctionnalité dépend de l'authentification de l'utilisateur. S'il n'est pas connecté, ce dernier sera invité à introduire ses identifiants pour pouvoir récupérer l'ensemble des réservations effectuées.

Dans l'application web, l'affichage des réservations de l'utilisateur connecté sera présentée sous forme de liste. Chaque ligne représente les informations importantes d'une réservation. L'affichage du détail complet de chaque réservation est possible à travers un bouton « détails » qui sera implémenté à cette fin.

Au niveau de l'application mobile, et une fois que l'ensemble des réservations sera chargé, l'affichage des réservations sera réalisé à travers deux onglets. Le premier sera consacré à l'historique des réservations, celui-ci reprend toutes les réservations dont la date de début de séjour est antérieure à la date du jour. Le deuxième affichera les réservations futures, cet onglet reprend les réservations dont la date de début de séjour est postérieure à la date du jour.

Cas d'utilisation	Afficher mes réservations
Etat	- L'utilisateur s'est connecté avec son compte.
Scénario principal	
Navigateur / App mobile	Serveur web
Utilisateur	Application web api
1- L'utilisateur lance le processus de recherche.	2- L'application récupère l'identifiant de l'utilisateur connecté. 3- L'application charge l'ensemble des réservations de cet utilisateur selon son identifiant récupéré. 4- L'application renvoie les résultats.
5- L'utilisateur dispose de l'ensemble des réservations qu'il a effectué.	

## Section V : Diagrammes de séquences

### 1. Authentification

#### *1.1 Authentification interne*

Ce schéma nous illustre les étapes majeures que parcourt le processus d'authentification interne gérée dans « Asp.net ». Celui-ci se base sur une validation des identifiants fournis, et la génération d'un token, au cas où la validation est réussie. Ce token représente une autorisation d'accès aux actions qui nécessitent que l'utilisateur soit connecté.



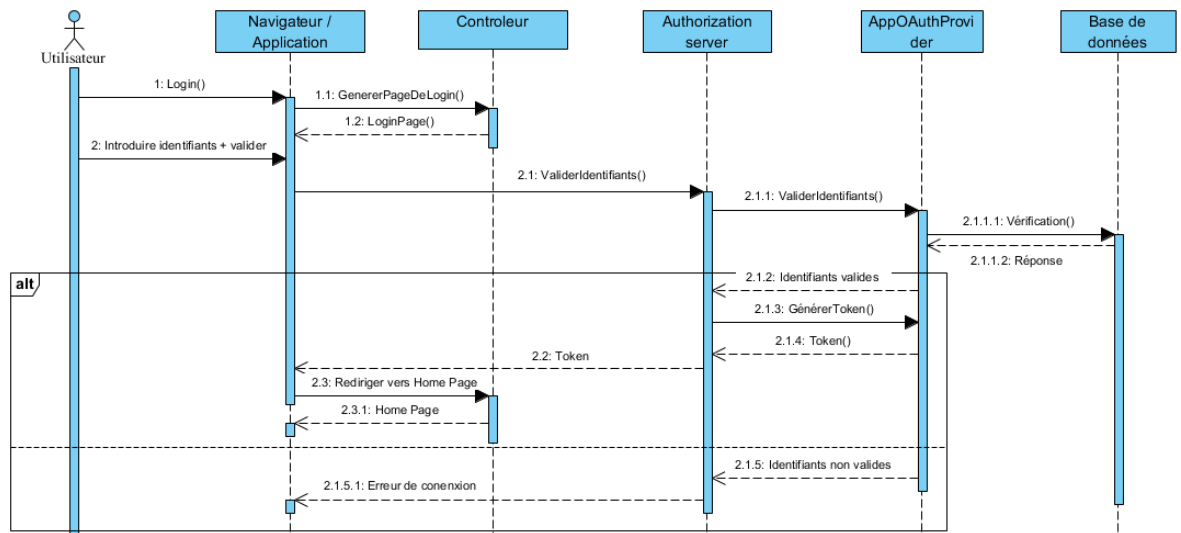


Schéma 5 : Diagramme de séquence de l'authentification interne "Asp.net"

## 1.2 Authentification externe

L'authentification externe repose sur l'obtention des informations relatives au profil de base de l'utilisateur auprès du serveur d'autorisation du fournisseur de login externe. Cette opération se fait à travers la redirection de l'utilisateur vers ce serveur. Une fois redirigé, l'utilisateur introduit ses identifiants (nom d'utilisateur /mot de passe) et accepte ou refuse l'autorisation demandée et qui concerne son profil de base. Si l'opération est validée par l'utilisateur, un code d'accès est généré par le serveur d'autorisation, ce code va servir, de la même façon que le token pour l'authentification interne, à récupérer les informations de l'utilisateur tel que le nom, prénom, image de profil et autre.

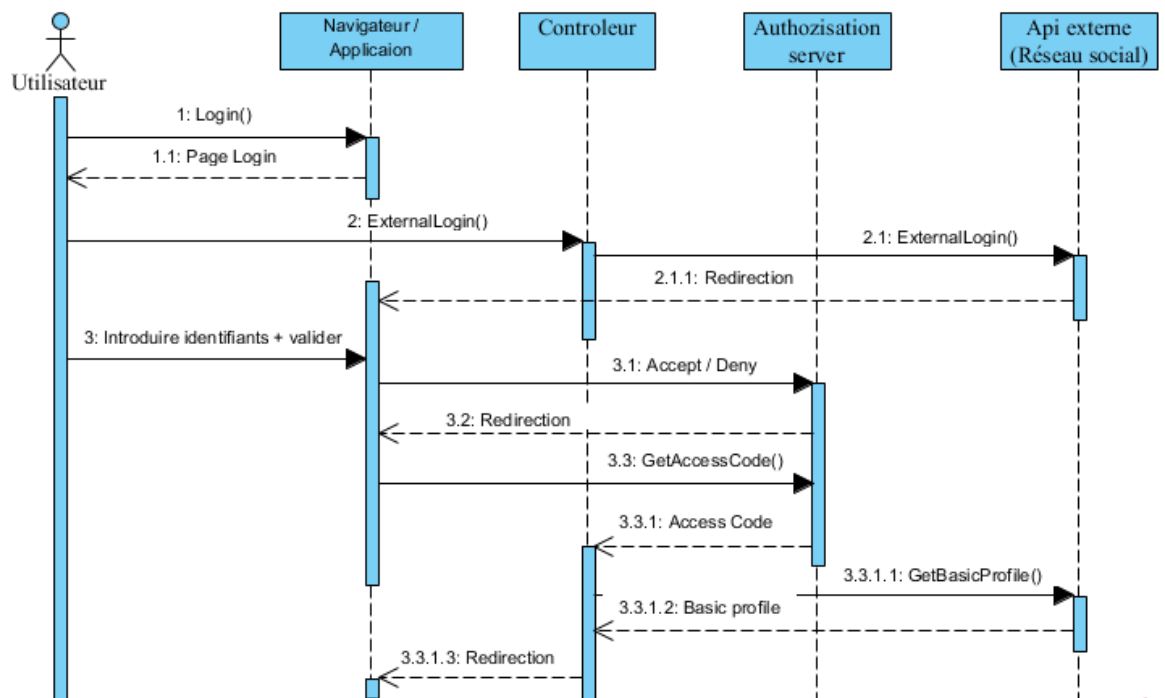


Schéma 6 : Diagramme de séquence de l'authentification externe "Réseau sociaux"

## 2. Créer une réservation

Le processus de la création d'une réservation passe par les étapes suivantes que je vais aborder comme suit :

### 2.1 Récupération des disponibilités

Une fois que l'utilisateur choisit de réserver un hôtel, le processus démarre avec la récupération des informations utiles à la génération du formulaire de réservation. Cette opération consiste à charger les informations suivantes :

- Quantité de chambres disponibles pour chaque catégorie
- Image de chambre pour chaque catégorie
- Prix pour chaque catégorie, le prix affiché est calculé pour la première nuit. Ce prix affiché sera calculé sur base du tarif de la catégorie et du coefficient appliqué par l'hôtel ; **tarif appliqué = tarif catégorie x coefficient hôtel.**
- Les formules de restauration possibles dans l'hôtel sélectionné.

Une fois que les disponibilités sont chargées, la vue avec le formulaire de réservation est générée et l'utilisateur peut choisir la quantité de chambres qu'il souhaite réserver selon les limites des chambres disponibles.

### 2.2 Calcul des prix

Pour chaque chambre sélectionnée, et selon les changements des formules, une requête de calcul de prix est effectuée. Cette vérification se fait sur base de la date d'arrivée, date de départ et la formule choisie.

Une boucle récupère en premier temps le tarif pour chaque nuit entre la date d'arrivée et la date de départ. Ce montant sera additionné par la suite au total de la formule multipliée par le nombre de nuit, ce qui nous donne le montant total pour la chambre. Pour chaque chambre sélectionnée, l'affichage du détail de tarif est possible et permet de voir les variations des tarifs/nuit pendant le séjour.

Suite aux changements au niveau des quantités de chambres ou de formules, le montant total de la réservation est calculé et mis à jour.

### 2.3 Enregistrement de la réservation

Après avoir effectué son choix de chambres, l'utilisateur valide son choix et le processus de l'enregistrement de la réservation démarre sous forme de transaction. Celui-ci commence par la création de la réservation sur base des éléments suivants :

- Date de réservation qui est la date du jour
- Nom du client bénéficiaire de la réservation
- Email de réservation
- Adresse de facturation
- Montant de la réservation

Suite à la création de la réservation, l'identifiant de la réservation insérée est récupéré. Ensuite, une boucle démarre pour affecter les différentes chambres à la réservation qu'on vient d'enregistrer. L'affectation se fait, pour chaque chambre, sur les éléments suivants :

- Nombre d'adultes
- Nombre d'enfants

- Date d'arrivée
- Date de départ
- Nom complet du client qui va occuper la chambre
- Arrangement choisi pour cette chambre
- L'information relative à la nécessité d'un lit supplémentaire
- Observations
- Montant total de la chambre

Après l'affectation des chambres, le système vérifie que la réservation de chambres affectées ne chevauche pas d'autres réservations existantes ou nouvellement créées pour les mêmes chambres. Si cette vérification réussit, la réservation est enregistrée sinon toutes les opérations effectuées dans cette étape seront annulées.

## ***2.4 Gestion du paiement***

Ensuite vient le processus de gestion de paiement où l'utilisateur est redirigé vers la page de paiement en ligne où il peut introduire les informations relatives à son moyen de paiement.

### **2.4.1 Application web**

Pour effectuer cette opération au niveau de l'application web, l'application a besoin d'instancier un formulaire avec les composants suivants :

- PSPID : ce champ concerne le marchand pour qui l'opération sera effectuée.
- ORDERID : concerne le numéro de la transaction, ceci peut être l'id de la réservation récemment créée.
- AMOUNT : concerne le montant total de la transaction.
- CURRENCY : devise dans laquelle la transaction sera effectuée.
- CN : fait référence au nom du client (Customer Name)

Une fois renseigné, le formulaire est transmis et une redirection est effectuée vers la page du paiement. Cette page présente les différents moyens de paiement possibles, le client peut choisir le moyen qui lui convient et introduire les informations de son mode de paiement.

Si le paiement s'exécute avec succès, la réservation passe le stade d'enregistrement et s'attribue le statut « Confirmée », sinon il obtient le statut « Annulée ».

### **2.4.2 Application mobile**

Au niveau de l'application mobile, la solution désignée pour le moment consiste à envoyer un mail à l'utilisateur avec le lien vers la page du paiement. Le client peut suivre le lien pour effectuer son paiement.

Le diagramme suivant démontre les différentes étapes du processus de réservation.

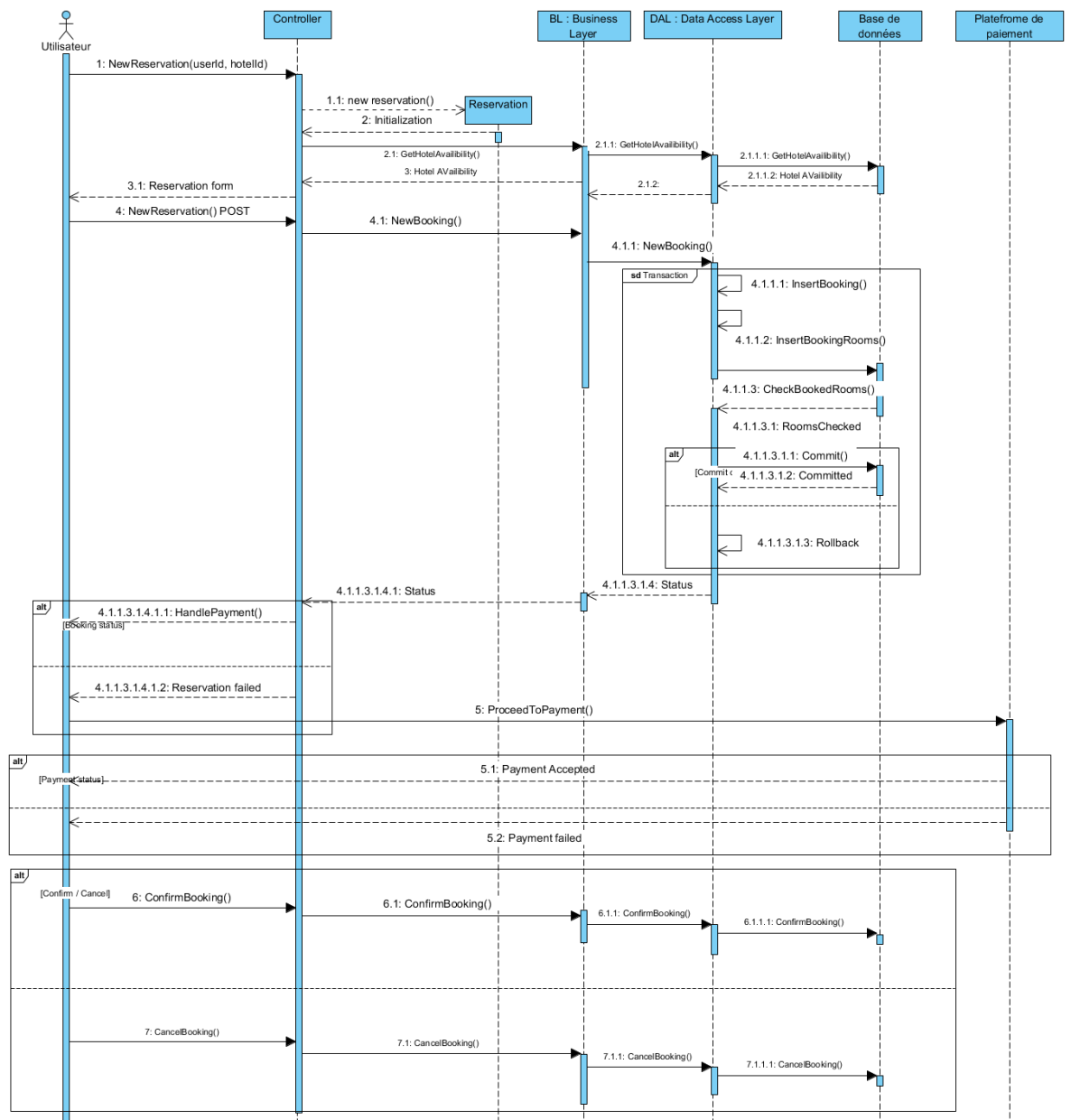


Schéma 7 : Diagramme de séquence - Créer une réservation via l'application web

### 3. Recherche des hôtels

Le diagramme suivant nous montre le déroulement de la recherche des hôtels demandés par l'utilisateur. On constatera qu'une fois les hôtels chargés depuis la base données, ils seront filtrés afin de garder ceux qui répondent au besoin du client en matière de capacité d'accueil.

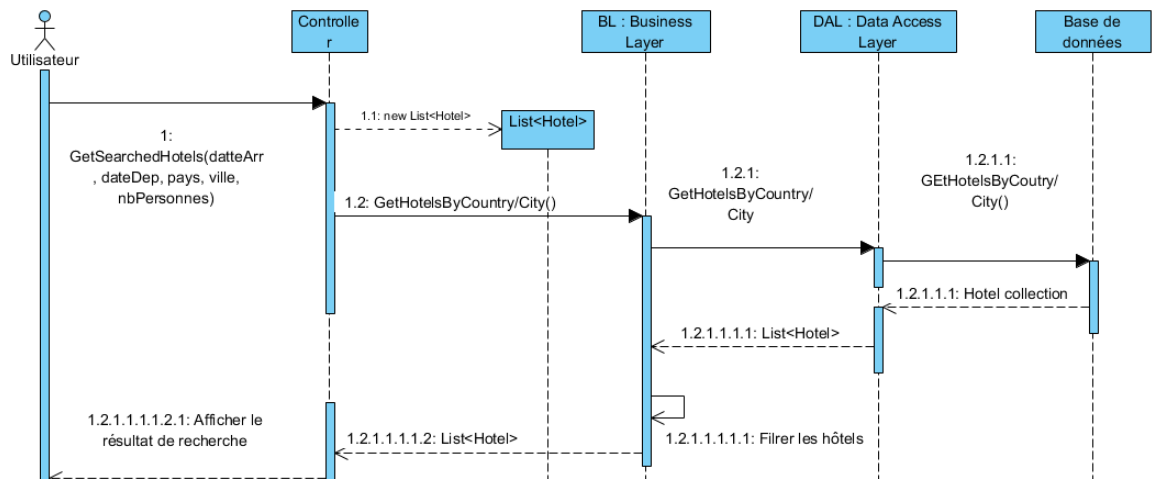


Schéma 8 : Diagramme de séquence - Recherche des hôtels via l'application web

## 4. Afficher mes Réservations

Le diagramme ci-dessous illustre les différents échanges entre l'application mobile et l'interface de programmation applicative dans le but d'afficher l'ensemble des réservations de l'utilisateur connecté.

Nous constatons aussi qu'avant le chargement des réservations, l'identifiant unique de l'utilisateur est récupéré par l'application. Ceci est implémenté de cette façon pour garantir de charger les réservations de l'utilisateur connecté ainsi que d'éviter qu'un utilisateur tente de récupérer les réservations d'un autre utilisateur.

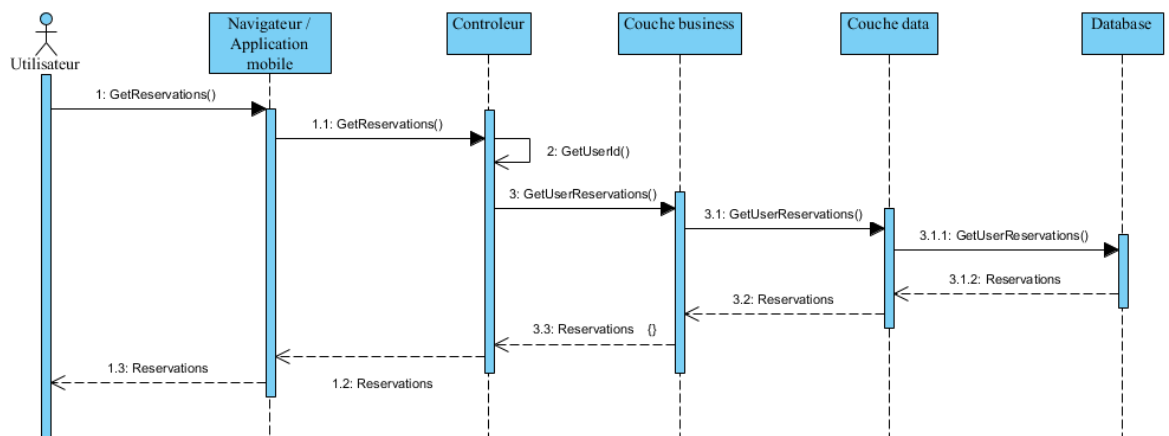


Schéma 9 : Diagramme de séquence - Afficher mes réservation via l'application mobile

## Section VI : Analyse des processus

Dans cette section, nous verrons l'exécution des différentes fonctionnalités selon une approche différente de celle pratiquée dans les différentes sections de ce présent chapitre. Cette approche s'appuie sur une méthodologie orientée processus. Ce qui veut dire que nous allons simuler l'exécution de chaque fonctionnalité sur base d'un flux séquentiel avec des points de contrôle. Certains processus seront découpés en sous-processus quand on constate qu'ils deviennent trop chargés ou qu'ils présentent plusieurs croisements au niveau des flux. Les sous-processus représentent aussi des petits processus réutilisables ce qui nous évitera de modéliser le même sous-processus plusieurs fois alors qu'on peut simplement déclencher le sous-processus existant.

## 1. Créer une réservation

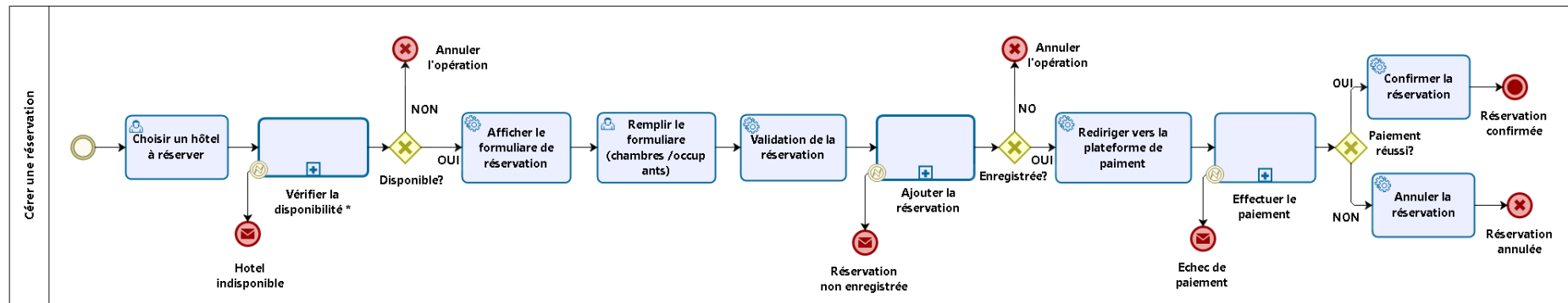


Schéma 10 : Processus - Créer une réservation

Ce processus, illustré par le schéma 10 ci-dessus, nous reflète la création d'une réservation au niveau de l'application web. Ce dernier est composé d'une série de tâches et de sous-processus marqués par un « + ». Chaque sous-processus est susceptible de soulever une erreur en cas d'échec de l'opération qui y est traitée. Nous constatons aussi la présence de quelques points de contrôle qui veillent sur la bonne exécution du processus global et d'en annuler l'exécution en cas d'erreur.

Dans les prochaines pages, nous allons voir les différents sous-processus qui rentrent dans la composition de ce processus de création de réservation. J'établirai un résumé pour chacun des sous-processus. La décomposition de la grande partie de cette fonctionnalité en sous-processus me semble logique, surtout en ce qui concerne la réutilisation des sous-processus par d'autres fonctionnalités ainsi que par souci de clarté dans la modélisation.

## 1.1 Vérifier les disponibilité des chambres

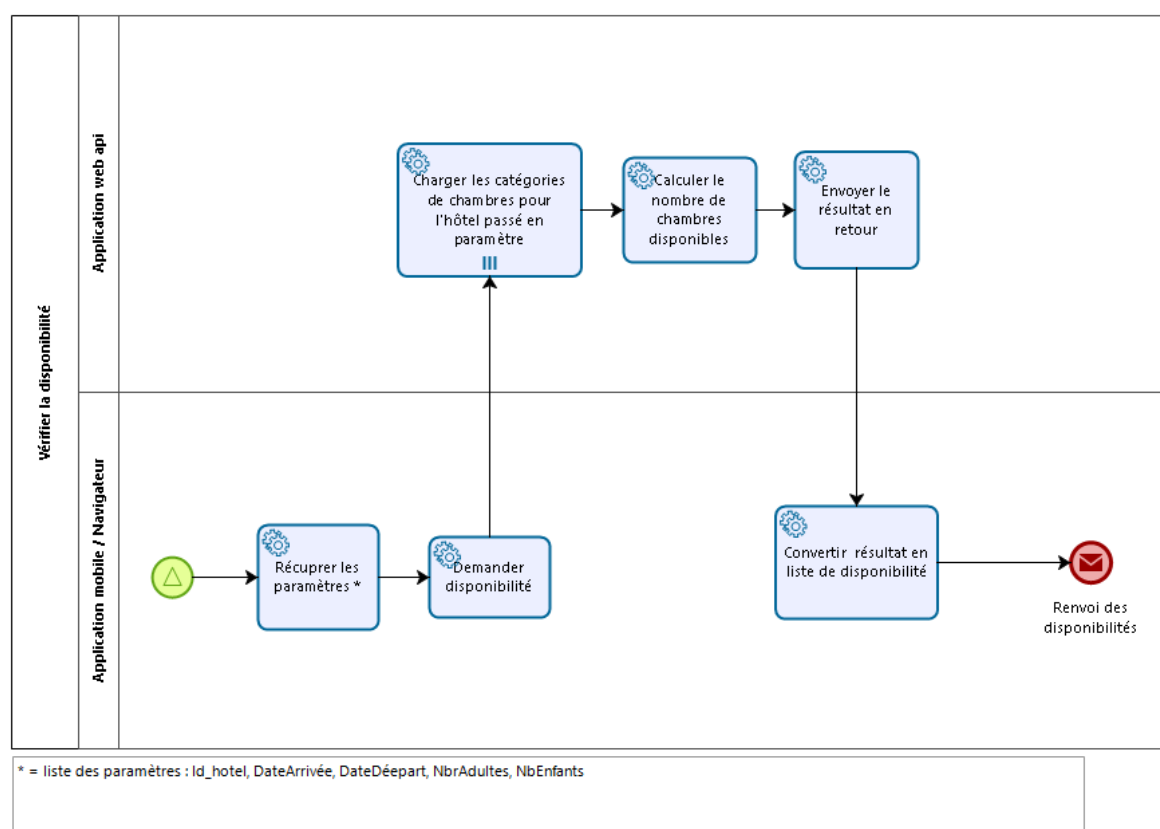


Schéma 11 : Sous-processus - Vérifier les disponibilité d'un hôtel

Ce sous-processus consiste à charger les disponibilité pour chaque catégorie de chambre de l'hôtel faisant l'objet de la réservation. Il commence par charger les catégorie de chambres que l'hôtel offre ainsi que leur prix minimum et calcule pour chaque catégorie le nombre de chambres disponibles sur base des dates de séjours introduites.

Le formulaire de réservation qui sera affiché par la suite dépend du résultat de ce sous-processus.

## 1.2 Ajouter une réservation

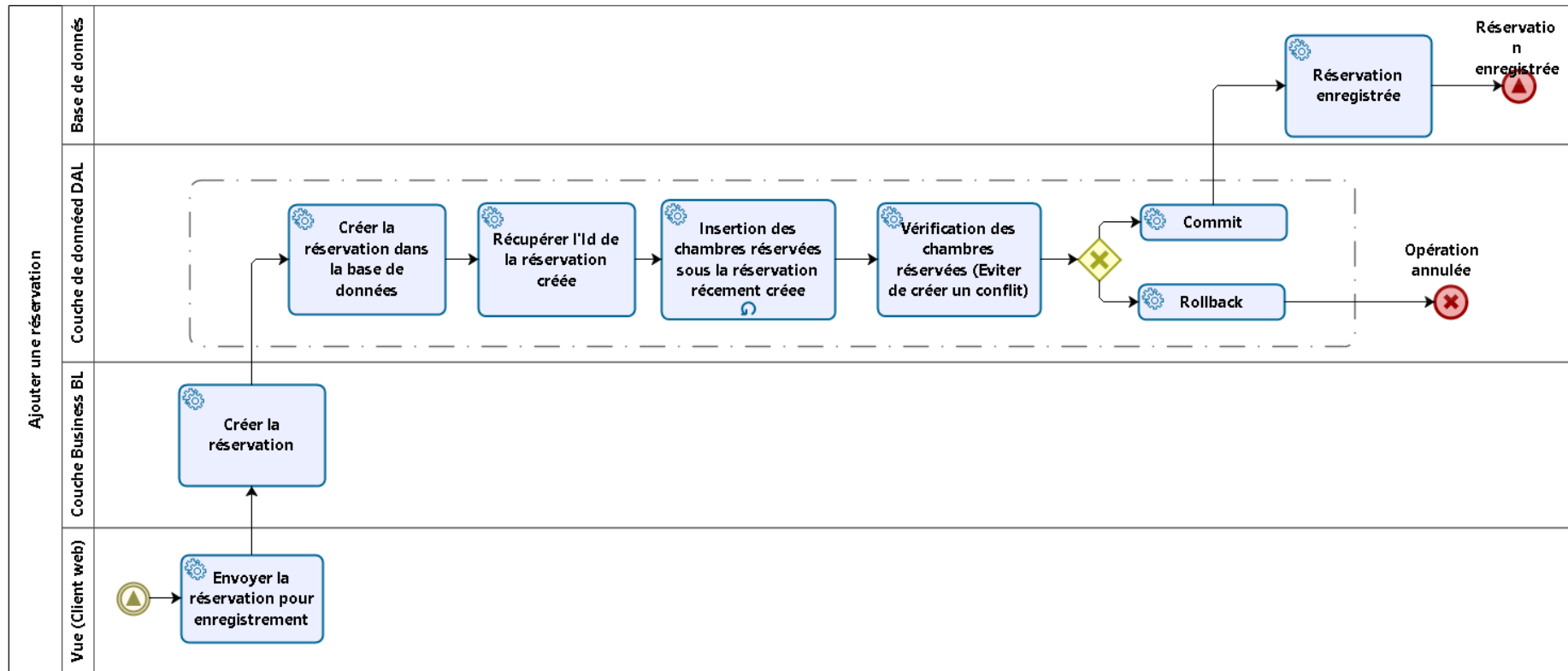


Schéma 12 : Sous-processus - Ajouter une réservation

Une fois que l'utilisateur valide son formulaire de réservation avec les différentes catégories de chambres désirées, il est temps d'enregistrer la réservation. Cette opération se fait sous forme de transaction. J'ai choisi de traiter cet enregistrement de cette sorte pour éviter tout type de conflit en matière de réservation. La transaction nous évite de chevaucher deux ou plusieurs réservations pour la même chambre. Prenons le cas d'un hôtel qui dispose d'une seule chambre libre et deux clients qui essaient de la réserver en même temps. Ce sera le premier à valider la réservation qui l'emporte.



### 1.3 Effectuer un paiement

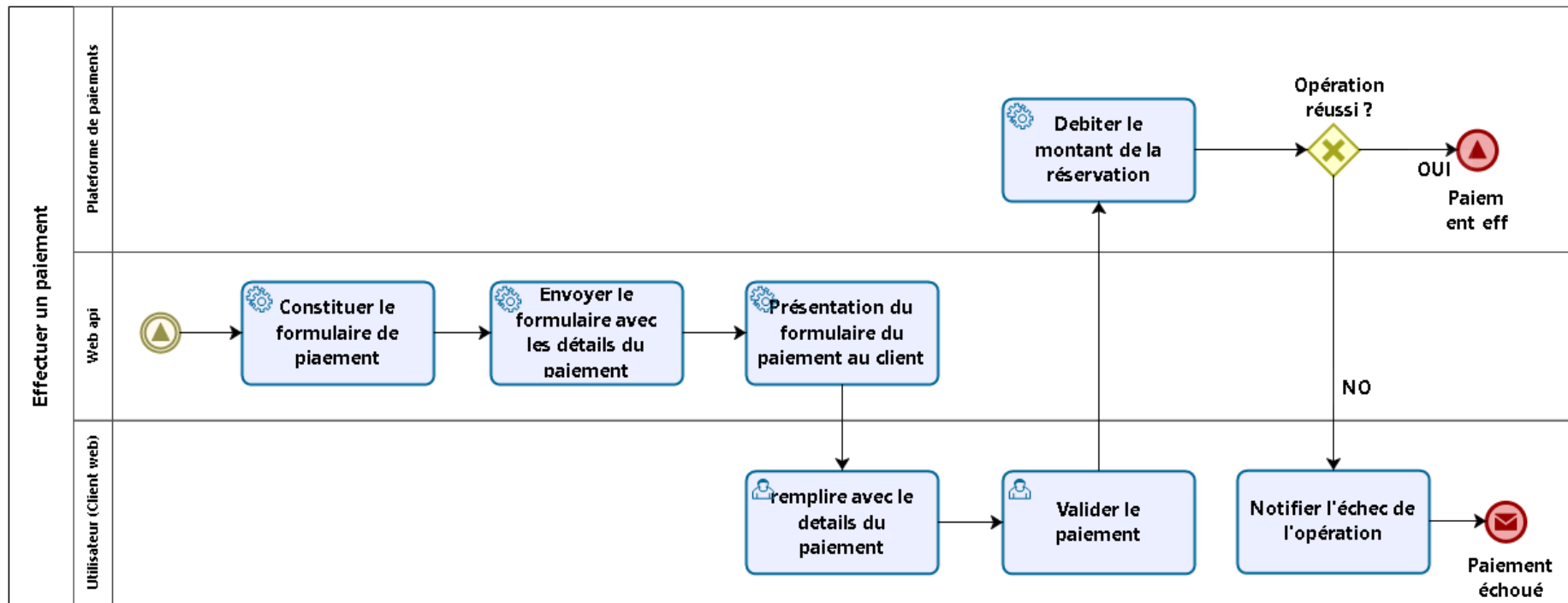


Schéma 13 : Sous-processus - Effectuer un paiement

Après l'enregistrement d'une réservation, il est temps de réaliser le paiement relatif à celle-ci, ceci consiste à préparer un ensemble d'informations qui sera utile pour le traitement des paiements tels que le montant et la référence de la réservation, le compte qui recevra le paiement. Une fois les informations rassemblées, le client sera redirigé vers la plateforme de paiement en ligne. Sur cette dernière, le client sera invité à introduire les informations relatives au mode de paiement qui lui convient. La réussite de l'opération confirme la réservation alors que son échec l'annule.

## Section VII : Structure des données

J'exposerai dans cette section la structure que devrait avoir notre base de données. La réflexion autour de cet élément qui, pour moi, est le pilier porteur du projet qui nécessite :

- La rigueur en matière de conception
- L'anticipation par rapport aux évolutions possibles
- Une grande flexibilité afin de garantir l'évolutivité

Notre schéma entité-association prend en compte les trois critères fondamentaux nécessaires à sa conception ;

- Entités : je regroupe sous ce terme toutes les classes d'objets qui seront mise en relation directe ou indirecte.
- Associations : Les liaisons entre nos entités assurent le lien entre celles-ci. L'utilisation de termes clairs facilite aussi l'interprétation du schéma.
- Cardinalités : un élément non négligeable car, avec son intervalle minimal et maximal, il peut complètement changer la structure d'une base de données à travers la création de nouvelles tables ou simplement par la migration des clés primaires vers les tables directement liées via les associations.

Après la conception de notre schéma entité association, je procèderai à sa conversion en schéma relationnel, ce schéma me permettra d'anticiper et de voir le comportement de certaines associations et leur impact sur la structure générale de la base de données.

Je vais essayer, à travers les deux illustrations suivantes, d'interpréter chacun des deux schémas afin de parcourir certaines particularités.

## 1. Schéma entité-association

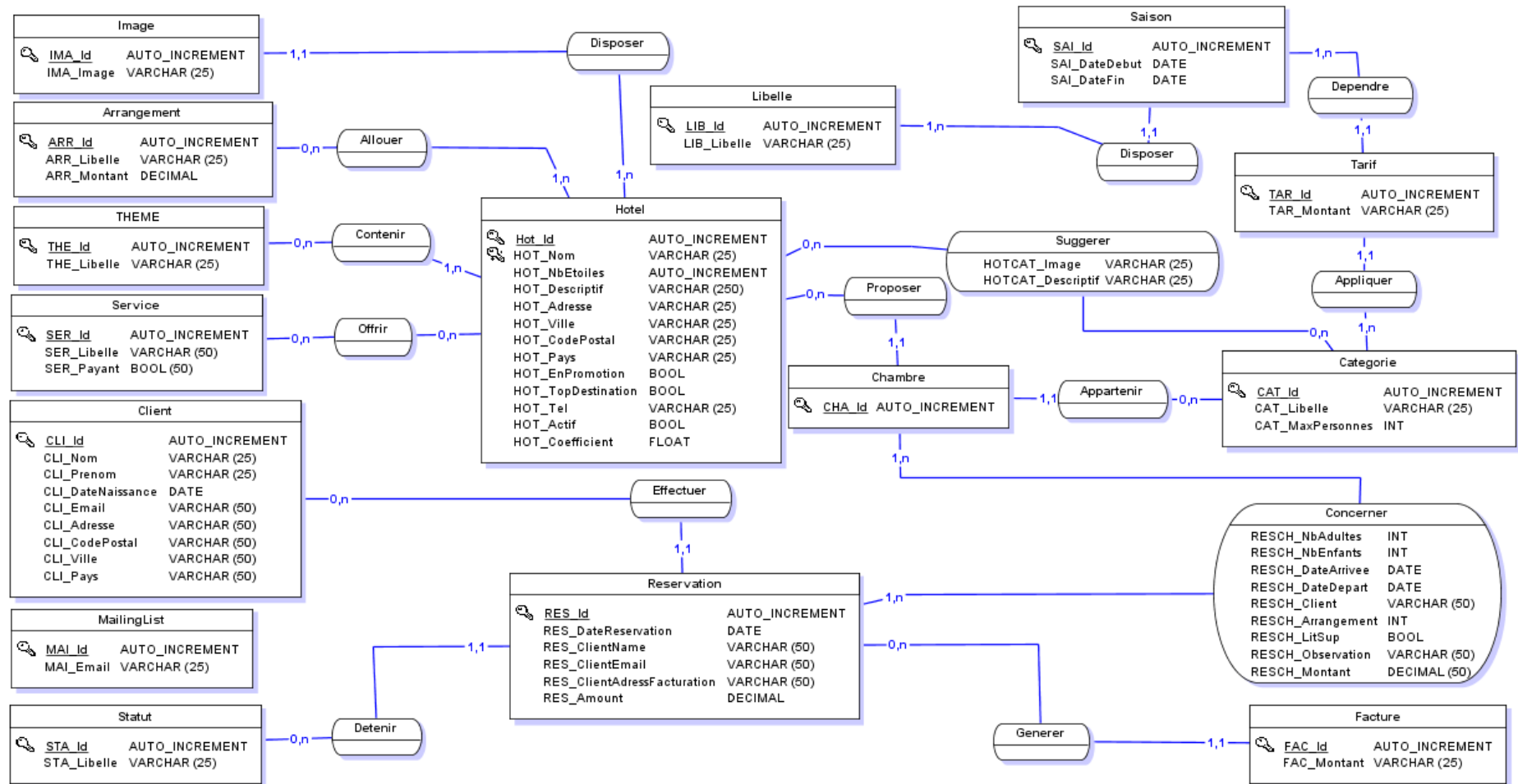


Schéma 14 : Schémas entité-association de la base de données

## 2. Schéma relationnel

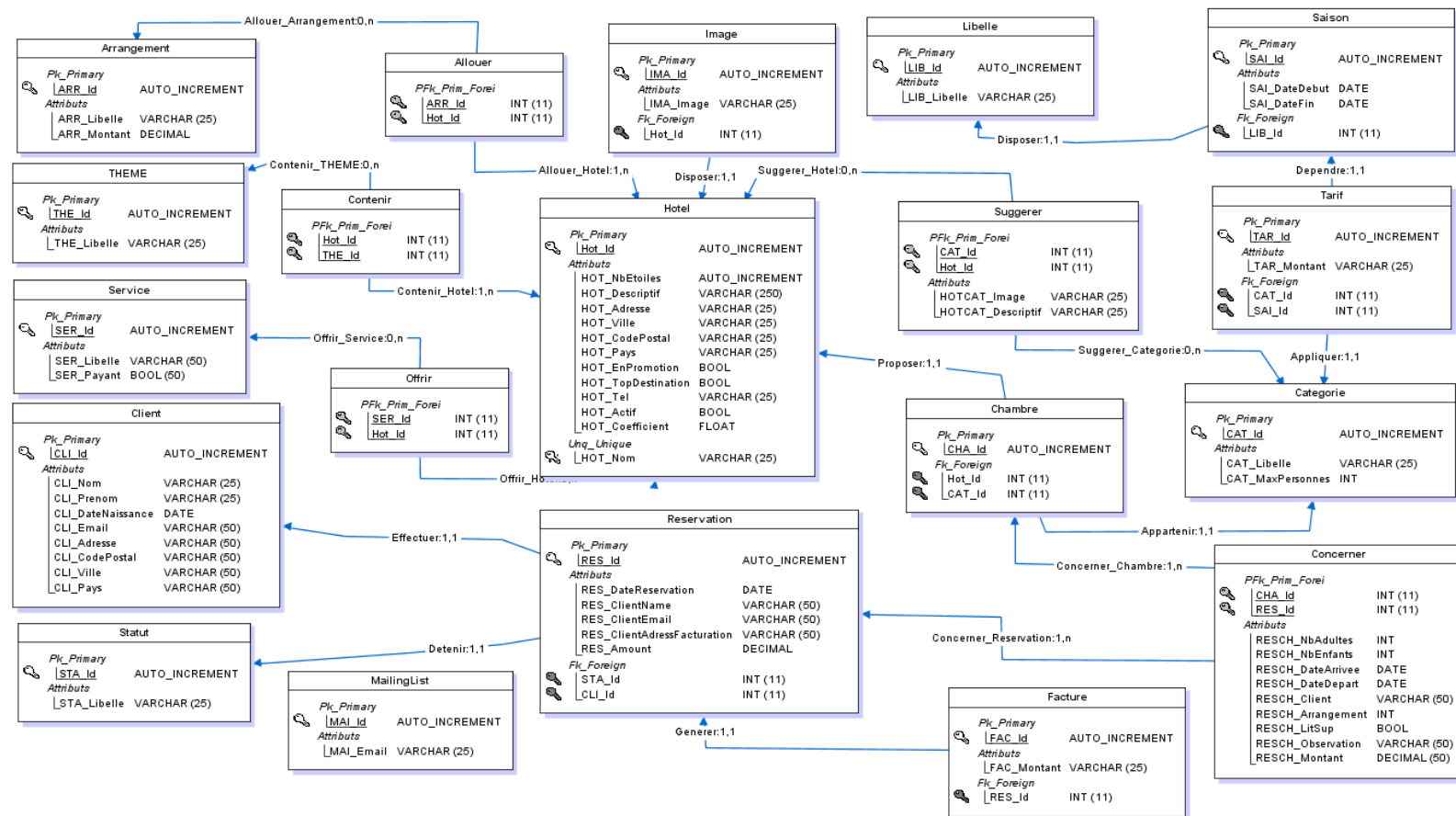


Schéma 15 : Schéma relationnel de la base de données

## 3. Tables & contraintes métier

### 3.1 Hotel

Un hôtel ne se caractérise par les attributs suivants :

- Un identifiant unique
- Nom
- Nombre d'étoiles
- Descriptif
- Adresse
- Numéro de téléphone
- Un coefficient qui sert à calculer le tarif des chambres
- Champ booléen qui indique si l'hôtel est en promotion
- Champ booléen qui indique si l'hôtel fait partie des top destination

Tous les champs de cette entité sont obligatoires.

Le nom d'un hôtel doit être unique du fait que deux hôtels ne peuvent pas avoir les mêmes noms.

Un hôtel doit disposer d'au moins une image, proposer une ou plusieurs formules de restauration « Arrangement ». Les arrangements disponibles seront les suivants :

- Chambre seul (sans formule)
- Petit déjeuner
- Demi-pension
- Pension complète

Chaque hôtel appartient au moins à un thème touristique et peut proposer plusieurs services gratuits ou facturables.

Chaque hôtel suggère plusieurs catégories de chambres en location. Pour chaque catégorie, chaque hôtel peut disposer de plusieurs chambres.

### 3.2 Image

Une image se caractérise par les éléments obligatoires suivants :

- Un identifiant unique
- Champ contenant le lien hypertexte vers une image

Chaque image appartient à un seul hôtel.

### 3.3 Arrangement

Un arrangement (formule culinaire) est composé des attributs obligatoires suivant :

- Un identifiant unique
- Un libellé
- Un prix qui peut être égal à 0 en cas de gratuité

Un arrangement peut être proposé par plusieurs hôtels.

### **3.4 Thème**

Un thème est caractérisé par un identifiant unique ainsi qu'un libellé du thème, tous ces champs sont obligatoires.

Un thème peut contenir plusieurs hôtels.

### **3.5 Service**

Un service se caractérise par les éléments suivants :

- Un identifiant unique
- Un libellé
- Un champ indiquant si le service est payant ou pas
- Un tarif

Un service peut être proposé par plusieurs hôtels. Dans le cas où il est payant, le tarif du service varie d'un hôtel à un autre. Tous les attributs sont obligatoires sauf le tarif.

### **3.6 Chambre**

Une chambre est caractérisée par un identifiant unique. Chaque chambre appartient à une seule catégorie et est proposée par un et un seul hôtel. Chaque chambre peut être concernée par plusieurs réservations.

### **3.7 Catégorie**

Une catégorie est reconnue à travers les attributs obligatoires suivants :

- Un identifiant unique
- Un libellé
- Le nombre maximum d'adultes autorisé

Une catégorie de chambre dispose aussi d'une image mais celle-ci varie d'un hôtel à un autre. Une catégorie applique un tarif, celui-ci est défini selon la saison.

### **3.8 Tarif**

Un Tarif détient un identifiant unique et un montant. Celui-ci dépend d'une seule saison et appliqué par une seule catégorie.

### **3.9 Saison**

Une saison est caractérisée par un identifiant unique, un libellé, une date de début et une date de fin.

Le libellé de la saison peut contenir une deux valeur suivantes : « Haute » et « Basse ».

Tous les attributs d'une saison sont obligatoires.

### **3.10 Client**

Un client est caractérisé par les attributs suivants :

- Un identifiant unique
- Nom
- Prénom
- Date de naissance

- Adresse e-mail
- Adresse
- Code postal
- Ville
- Pays

Tous les attributs du client sont obligatoires.

Un client peut s'inscrire aux « MailingList » et peut effectuer plusieurs réservations.

### ***3.11 MailingList***

Chaque enregistrement de la mailing liste est caractérisé par un identifiant unique et un champ contenant l'adresse email. Tous les champs sont obligatoires.

### ***3.12 Réservation***

Une réservation est caractérisée par les éléments obligatoires suivants :

- Un identifiant unique
- Date de réservation
- Nom du client pour lequel la réservation est effectuée
- Email du client pour lequel la réservation est effectuée
- Adresse de facturation

Chaque réservation peut être effectuée par un et un seul client.

Chaque réservation peut changer de statut selon les opérations qu'elle subit mais dispose d'un seul statut à la fois.

Chaque réservation peut concerner une ou plusieurs chambres. Pour chaque réservation de chambre, il faut prévoir les attributs additionnels suivants :

- Nombre d'adultes pour cette chambre
- Nombre d'enfants
- Date d'arrivée
- Date de départ
- Champs qui contiennent le nom complet de l'occupant
- L'arrangement inclus pour cette chambre
- Le montant total pour cette chambre
- Une éventuelle observation
- Un champ qui indique la nécessité d'un lit supplémentaire

Les attributs de chaque chambre réservée sont obligatoires sauf pour l'observation.

Une réservation peut générer une facture ou plusieurs factures.

### ***3.13 Statut***

Un statut est caractérisé par un identifiant unique et un libellé. Ces attributs sont obligatoires. Les statuts de réservation disponibles seront les suivants :

- Enregistrée
- Confirmée
- Annulée

### ***3.14 Facture***

Une facture dispose d'un identifiant unique et un montant total. Chaque facture est générée par une seule réservation.

## **4. Tables & relations**

En traduisant mon schéma entité-association en schéma relationnel, on constate certains changements au niveau des associations. Certaines se traduisent en tables intermédiaires alors que les autres disparaissent. Je vais aborder ces changements en décrivant d'abord les tables qui connaissent la disparition des associations. J'aborderai ensuite les tables intermédiaires qui ont remplacé certaines associations.

### ***4.1 Image***

Cette table représente les images des hôtels. En plus de l'id et le lien de l'image, l'identifiant de l'hôtel auquel chaque image fait référence s'est rajouter pour remplacer l'association qui existaient entre l'image et l'hôtel.

### ***4.2 Réservation***

Cette table représente les réservations, on constate au niveau de cette table que l'identifiant du client qui effectue la réservation ainsi que le statut de la réservation ont été importés afin de remplacer les associations.

### ***4.3 Facture***

Au niveau de la table Facture, nous constatons que l'identifiant de réservation a été importé dans cette table afin de remplacer l'association qui existait entre ces tables.

### ***4.4 Chambre***

Appartenant à un seul hôtel, l'identifiant de celui-ci a été importé dans cette table ainsi que la catégorie à laquelle elle appartient. Cette importation vise à faire référence à ces tables et remplacer les associations.

### ***4.5 Tarif***

La table Tarif représente l'entité tarif sauf que l'identifiant de la saison et la catégorie qui applique ces tarifs ont été importés. Par conséquent, cette importation remplace les associations.

### ***4.6 Saison***

La table Saison représente l'entité saison, l'identifiant de chaque saison a été transférée dans la table Tarif pour assurer la référence.

### ***4.7 Allouer***

Cette table représente l'association qui existait entre l'hôtel et l'arrangement. Vu les cardinalités plusieurs à plusieurs, il est évident que l'association soit remplacée par une table intermédiaire. Chaque enregistrement de cette table représentera les arrangements que chaque hôtel propose.



#### ***4.8 Contenir***

L'association entre l'hôtel et le thème a aussi été remplacé par une table intermédiaire afin de référencier tous les thèmes auxquels chaque hôtel appartient.

#### ***4.9 Offrir***

Cette table représente tous les services que chaque hôtel offre.

#### ***4.10 Concerner***

L'association qui existait entre la réservation et la chambre contenait déjà un ensemble d'attributs. L'identifiant de la réservation et de la chambre ont été importés afin de référencier, pour chaque chambre réservée, la réservation qu'elle concerne.

#### ***4.11 Suggester***

En plus des attributs de base de l'association qui existait entre l'hôtel et les catégories qu'il propose, l'identifiant de celui-ci ainsi que de la catégorie ont été importé pour bien référencer chaque enregistrement.

## **Synthèse**

---

Dans cette seconde partie, j'ai parcouru la phase d'analyse et ses différents aspects. J'ai commencé par décrire la méthodologie employée dans ce chapitre.

J'ai effectué l'analyse de ce projet selon deux approches différentes, la première porte sur une modélisation basée sur le langage unifié de modélisation UML alors que la deuxième repose sur une approche orientée processus avec le langage de modélisation de processus métier BPMN.

Les différentes fonctionnalités de notre projet analysées, j'ai procédé dès lors à la modélisation conceptuelle et logique de données qui recevront les informations générées et stockées par les différentes applications qui font l'objet de ce travail.

La partie analytique terminée, j'entame cette troisième et dernière phase : la conceptualisation des différentes solutions.

# Troisième Partie : Réalisation

---

# Introduction

---

Dans cette troisième et dernière partie, je décrirai le déroulement des différentes étapes que mon projet à parcouru depuis la phase d'analyse des besoins jusqu'au déploiement.

Lors de l'analyse fonctionnelle des différentes fonctionnalités, j'avais dans le travail simultanément pour l'application mobile et web api. Néanmoins, pour la phase de développement, le procédé a changé puisque l'application mobile a besoin d'une source de données pour assurer les différents tests et fonctions. C'est la raison pour laquelle j'ai commencé par le développement de l'application web api avant d'entamer celui de l'application mobile.

## Chapitre I : Base de données

---

### Section I : Conception

Le dernier point abordé dans la partie précédente concernait la structure de données. J'ai réalisé mon schéma entité-association à l'aide d'un outil performant et gratuit à la fois, cet outil est distribué sous le nom « JMerise »<sup>3</sup>. Le choix de cet outil est dû à mon expérience avec l'outil puisque j'ai commencé à l'utiliser depuis ma deuxième année d'études à l'Ephec. La figure 1 de la page suivante nous donne une idée sur l'interface graphique de cet outil.

Ce qui fait aussi la force de cet outil est qu'il offre à son utilisateur une certaine rigueur au niveau de la conception. En travaillant sur cet outil, ça m'évite d'ignorer certains détails importants au niveau des entités comme les identifiants ou de les déclarer en clé primaire, je profite surtout de cette aide précieuse pour me concentrer sur d'autres détails importants tels que les cardinalités qui pourrait changer complètement la structure d'une base de données.

Je me suis assuré de créer toutes mes entités et de bien paramétrer mes cardinalités, cet outil me permet d'anticiper les ambiguïtés éventuelles. Prenons comme exemple une entité qui dispose de deux associations vers d'autres entités, le fait de nommer l'association de la même manière (même action d'association) pourrait créer une ambiguïté même au niveau de la base de données.

Après avoir vérifié que mon schéma est correct sur tous les niveaux, je procède à la génération du schéma relationnel. Cette conversion est exécutée grâce à l'outil JMerise sur base du schéma entité « association » initial. Une observation méticuleuse du schéma généré pourrait nous aider à nous assurer de la structure définitive de la base de données. Une simple modification du schéma entité « association » pourrait nous générer un schéma relationnel différent du premier.

Après la génération des différents schémas, on peut générer le script de création de la base de données. A l'aide de JMerise, la création du script est facile et peut se faire pour plusieurs types de serveurs de données (SQL server, oracle, Access...etc.)

---

<sup>3</sup> JMerise est téléchargeable directement en suivant ce lien : <http://www.jfreesoft.com/JMerise/>

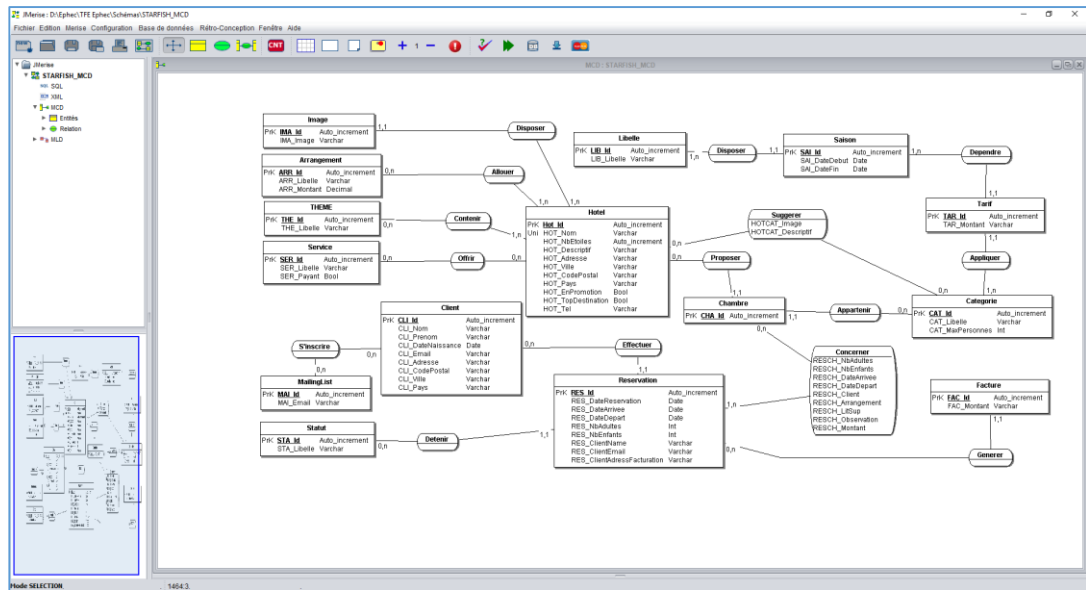


Figure 1 : Interface graphique de l'outil de modélisation conceptuelle et logique de données

## Section II : Création

Dans le cadre de ce projet, je crée une base de données sous SQL server et celle-ci sera nommée « StarFishHotels ». Au niveau de la « collation », on va préciser l'insensibilité à la casse et aux accents.

Une fois créée, on peut importer notre script SQL précédemment généré avec JMerise et l'exécuter afin de générer les tables et les liens nécessaires. Après l'exécution du script, notre base de données est prête à recevoir des données.

Pour faciliter la recherche et surtout structurer l'accès aux données stockées dans notre base de données « StarFishHotels », j'explique dans la section suivante les procédures stockées utilisées.

## Section III : Gestion des flux de données

Afin de mettre en application les connaissances acquises à l'Ephec en matière d'administration des bases de données, la gestion des flux d'informations en entrée et en sortie sera implémentée selon deux approches.

La première porte sur l'utilisation des procédures stockées, cette technique sera surtout utilisée pour la plupart des fonctionnalités, qui interagissent avec les utilisateurs disposant des rôles « anonyme » et « client », sauf celle qui concerne la création de réservation.

La deuxième approche concerne la récupération et l'insertion des données à l'aide d'Entity Framework<sup>4</sup> qui charge un modèle de données relationnel dans l'application et qui permet de gérer les requêtes d'insertion et de recherche de données. Les différentes fonctionnalités qui interagissent avec l'administrateur seront implémentées selon cette approche y compris la création des nouvelles réservations.

<sup>4</sup>Entity Framework : Programme qui assure l'interfaçage entre une application et une base de données relationnelle afin de simuler une base de donnée orientée objet.

Pour chacune des approches utilisées, j'exposerai une série de méthodes importantes employées dans ce projet. Ces dernières seront regroupées par domaine d'application.

## 1. Procédures stockées

### 2.1 Gestion des hôtels

Procédure	Descriptif
GetAllHotels	Procédure qui retourne l'ensemble des hôtels enregistrés dans la base de données.
GetAllHotelsByCity	Procédure qui retourne la liste des hôtels situés dans la ville passée en paramètre.
GetAllHotelsByCountry	Procédure qui retourne la liste des hôtels situés dans le pays passé en paramètre.
GetHotelsByTheme	Procédure qui retourne l'ensemble des hôtels situés dans la ville passée en paramètre.
GetHotelDetails	Procédure qui retourne toutes les informations de l'hôtel passé en paramètre.
GetAllPromotion	Procédure qui retourne tous les hôtels qui sont en promotion.
GetAllTopDestination	Procédure qui retourne la liste des hôtels qui sont dans les top destinations.
GetHotelAvailability	Procédure qui récupère les disponibilités de l'hôtel passé en paramètre.
GetHotelAvilibleBeds	Cette procédure retourne la capacité d'accueil d'un hôtel en comptant le nombre de lits libres.
GetHotelTarifsTable	Cette procédure récupère la table tarifaire d'un hôtel passé en paramètre.

### 2.2 Gestion des réservations

Procédure	Descriptif
GetAllReservations	Procédure qui retourne l'ensemble des réservations.
GetAllReservationsByCustomerId	Cette procédure retourne la liste des réservation d'un client passé en paramètre
GetReservationRooms	Cette procédure retourne la liste des chambres réservées pour une réservation passée en paramètre.
GetReservationDetailsByReservationId	Procédure qui retourne le détail d'une réservation passée en paramètre.

### 2.3 Gestion des utilisateurs

Procédure	Descriptif
GetMailingList	Cette procédure récupère les adresses des utilisateurs inscrits pour recevoir des newsletters.

GetCustomerDetails	Procédure qui retourne les informations de l'utilisateur passé en paramètre.
GetCustomerByNetUserId	Récupération de l'id d'un utilisateur à travers son id NetUser.

## 1. Entity Framework

### 2.1 Gestion des hôtels

Procédure	Descriptif
CreateNewHotel	Fonction responsable de la création d'un nouvel hôtel.
UpdateHotel	Cette fonction se charge de la mise à jour des informations stockées et qui concernent un hôtel.
DeasctivateHotel	Fonction qui s'occupe de la désactivation des hôtels.

### 2.2 Gestion des réservations

Procédure	Descriptif
NewReservation	Fonction responsable de la création d'une nouvelle réservation pour les clients.
UpdateReservation	Cette fonction est responsable de la mise à jours d'une réservation passée en paramètre.
CofirmReservation	Fonction employée pour la confirmation d'une ou plusieurs réservations.
CancelReservation	Fonction employée pour l'annulation d'une ou plusieurs réservations.

### 2.3 Gestion des utilisateurs

Procédure	Descriptif
CreateNewUser	Fonction qui permet aux administrateur de créer un nouvel utilisateur et lui assigner le rôle souhaité.
UpdateUser	Fonction responsable de la mise à jour du profil utilisateur.
DeactivateUser	Fonction permettant de désactiver un utilisateur.

### 2.4 Divers

Procédure	Descriptif
SendPromos	Fonction qui permet d'envoyer des promotions aux utilisateurs inscrits aux new lettres (mailing listes).

GetCarousel	Fonction qui récupère les image du carrousel affiché en haut des pages de l'application web.
UpdateCarousel	Fonction qui met à jours les images du carrousel.
SendNotification	Cette fonction permet l'envoi des notifications depuis la console d'administration.

## Chapitre II : Développement applicatif

### Section I : Application web

Dans cette partie, consacrée au développement de l'application web, je décrirai la méthodologie et l'outil de développement utilisés, les évolutions possibles du projet ainsi que les différentes difficultés rencontrées.

#### 1. Type de projet

J'ai préféré, dans le cadre de ce projet, d'utiliser Visual studio pour la création et le développement du projet. Ce dernier sera de type « application web api », ce type d'application permet de gérer à la fois une application web classique et une api (interface de programmation applicative). Mon choix se justifie aussi par le niveau de sécurité qu'offre le framework Asp.net dédié au développement web.

#### 2. Architecture du projet

Une fois créé, le projet dispose de la même architecture qu'une application web classique en ce qui concerne l'arborescence et les composants du projet sauf pour la section des contrôleurs qui sera devisée en deux groupes. Le premier groupe est de type contrôleur classique qui garantit la gestion des actions web selon le pattern MVC. Le deuxième groupe concerne les contrôleur api qui se chargeront de la gestion des appels venant des autres serveurs et applications.

Afin de respecter l'architecture dressée dans la première partie de ce rapport, qui est illustrée par le schéma 3, portant sur l'architecture de la solution, il a fallu instaurer cette notion de couches afin de gérer les échanges de données comme suit :

La récupération et l'insertion des données de lance puis les différents contrôleurs vers la couche business, cette dernière échange avec la couche de donnée qui exécute à son tour l'opération de chargement ou insertion de données vis-à-vis de la base de données. Le but de la mise en place des différentes couches est surtout préventif car cette méthodologie protège au maximum l'accès aux données.

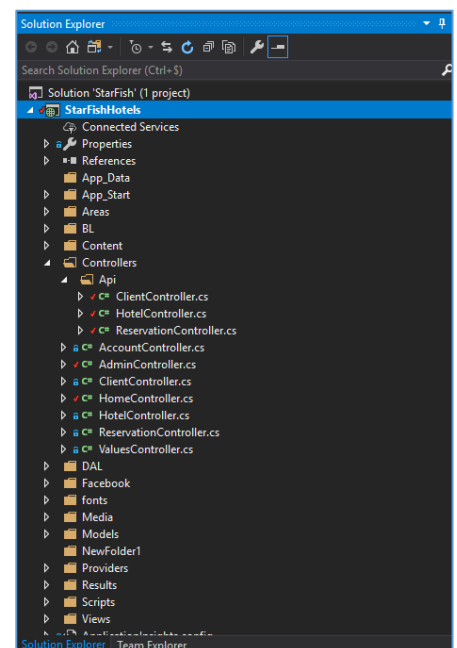


Figure 2 : Contrôleur classiques et api - Explorateur de la solution

### 3. Liaison du projet à la base de données

Dans cette section, je vais aborder l'établissement des liens entre l'application web et la base de données. Il a fallu, pour assurer des échanges sécurisés, charger le model de la base de données dans l'application. Ce modèle de donnée est une projection de la structure des entités ainsi que les liens entre elles grâce au mappeur Entity Framework.

Le model de données qui concerne notre base de données a été chargé dans un dossier créé préalablement et nommé « DAL » (Data Access Layer / Couche d'Accès aux Données). Afin de sécuriser l'accès aux données, les requêtes de la couche business portant sur des informations stockées dans la base de données passeront par la « DAL » qui englobe le model de données.

### 4. Mesures de sécurité

Un élément primordial dans le développement des projets informatiques est celui qui concerne la sécurité du code source. Une des grandes mesures de sécurité vise à prévoir un système de sauvegarde des différentes versions du code source. C'est la raison pour laquelle j'ai choisi d'utiliser Visual studio Team Services, qui me sert déjà d'outil de gestion de projet, pour gérer l'évolution de mon code source et les différentes sauvegardes.

Grâce à cette mesure, j'ai pu revenir sur mes anciennes versions de code suite à des mauvaises manipulations ou simplement récupérer mon projet sur un autre ordinateur sur lequel je pouvais continuer à développer mon projet sans devoir transporter mon ordinateur personnel partout.

### 5. Difficultés rencontrées

La phase du développement pour l'application web s'est bien passée en général malgré certains bugs du quotidien, j'ai rencontré certaines difficultés pendant le développement de quelques fonctionnalités telle que la gestion du login via les réseaux sociaux. Mais avec un peu de recherche, j'ai pu trouver le bon chemin pour accomplir l'implémentation de cette fonctionnalité. J'estime qu'on apprend beaucoup de choses en matière de développement à travers les recherches qu'on effectue pour solutionner un bug rencontré. En ce qui me concerne, les forums d'échange informatique m'ont beaucoup aidé dans mon parcours de développement.

## Section II : Application mobile

Cette deuxième section sera consacrée au déroulement du développement de l'application mobile. Je vais aborder la méthodologie et les outils utilisés pour la réalisation la solution.

### 1. Type de projet

Dans le cadre du développement de l'application mobile, j'ai choisi d'utiliser le même environnement. Visual Studio permet développer des application cross plateforme grâce à Xamarin.Forms. Cet outil permet la création d'applications mobiles qui peuvent fonctionner sur les trois plateformes mobiles (Android, Ios, Windows). J'ai donc créé mon projet sous Xamarin.Forms et je donnerai de plus amples informations dans les paragraphes suivants.



## 2. Architecture du projet

Une fois générée, notre solution dispose de quatre projets, trois d'entre eux concernent les différentes plateformes mobiles et chacun d'entre eux se charge de l'exécution de l'application mobile sur la plateforme pour laquelle il est destiné. Le quatrième projet restant est une librairie de classes (Class Library) qui compte contenir les parties de code commun entre les plateformes (logique de l'application). J'ai beaucoup travaillé sur cette librairie commune la plupart du temps vue que c'est elle qui contient la plus grande partie de la logique de l'application mobile. Cette librairie contient toutes les vues de l'application et gère toutes les requêtes interaction entre l'utilisateur et les différentes sources de données.

## 3. Mesures de sécurité

Afin d'éviter toute perte du code source ainsi qu'une chronologie de versions, j'ai utilisé le même principe que pour l'application web en utilisant Visual Studio Team Services. Ce qui m'a vraiment assuré un bon suivi des versions et une sécurité en cas de crash.

## 4. Difficultés rencontrées

En matière de développement mobile, j'ai fait face à beaucoup de difficultés vu que je suis encore novice en développement mobile. Ceci reste mon premier projet en développement mobile mais restera un des meilleurs et ne sera surtout pas le dernier. La recherche sur internet ainsi que l'aide et l'orientation de certaines personnes de mon entourage m'ont beaucoup aidé pour faire face aux différentes difficultés que j'ai pu rencontrer tout au long du développement.

# Chapitre III : Le déploiement

---

Ce chapitre est consacré au déploiement de mon projet une fois que tous le développement est accompli. Afin de faire fonctionner mon application web api, j'ai opté de la déployer sur mon espace sur Microsoft Azure Cloud, cet environnement offre plusieurs possibilités en matière de déploiement d'applications web et de bases de données et d'autres fonctionnalités telle que la virtualisation des systèmes d'exploitation et des réseaux.

Quant à l'application mobile, le projet est archivé sous forme de release (version) et exporté sous forme d'application mobile selon les plateformes cibles.

L'outil que j'ai choisi d'utiliser pour le suivi de cet application est Microsoft app center. Cet environnement permet de suivre l'évolution de l'utilisation de l'application, remonter les crashes ainsi qu'envoyer des notifications.

## Section I : Microsoft Azure

Sur mon espace réservé sur Microsoft Azure, il a fallu d'abord créer une instance pour l'application web et une pour la base de données SQL server dans le but d'y déployer notre base de données ainsi que notre application web api.

La figure 4 (page suivante) nous montre les deux instances créées et prêtes à recevoir les projets.

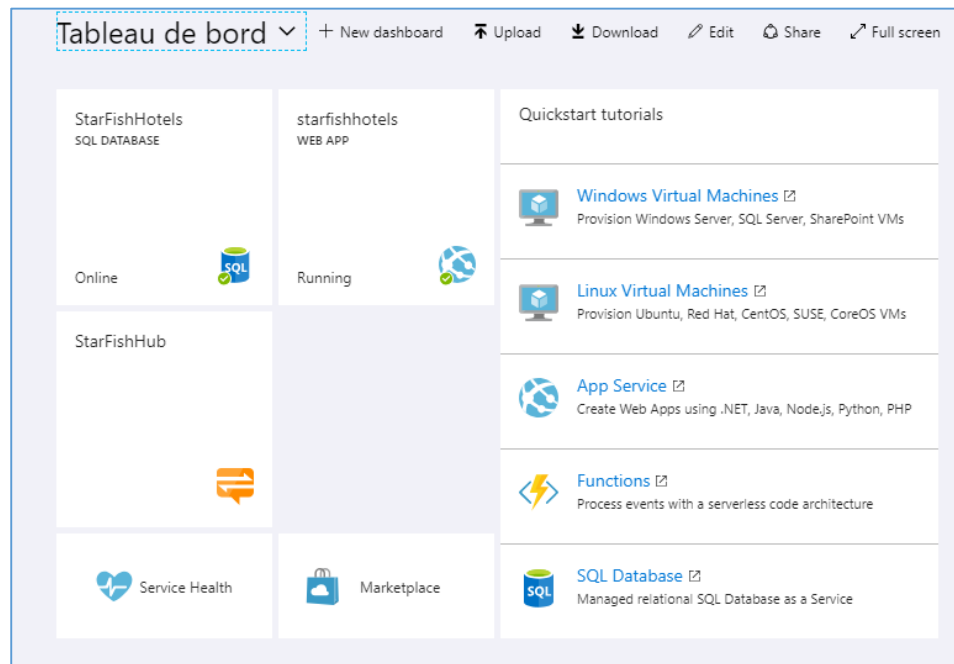


Figure 3 : Instances créées pour la base de données et l'application web

### 1. Base de données

J'ai commencé par déployer la base de données car sans elle, mon application mobile ne peut pas fonctionner et surtout que la déployer en premier me permet de récupérer la chaîne de caractères de connexion (connexion string), élément indispensable permettant d'établir des connexions vers notre base de données.

J'ai procédé au déploiement de la base de données grâce à « Microsoft Data Migration Assistant » un assistant graphique fourni par Microsoft et permettant de migrer toutes les versions de base de données SQL server vers azure avec une gestion de compatibilité des bases de données.

Après la migration de ma base de données, il est temps de déployer l'application web et mobile.

### 2. Application web

Pour pouvoir déployer mon application web api et la faire fonctionner, j'ai créé une instance d'application web sur mon espace cloud d'Azure préalablement avant de la déployer depuis Visual Studio directement vers Microsoft Azure.

Avant d'effectuer le déploiement de l'application, il est nécessaire d'adapter le fichier « webconfig » pour qu'il puisse pointer vers la base de données créée au niveau de Microsoft Azure et non vers la base de données locale.

Après ce changement, j'ai procédé au déploiement de mon application web. Tout s'est déroulé comme prévu et maintenant je dispose du lien direct vers mon application web api : <http://starfishhotels.azurewebsites.net>

## Section II : Microsoft app center

Pendant le développement de l'application mobile, j'ai choisi de l'interfacer avec l'environnement de gestion pour application mobile « Microsoft App Center » car l'implémentation de cet interfaçage reste simple et efficace. Après que j'ai créé mon espace personnel sur l'App center, il a fallu créer une instance d'application par plateforme d'exécution. Il est nécessaire de créer une instance d'application pour chaque plateforme ciblée (Android, Ios et Windows). Chaque instance nous offre les mêmes avantages d'analyse et statistiques, analyse des crashes ainsi que la gestion des notifications.

La figure suivante (Figure 5) nous illustre l'interface graphique de la console de gestion de l'instance Android de l'application mobile.

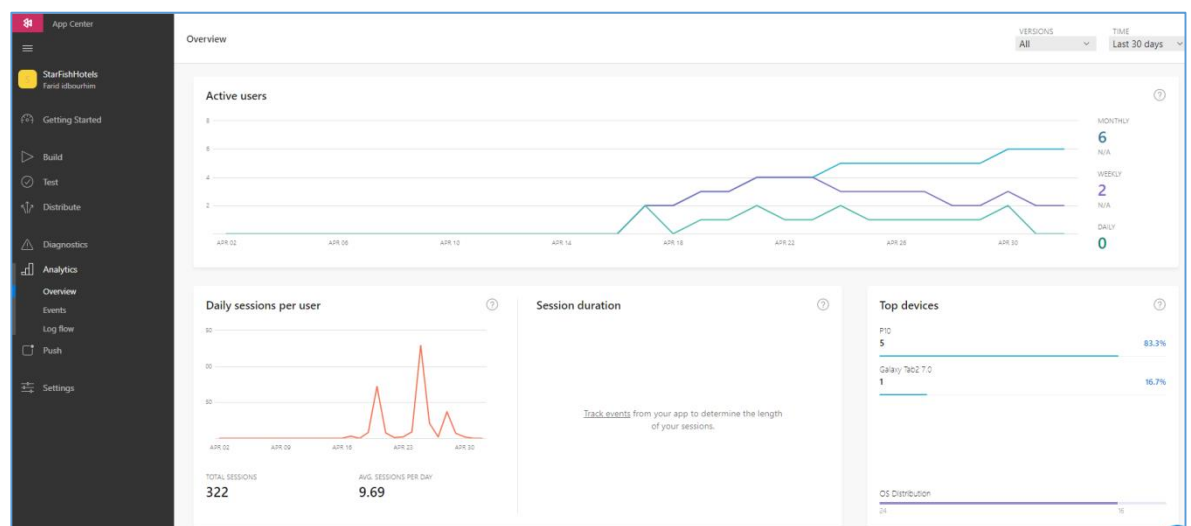


Figure 4 : Interface graphique de la console de gestion dédiée à l'application mobile

### 1. Analytics

Grâce à l'App center, on peut suivre l'utilisation de l'application mobile à travers le monde. Ce suivi peut concerner le volume d'utilisateurs, le type de appareils sur lesquels elle est installée ainsi que le temps moyen d'utilisation par jour.

### 2. Gestion des crashes

L'un des gros avantages de l'App center est qu'il permet de faire un diagnostic de l'application en remontant tous les bugs et crashes et surtout en précisant l'origine de ces derniers. Cela m'a permis plusieurs fois d'accélérer le débogage des crashes sans devoir passer beaucoup de temps à comprendre la raison de ces derniers.

### 3. Gestion des notifications

Les notifications jouent un grand rôle dans le monde du mobile. Elles permettent de garder un lien avec l'utilisateur en le tenant à jour des offres et promotion ainsi en l'incitant à revenir sur l'application. Pour ce fait, j'ai effectué les tests d'envoi de notification avec succès et je suis parvenu à implémenter cette fonctionnalité au niveau de l'application web api, dans l'interface d'administration, afin de permettre aux administrateurs de publier les notifications sans devoir s'authentifier dans l'App center puisque les informations d'authentification sont déjà implémentées au niveau de l'application web api.

## Synthèse

---

J'ai parcouru, dans cette dernière partie, les différentes étapes qui ont marqué le développement et le déploiement des deux solutions attendues par le client.

J'ai commencé par la création de la base de données à travers l'exécution du script générée pour cette fin ainsi que les procédures stockées responsables de la récupération des informations nécessaires.

J'ai aussi abordé les différents aspects relatifs au développement applicatif, ceci concernait la réalisation de l'application web api et mobile. J'ai aussi mis le doigt sur certaines mesures de sécurité au niveau de l'accès au données à travers les différentes couches de transition de données et au niveau de l'évolution du code source à travers la gestion du versionning.

Enfin, j'ai illustré le déroulement du déploiement effectué pour les différentes applications ainsi que pour la base de données.

# Conclusions et recommandations

---

Pour la réalisation de ce travail, j'ai parcouru les différentes étapes indispensables que chaque projet informatique traverse. En allant de l'analyse des besoins jusqu'au déploiement des applications et base de données, il est nécessaire de respecter certaines règles et dépendances. La notion de dépendance a vraiment sa place dans le développement car chacune des phases du projet dépend de sa précédente. Si cette dernière n'a pas eu le temps de réflexion nécessaire, le reste du projet risque de ne pas répondre aux attentes du client.

L'analyse des besoins est la clé de la réussite de tout projet. Si cette étude est effectuée avec soin, elle donnera lieu à une analyse correcte et optimale. Ce qui permettra par la suite une solution qui répond correctement aux attentes du client.

Dans le cadre des évolutions possibles du projet « StarFish Hotels » je conseillerai à mon client de penser d'implémenter les fonctionnalités nécessaires pour la gestion opérationnelle de chacun des hôtels qu'elle gère. Ceci comportera les fonctionnalités suivantes :

- Affectation des chambres au client
- Ajout des point ventes (Restauration, Soins et autres...etc.)
- Check-in et Check-out
- Gestion de la facturation

Cette proposition va donner la possibilité de suivre le client depuis la réservation jusqu'à son départ de l'hôtel. Cela permet aussi d'avoir un seul outil complet pour la gestion de son activité.

# Bibliographie

---

## 1. Livres

- Creating Mobile Apps with Xamarin.Forms Book First Edition par CHARLES PETZOLD.

## 2. Guides en ligne

- GitHub: Settings Plugin for Xamarin and Windows, Par James Montemagno.

## 3. Formations en ligne

- Udemy: Xamarin Forms: Build Native Cross-platform Apps with C#, par Mosh Hamedani.
- Pluralsight: ASP.NET Ajax JavaScript and JQuery, par Fritz Onion et Scott Allen.
- Pluralsight: Introduction to Bootstrap, par Scott Allen.
- Series YouTube: ASP.NET Web API, Par Kudvenkat.
- Series YouTube: Cross Platform App with Xamarin Forms, Par Housseem Dellai.

# Annexes

## Annexe I : Page index de l'application web

StarFish hotels

Hotels

Promotions

Register

Login

132

Voyagez en famille  
à partir de  
79€ /personne

Recherche Rapide

Pays:  
Tous

Date d'arrivée:

Adults:  
12 ans +

Enfants:  
0-12 ans

Ville:  
Toutes

Date de départ:

Rechercher

NewsLetters

Inscrivez-vous afin de recevoir nos meilleures offres:

S'inscrire

Reseaux sociaux

Like us on Facebook

Follow us on Twitter

Partenaires

Neckermann Polska  
TUI group  
One&Only RESORTS

Promotions

Le Tivoli  
★★★★  
à partir de: 70 €  
Situé à 10 minutes à pied du centre-ville d'Agadir, à 250 mètres de la plage, l'hôtel Blue Sea Le Tivoli compte une piscine extérieure, une plage privée et 2 courts de tennis.  
Réserver Découvrir

Royal Atlas  
★★★★★  
à partir de: 64 €  
Situé en bord de mer, le Royal Atlas & Spa est un établissement 5 étoiles proposant des chambres climatisées dotées d'un balcon.  
Réserver Découvrir

The Pearl Marrakech  
★★★★★  
à partir de: 58 €  
Le Pearl Marrakech dispose d'une piscine sur le toit, de 3 restaurants et d'un bar-salon doté d'une galerie d'art.  
Réserver Découvrir

Voir toutes les promotions

Top Destinations

Le Tivoli  
★★★★  
à partir de: 70 €  
Situé à 10 minutes à pied du centre-ville d'Agadir, à 250 mètres de la plage, l'hôtel Blue Sea Le Tivoli compte une piscine extérieure, une plage privée et 2 courts de tennis.  
Réserver Découvrir

The Pearl Marrakech  
★★★★★  
à partir de: 58 €  
Le Pearl Marrakech dispose d'une piscine sur le toit, de 3 restaurants et d'un bar-salon doté d'une galerie d'art.  
Réserver Découvrir

One&Only Royal Mirage  
★★★★★  
à partir de: 64 €  
Cet établissement a également été bien noté pour son excellent emplacement à Dubai  
Réserver Découvrir



## Annexe II : Détail d'un hôtel

Informations de l'hôtel

The Pearl Marrakech - Marrakech, Maroc

★★★★★


DESCRIPTION GENERALE

Le Pearl Marrakech dispose d'une piscine sur le toit, de 3 restaurants et d'un bar-salon doté d'une galerie d'art. Lors de votre séjour, vous pourrez découvrir les boutiques de luxe sur place, ou visiter le palais royal et la place Jamaâ El Fna en 15 minutes de marche. Affichant une décoration aux influences orientales, les chambres et les suites élégantes de l'établissement The Pearl Marrakech comprennent un balcon privé, un coin salon ainsi qu'une salle de bains privative pourvue de peignoirs et de chaussons. Cet hôtel 5 étoiles vous servira tous les matins des viennoiseries et des gâteaux faits maison au petit-déjeuner. Vous pourrez également déguster des spécialités italiennes, françaises et marocaines sur les terrasses, tout en admirant la vue panoramique. Accessible moyennant des frais supplémentaires, le spa du Pearl Marrakech vous prodiguera des soins de beauté et comporte un hammam ainsi qu'un sauna. Vous pourrez stationner gratuitement sur place. L'aéroport de Marrakech-Menara se trouve à 13 minutes en voiture de l'hôtel. Ce quartier (L'Hivernage) est un choix idéal pour les voyageurs qui s'intéressent à ces thèmes : les climats ensoleillés, les habitants chaleureux et l'ambiance.

PLUS DE DETAILS

Nombre de chambres :	20
Telephone :	00212435526255
Adresse :	3 , Rue des Temples, quartier hivernage 40000, Marrakech - Maroc
Check-in:	14:00
Check-out:	12:45
Arrangements :	CHAMBRE SEULE + 0,00 €, PETIT DEJEUNER + 25,00 €,
Type de chambres :	Double, Single, Suite executive, Suite junior, Suite royale, Triple,
Thèmes de l'hôtel :	Famille,
Services Payants :	COURT DE TENNIS, SPA, JET SKI, KIDS CLUB, VISITE GUIDEÉE, NAVETTE AEROPORT, TRECKING,
Services offerts :	INTERNET, ANIMATION, CONCIERGERIE,

GALLERIE



Réserver

Fermer



## Annexe III : Page de réservation d'un hôtel

StarFish hotels

Hotels

Promotions

Mes Réservations

Bienvenue : user@a.a

Logout

Le Tivoli

★★★★

SEJOUR

Date d'arrivée:

19-05-18 00:00:00

Date de départ:

20-05-18 00:00:00

Adultes:


1

Enfants:

0

Revoir

CHAMBRES



Type de chambre: Single

Chambre équipée d'un lit d'une personne. Le lit pouvant être un grand lit pouvant recevoir 2 personnes et pas forcément un "petit lit" d'une personne.


Quantité:

0

Prix / nuit:

59,00

€



Type de chambre: Double

Chambre d'hôtel équipée d'un grand lit de 2 personnes et pouvant donc accueillir 2 personnes.


Quantité:

0

Prix / nuit:

76,00

€



Type de chambre: Triple

Chambre d'hôtel équipée de 3 couchages et pouvant donc accueillir 3 personnes. La configuration peut être 3 lits d'une personne ou bien 1 lit double de 2 personnes et 1 d'une personne.


Quantité:

0

Prix / nuit:

109,00

€



Type de chambre: Suite Junior

Chambre composé d'une ou plusieurs chambres et d'une ou plusieurs pièces supplémentaires (salon) avec une capacité maximale de 3 personnes.


Quantité:

0

Prix / nuit:

139,00

€



Type de chambre: Suite exécutive

Chambre composé d'une ou plusieurs chambres et d'une ou plusieurs pièces supplémentaires (salon) avec une capacité maximale de 4 personnes.


Quantité:

0

Prix / nuit:

235,00

€



Type de chambre: Suite royale

Chambre composé d'une ou plusieurs chambres et d'une ou plusieurs pièces supplémentaires (salon) avec une capacité maximale de 4 personnes.

Quantité:

0

Prix / nuit:

359,00

€

Montant total:

0,00

€

Détails de facturation

Cliant:

client client

Adresse de facturation:

Rue de la gloire 1, Bruxelles 1000 - Belgique

Email de confirmation:

user@a.a

Réserver

55

## Annexe IV : Première page de paiement



**Numéro de commande : 248010115**  
**Total à payer : 45900.00 EUR**  
**Bénéficiaire : StarFish hotels**

Carte : transactions encryptées SSL

**Veuillez sélectionner une méthode de paiement en cliquant sur le logo.**



**Paiement sécurisé par**  
**ingenico**  
ePayments

  
powered by Symantec

[A propos de Ingenico](#) | [Protection de la vie privée](#) | [Sécurité](#) |  
[Informations légales](#)  
[Annulation](#)

## Annexe V : Deuxième page de Paiement



**Numéro de commande : 248010115**  
**Total à payer : 45900.00 EUR**  
**Bénéficiaire : StarFish hotels**

**Payer avec : VISA**

**Titulaire de la carte\* :**

**Numéro de la carte\* :**

**Date d'expiration (mm/aaaa)\* :**  /

**Code de vérification de la carte\* :**  [Qu'est-ce que c'est ?](#)

Un \* indique les champs obligatoires



**Paiement sécurisé par**  
**ingenico**  
ePayments

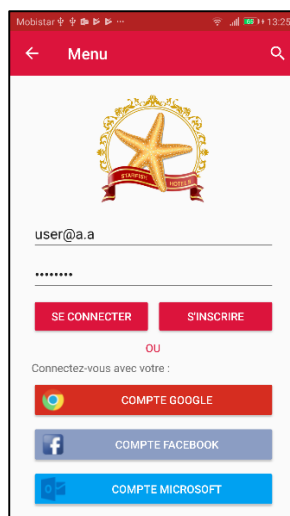
  
powered by Symantec

[A propos de Ingenico](#) | [Protection de la vie privée](#) | [Sécurité](#) |  
[Informations légales](#)  
[Retour](#) [Annulation](#)

## Annexe VI : Aperçu des vues de l'application mobile



Page d'accueil



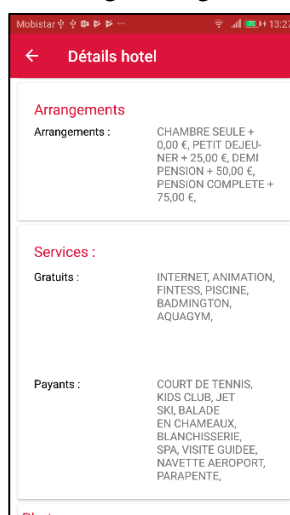
Page de login



Détail d'un hôtel



Détail d'un hôtel (suite)



Détail d'un hôtel (suite)



Détail d'un hôtel (suite)



Page de réservation



Page de réservation (suite)