

UNIVERSITE SIDI MOHAMED BEN ABDELLAH
FACULTE DES SCIENCES ET TECHNIQUES FES
DEPARTEMENT D'INFORMATIQUE



Projet de Fin d'Etudes

Licence Sciences et Techniques Génie Informatique

GDS

Systeme Centralisé de Réservation d'hôtels



Lieu de stage : G-Fit MAROC

Réalisé par :

HANAFI Ismail

TAHIFA Mohammed

Encadré par :

Mr. BENABBOU Abderrahim

Mr. YAHIAOUI Mohammed

Soutenu le 17/07/2010 devant le jury composé de :

Pr. Rachid BENABBOU

Pr. Abderrahim BENABBOU

Pr. Aicha MAJDA

Année Universitaire 2009-2010

Dédicace

A mes très chers parents,

Merci, pour votre patience, vos sacrifices et votre soutien tout au long de mon parcours.

Que Dieu vous garde pour moi et vous donne une vie pleine de santé.

A mon frère Naoufal, mes sœurs Amina, Hind, ma tante Maria.

A l'ensemble de ma famille, pour qui j'éprouve un grand respect.

A mon cousin HANAFI Zouhir, pour m'avoir encouragé et soutenu.

A mes professeurs.

A mes chers amis avec qui j'ai passé des instants inoubliables.

Je ne saurais exprimer ma reconnaissance la plus sincère et mon respect le plus profond pour toute la sympathie et l'affection dont ils ont fait preuve, ainsi qu'à leurs encouragements qui ont été pour moi un grand soutien moral tout au long de mes études.

A tous ceux qui m'ont soutenu, Merci infiniment.

Ismail

Dédicace

A mes chers parents

*Je ne saurais exprimer ma gratitude et mes profonds sentiments
par de simples mots*

A mes sœurs Samar, Salma, Inass

*Avec tout mon amour et mon affection, je vous souhaite plein
de bonheur à vous et à vos maris.*

A toute ma famille

A tous mes amis

A tous ceux qui m'aiment

*Pour tout le soutien que vous m'avez offert je vous dis **Merci**.*

Mohammed

Remerciements

Nous remercions ALLAH, l'Unique pour ses bienfaits trop souvent négligés.

Au terme de ce travail, nous tenons à exprimer notre reconnaissance et notre profonde gratitude

À :

- Monsieur ZAROUK Amine le Directeur Général de la société G-FIT MAROC, pour nous avoir ouvert les portes de son entreprise, et nous avoir offert l'opportunité de réaliser ce stage.*
- Monsieur YAHIAOUI Mohammed encadrant et Chef de projet au sein de G-FIT MAROC pour son soutien, pour le temps qu'il nous a consacré, ainsi que pour les conseils utiles qu'il nous a donnés tout au long de notre stage.*
- Monsieur ABDERAHIM BENNABOU notre encadrant à la FST de Fès.*
- Nos vifs remerciements s'adressent également aux membres du jury qui ont accepté d'évaluer notre travail.*
- Tous les enseignants de la FST à qui nous devons du respect. Et le personnel administratif pour leur sympathie et les services qu'ils nous ont rendus.*
- Nos très chers parents qui ont consenti tous les sacrifices pour notre éducation, nous seront toujours reconnaissants pour leurs soutiens moral et matériel.*

Liste des Abréviations

Abréviation	Désignation
JEE	Java Entreprise Edition
JPA	Java Persistence Api
MVC	Model View Controller
UML	Unified Modeling Langage
XML	extensible Mark up Langage
JDBC	Java Data Base Connectivity
JSF	Java Server Faces
SGBD	Système de Gestion des Bases de Données
IDE	Integrated développement environnement
SQL	Structural query langage.
IHM	Interface Homme Machine
POM	Project Object Model
XMI	XML Metadata Interchange
UI	User Interface
DAO	Data Access Object
ORM	Object Relational Mapping
SWF	Spring Web Flow

Liste des Figures

Figure 2 : Les références clients G-Fit	14
Figure 1 : Les objectifs	14
Figure 3 : Spring MVC	18
Figure 4 : Architecture MVC	19
Figure 5 : multicouche JPA	20
Figure 6 : Hibernate dans JPA	21
Figure 7 : principe de SWF	22
Figure 8 : exemple de View-state	23
Figure 9 : structure de Maven	25
Figure 10 : Diagramme de cas d'utilisation pour le client	29
Figure 11 : Diagramme de cas d'utilisation pour le partenaire	30
Figure 12 : Diagramme de cas d'utilisation pour l'administrateur	31
Figure 13 : Diagramme de déploiement	32
Figure 14 : Diagramme de déploiement	33
Figure 15 : Diagramme de classe	33
Figure 16 : Vue Hôtels	38
Figure 17 : Vue d'un Hôtel	39
Figure 18 : Vue de saisie de coordonnées	40
Figure 19 : Vue de Saisie de coordonnées détaillées du client	41
Figure 20 : Vue de confirmation de réservation	42
Figure 21 : Vue d'authentification du client	42
Figure 22 : Vue de visualisation de la réservation	43
Figure 23 : Vue d'annulation de la réservation	43
Figure 24 : Vue de Saisie des coordonnées d'hôtel	44
Figure 25 : Vue d'insertion des chambres d'un Hôtel	45
Figure 26 : Vue du formulaire du Responsable	45
Figure 27 : Vue de confirmation d'inscription	46
Figure 28 : vue d'authentification de l'administrateur	46
Figure 29 : Vue des nouveaux inscrits	47
Figure 30 : Vue de visualisation d'Hôtel	47

Liste des Tableaux

<i>Tableau 1: Fiche d'identité</i>	12
<i>Tableau 2 : tableau des métiers</i>	13
<i>Tableau 3 : tableau de View-state</i>	23
<i>Tableau 4 : Classe Hôtel</i>	34
<i>Tableau 5 : Classe Chambre</i>	34
<i>Tableau 6 : Classe Responsable</i>	35
<i>Tableau 7 : Classe Client</i>	35
<i>Tableau 8 : Classe Commentaire</i>	36
<i>Tableau 9 : Classe Carte Crédit</i>	36
<i>Tableau 10 : Classe Réservation</i>	36
<i>Tableau 11 : Classe paramètre</i>	37

Sommaire

Remerciements	4
Liste des Abréviations	5
Liste des Figures.....	6
Liste des Tableaux.....	7
Sommaire	8
Introduction	10
Chapitre 1	11
Présentation du cadre général du projet	11
1. Présentation générale de l'entreprise.....	11
1.1. Fiche d'identité	12
1.2. Les métiers	12
1.3. Les objectifs	13
1.4. Les références	14
2. Cadre général du projet	14
Chapitre 2	16
Analyse et Conception du projet	16
1. Contraintes principales de développement.....	16
1.1. Outils et Langages de développement	16
1.1.1. Framework JEE	16
1.1.2. Framework Spring	17
1.1.3. Spring MVC [Spring MVC par l'exemple, 06]	18
1.1.4. MySQL	20
1.1.5. Hibernate/JPA [Persistence Java 5 par la pratique, 07]	20

1.1.6.	Web Flow [Spring Web Flow 2 Web Development, 09]	22
1.1.7.	Framework JSF	24
1.2.	Structuration du projet java avec Maven	24
1.3.	Modélisation UML	25
1.3.1.	Présentation UML	25
1.3.2.	Présentation de l’outil TopCased	26
1.3.3.	Présentation d’Apache Tomcat	26
2.	Etude et spécification des besoins	27
2.1.	Spécification des besoins	28
2.2.	Diagramme des cas d'utilisation	28
3.	Etude et conception	32
3.1	Diagramme de déploiement	32
3.2	Analyse du domaine	33
Chapitre 3		38
Réalisation de l’application		38
1.	IHMs Client	38
1.1.	Réserver une chambre dans un hôtel	38
1.2.	Modifier ou annuler la réservation	42
2.	IHMS Partenaire	44
	Inscription dans le site	44
3.	IHMS Administrateur	46
	Gestion des demandes	46
Conclusion		48
Bibliographies		49
Webographie		49

Introduction

Les GDS sont des plates-formes électroniques de gestion des réservations qui permettent de connaître l'état du stock des différents fournisseurs de produits touristiques (compagnies aériennes, chaînes d'hôtels ...) et de réserver à distance. En clair, ce sont des services de commerce électronique très performants.

On pourra se demander pourquoi informatiser le domaine de réservation d'hôtels ? La crise économique qui a sévi en 2009 n'a pas épargné les acteurs de l'hôtellerie qui doivent multiplier les relais de croissance afin non seulement de s'assurer d'un taux d'occupation convenable mais aussi de combattre la baisse des prix.

Selon les estimations, 25 à 35 % du volume total des réservations sont générés sur Internet dans le monde, ce qui présente un fort potentiel de développement. Outre leurs propres sites Web, les comparateurs de prix, les portails de tourisme multi-produits, les hôteliers positionnent de plus en plus leurs offres sur des sites entièrement dédiés à la réservation d'hôtels sur Internet. Ceux-ci diffusent les offres de milliers ou centaines d'hôtels de par le monde, indiquant en temps réel aux internautes les disponibilités de chaque établissement, qu'il s'agisse d'hôtels indépendants ou de grandes chaînes.

Dans ce contexte, la société G-fit, établissement d'accueil, nous a confié le développement d'une application GDS qui consiste en un système centralisé de réservation en ligne pour les hôtels.

Pour ce faire, on a choisi de réaliser ce projet en 3 chapitres. Le premier chapitre présentera le lieu du stage ainsi que le projet à réaliser. Le deuxième sera consacré à l'analyse du projet et sa conception générale ainsi que la présentation des outils de développement. La réalisation de l'application sera développée dans le troisième chapitre.

Chapitre 1

Présentation du cadre général du projet

Introduction

Ce chapitre présente le contexte général dans lequel s'est déroulé le projet. Il présente d'abord, G-fit Maroc, l'organisme d'accueil. Il décrit ensuite le contexte général ainsi que les principaux objectifs de l'application GDS.

1. Présentation générale de l'entreprise

G-fit France est une SSII (Société de service en ingénierie informatique) créée en 1998, dont les principales activités résident dans l'informatique de gestion au compte des grands clients français ou internationaux, l'assistance dans la maîtrise d'ouvrage et la maîtrise d'œuvre, la recherche et le développement (R&D) et le consulting pour recrutement.

G-fit permet de garantir à ses clients des prestations de qualité par une approche personnalisée, et un tissage de liens de partenariat et de confiance.

Le projet Off-shore établis par G-fit France, concrétisant le développement de la firme, a vu le jour avec la création de la filiale G-fit Maroc en Juin 2008 avec un capital de 500.000DHS. Ce projet vise à délocaliser les services de développement, afin de rendre l'entreprise plus concurrente que ça soit sur l'échelle nationale ou internationale.

1.1.Fiche d'identité

Année	Activité
2000 :	Démarrage de l'activité cabinet conseil en recrutement.
2002 :	Démarrage de la R&D avec KOALA® (Editeur de flux).
2003	Poursuite de la R&D avec FEUILLE@FEUILLE®
2005	Mise en ligne du site www.agorastore.fr .
2006	CA de 7 M€.
2007	160 collaborateurs et CA de 10 M€.
2008	180 collaborateurs et CA de 13 M€. G-Fit Maroc

Tableau 1: Fiche d'identité

1.2.Les métiers

Les principaux secteurs d'activités de G-fit se focalisent essentiellement sur :

- La Régie,
- Le forfait,
- La Tierce Maintenance Applicative,
- Cabinet Conseil en Recrutement.

Le tableau suivant résume les métiers G-Fit

Métiers	Description
La régie	<ul style="list-style-type: none"> Assistance technique Assistance à Maîtrise d'Œuvre Assistance à Maîtrise d'Ouvrage Direction, coordination et planification de projets
Le forfait	<ul style="list-style-type: none"> Engagement de résultats Respect des coûts et des délais Démarche et organisation au service de la qualité Externalisation ou non du projet Interlocuteur dédié : le Manager de Projet
La Tierce Maintenance Applicative	<ul style="list-style-type: none"> Engagement de moyens et de résultats Prise en charge de la maintenance corrective, évolutive et préventive Mise en œuvre d'un Plan Qualité Projet Externalisation ou non du projet
Cabinet Conseil en Recrutement	<ul style="list-style-type: none"> Pré-embauche ou Embauche directe chez le client Validation de la personnalité et des compétences techniques et/ou métiers Mise en place d'une convention de recrutement Engagement de résultat et suivi pendant la période d'intégration du personnel recruté

Tableau 2 : tableau des métiers

1.3. Les objectifs

Les objectifs principaux de G-fit est de parvenir d'ici fin 2010, à augmenter le chiffre d'affaire de 12% avec les produits recherches et développements, ainsi qu'à augmenter le personnel que ce soit en France ou au Maroc à 191 acteurs.

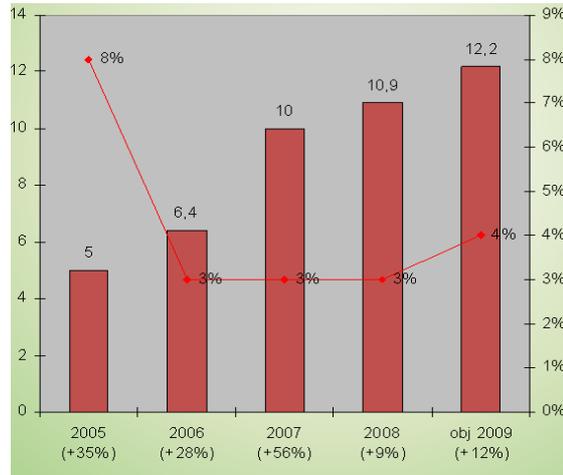


Figure 1 : Les objectifs

1.4. Les références

G-fit est référencé chez des grands clients, la figure suivante montre quelques comptes auxquels G-fit a travaillé pour, que ce soit pour des projets TMA, forfaits, ou régis.



Figure 2 : Les références clients G-Fit

2. Cadre général du projet

Dans le cadre d'un projet interne, G-fit Maroc a décidé de monter une application web pour la gestion centralisée de réservation pour les hôtels.

L'application consiste à :

- Permettre aux hôtels de publier leur produit (chambres) sur le net,
- Mise à la disposition des partenaires d'une liste des réservations effectuées,
- Permettre aux partenaires de modifier leur coordonnées, d'ajouter des chambres, de supprimer, etc.,
- Le visiteur pourra :
 - chercher un hôtel selon des critères multiples,
 - réserver une chambre en ligne selon la disponibilité,
 - rédiger ses remarques sur un hôtel qu'il a visité,
- Le partenaire a un accès pour visualiser les réservations des clients.

Chapitre 2

Analyse et Conception du projet

Introduction

Ce chapitre est composé de trois parties, la première partie présente les contraintes et les outils de développements utilisés. Dans une deuxième partie, on va enchaîner par une étude des spécifications des besoins suivie de la modélisation UML. La troisième partie abordera la conception du projet.

1. Contraintes principales de développement

En ce qui concerne les outils du travail, le langage UML a été choisi pour la conception préliminaire du projet, l'outil utilisé est TopCased. Pour le développement, la société G-fit nous a recommandé d'utiliser le Framework JEE et l'IDE Spring Tools Suite.

1.1. Outils et Langages de développement

1.1.1. Framework JEE

Pourquoi des Framework ? Simplement pour produire des applications professionnelles et sécurisées, pour faciliter les développements et leur fournir un contexte. Ce qui va donc faciliter le travail, accroître la productivité et la maintenabilité de l'application tout en permettant la capitalisation et la réutilisation de composants.

Java Enterprise Edition est une norme proposée par la société Sun, portée par un consortium de sociétés internationales, visant à définir un standard de développement d'applications d'entreprises multi-niveaux, basées sur des composants. On parle généralement de «plate-forme JEE» pour désigner l'ensemble constitué des services (API) offerts et de l'infrastructure d'exécution. JEE comprend notamment :

Les spécifications du **serveur d'application**, c'est-à-dire de l'environnement d'exécution : JEE définit finement les rôles et les interfaces pour les applications ainsi que l'environnement dans lequel elles seront exécutées. Ces recommandations permettent ainsi à des entreprises tierces de développer des serveurs d'application conformes aux spécifications ainsi définies, sans avoir à redévelopper les principaux services.

Des services, au travers d'**API**, c'est-à-dire des extensions Java indépendantes permettant d'offrir en standard un certain nombre de fonctionnalités. *Sun* fournit une implémentation minimale de ces API appelée **JEE SDK** (JEE Software Development Kit).

Dans la mesure où JEE s'appuie entièrement sur le Java, il bénéficie des avantages et inconvénients de ce langage, en particulier une bonne portabilité et une maintenabilité du code.

De plus, l'architecture JEE repose sur des composants distincts, interchangeables et distribués, ce qui signifie notamment :

- qu'il est simple d'étendre l'architecture ;
- qu'un système reposant sur JEE peut posséder des mécanismes de haute-disponibilité, afin de garantir une bonne qualité de service ;
- que la maintenabilité des applications est facilitée.

1.1.2. Framework Spring

L'utilisation du Framework Spring nous permet de garantir le respect strict de la séparation des couches applicatives. En effet, grâce à son moteur d'inversion de contrôle et ses mécanismes d'injections de dépendances, Spring permet aux couches supérieures de ne connaître et de n'utiliser que les interfaces publiques des services qu'elles souhaitent utiliser. C'est le Framework lui-même qui se chargera de faire correspondre Interfaces et Implémentations grâce à une configuration XML ou à des annotations Java5 depuis Spring 2.5. En outre, le Framework Spring met à disposition un certain nombre d'outils permettant de résoudre les problématiques transverses (sécurité, transactions, etc.) de manière élégante et non intrusive (mécanismes AOP).

1.1.3. Spring MVC [Spring MVC par l'exemple, 06]

Le Framework Spring apporte pour la construction des architectures 3tier. Son concept d'inversion de contrôle (IOC) permet notamment de construire une architecture 3tier avec des couches indépendantes les unes des autres.

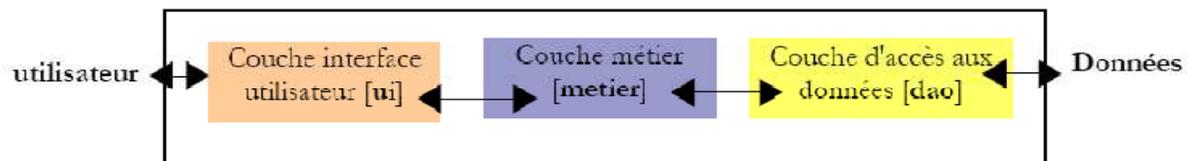


Figure 3 : Spring MVC

- ✚ La couche [**dao**] s'occupe de l'accès aux données, le plus souvent des données persistantes au sein d'un SGBD. Mais cela peut être aussi des données qui proviennent de capteurs, du réseau, ...
- ✚ La couche [**métier**] implémente les algorithmes " métier " de l'application. Cette couche est indépendante de toute forme d'interface avec l'utilisateur. Ainsi elle doit être utilisable aussi bien avec une interface console, une interface web, une interface de client riche. Elle doit ainsi pouvoir être testée en-dehors de l'interface web et notamment avec une interface console. C'est généralement la couche la plus stable de l'architecture. Elle ne change pas si on change l'interface utilisateur ou la façon d'accéder aux données nécessaires au fonctionnement de l'application.
- ✚ La couche [**interface utilisateur**] qui est l'interface (graphique souvent) qui permet à l'utilisateur de piloter l'application et d'en recevoir des informations.

La communication va de la gauche vers la droite :

- ✚ L'utilisateur fait une demande à la couche [**interface utilisateur**].
- ✚ Cette demande est mise en forme par la couche [interface utilisateur] et transmise à la couche [**métier**].
- ✚ Si pour traiter cette demande, la couche [métier] a besoin des données, elle les demande à la couche [**dao**].
- ✚ Chaque couche interrogée rend sa réponse à la couche de gauche jusqu'à la réponse finale à l'utilisateur.

Au sein de l'architecture 3tier, l'architecture MVC peut être représentée comme suit :

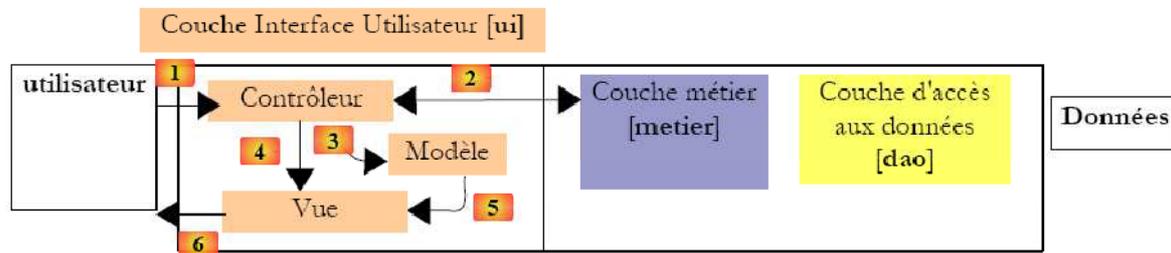


Figure 4 : Architecture MVC

Le traitement d'une demande d'un client se déroule selon les étapes suivantes :

1) Le client fait une demande au contrôleur. Celui-ci voit passer toutes les demandes des clients. C'est la porte d'entrée de l'application. C'est le C de MVC.

2) Le contrôleur C traite cette demande. Pour ce faire, il peut avoir besoin de l'aide de la couche métier. Une fois la demande du client traitée, celle-ci peut appeler diverses réponses. Un exemple classique est :

- ✚ Une page d'erreurs si la demande n'a pu être traitée correctement.
- ✚ Une page de confirmation sinon.

3) Le contrôleur choisit la réponse (= vue) à envoyer au client. Choisir la réponse à envoyer au client nécessite plusieurs étapes :

- ✚ Choisir l'objet qui va générer la réponse. C'est ce qu'on appelle la vue V, le V de MVC. Ce choix dépend en général du résultat de l'exécution de l'action demandée par l'utilisateur.
- ✚ Lui fournir les données dont il a besoin pour générer cette réponse. En effet, celle-ci contient le plus souvent des informations calculées par le contrôleur. Ces informations forment ce qu'on appelle le modèle M de la vue, le M de MVC.

L'étape 3 consiste donc en le choix d'une vue V et en la construction du modèle M nécessaire à celle-ci.

4) Le contrôleur C demande à la vue choisie de s'afficher. Il s'agit le plus souvent de faire exécuter une méthode particulière de la vue V chargée de générer la réponse au client. Dans ce document, nous appellerons vue, aussi bien l'objet qui génère la réponse au client que cette réponse elle-même. La littérature MVC n'est pas explicite sur ce point. Si c'est la réponse qui devait s'appeler vue, on pourrait appeler générateur de vue, l'objet qui génère cette réponse.

- 5) Le générateur de vue V utilise le modèle M préparé par le contrôleur C pour initialiser les parties dynamiques de la réponse qu'il doit envoyer au client.
- 6) La réponse est envoyée au client. La forme exacte de celle-ci dépend du générateur de vue. C'est peut être un flux HTML, PDF, Excel, etc.

1.1.4. MySQL

MySQL, le plus populaire des serveurs de bases de données SQL Open Source, est développé, distribué et supporté par MySQL AB. AB est une société commerciale, fondée par les développeurs de MySQL, qui développent leur activité en fournissant des services autour de MySQL.

MySQL est un serveur de bases de données relationnelles qui stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le serveur MySQL est un système client / serveur qui est constitué d'un serveur SQL multithread qui supporte différentes interfaces, clients, bibliothèques et outils d'administration, ainsi qu'une large gamme de pilotes pour différents langages (API).

1.1.5. Hibernate/JPA [Persistance Java 5 par la pratique, 07]

L'API JPA est récente. Elle n'a été disponible qu'à partir du JDK 1.5. La couche JPA a sa place dans une architecture multicouche. Considérons l'architecture utilisée dans l'application, celle à trois couches :

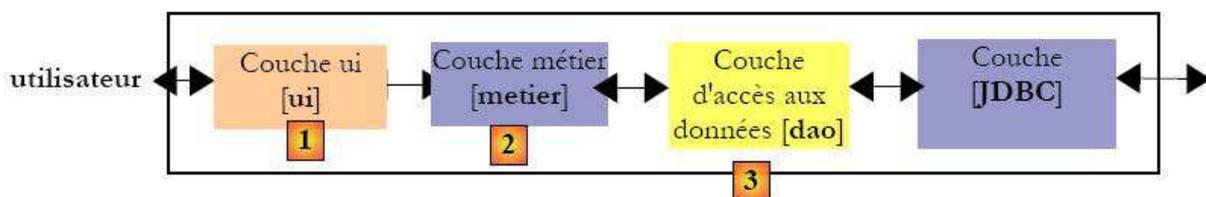


Figure 5 : multicouche JPA

- ✚ La couche [1], appelée ici [ui] est la couche qui dialogue avec l'utilisateur, via une interface web. Elle a pour rôle de fournir des données provenant de l'utilisateur à la couche [2] ou bien de présenter à l'utilisateur des données fournies par la couche [2].

- ✚ La couche [2], appelée ici **[métier]** est la couche qui applique les règles dites métier, c.à.d. la logique spécifique de l'application, sans se préoccuper de savoir d'où viennent les données qu'on lui donne, ni où vont les résultats qu'elle produit.
- ✚ La couche [3], appelée ici **[dao]** est la couche qui fournit à la couche [2] des données préenregistrées (fichiers, bases de données, etc.) et qui enregistre certains des résultats fournis par la couche [2].
- ✚ La couche [JDBC] est la couche standard utilisée en Java pour accéder à des bases de données. C'est ce qu'on appelle habituellement le pilote JDBC du SGBD.

De multiples efforts ont été faits pour faciliter l'écriture des ces différentes couches par les développeurs. Parmi ceux-ci, JPA vise à faciliter l'écriture de la couche [dao], celle qui gère les données dites persistantes, d'où le nom de l'API. Une solution qui a percé ces dernières années dans ce domaine, est celle d'Hibernate.

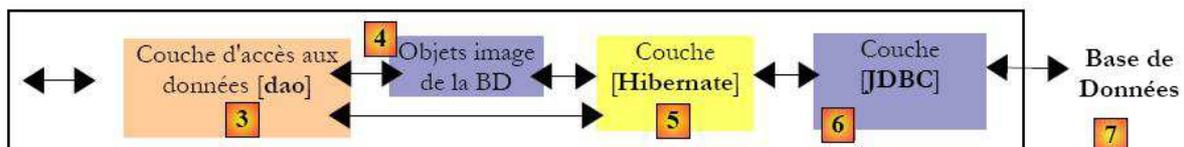


Figure 6 : Hibernate dans JPA

La couche **[Hibernate]** vient se placer entre la couche **[dao]** écrite par le développeur et la couche **[JDBC]**. Hibernate est un ORM, un outil qui fait le pont entre le monde relationnel des bases de données et celui des objets manipulés par Java. Le développeur de la couche **[dao]** ne voit plus la couche **[JDBC]** ni les tables de la base de données dont il veut exploiter le contenu. Il ne voit que l'image objet de la base de données, image objet fournie par la couche **[Hibernate]**. Le pont entre les tables de la base de données et les objets manipulés par la couche [dao] est fait principalement par des annotations Java dans le code, technique disponible seulement depuis le JDK 1.5.

La couche **[Hibernate]** est une couche d'abstraction qui se veut la plus transparente possible. L'idéal visé est que le développeur de la couche **[dao]** puisse ignorer totalement qu'il travaille avec une base de données. C'est envisageable si ce n'est pas lui qui écrit la configuration qui fait le pont entre le monde relationnel et le monde objet.

1.1.6. Web Flow [Spring Web Flow 2 Web Development, 09]

Spring Web Flow est un Framework qui peut être utilisé sans Spring MVC, l'intégration avec ce dernier est plus homogène.

- ✓ Principe de SWF.

Le web flow représente une conversation avec un utilisateur unique. Il peut être représenté à l'aide d'un diagramme d'état. Voici un exemple :

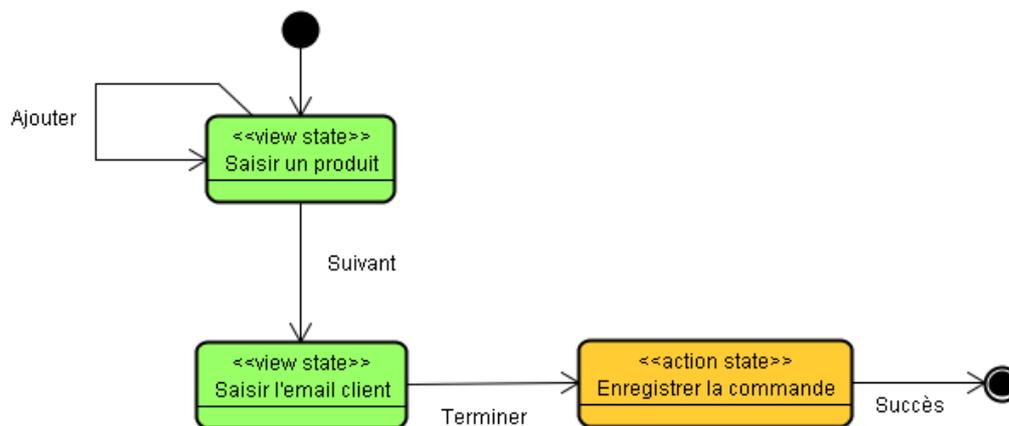


Figure 7 : principe de SWF

Autour de ce Framework, il ya trois mots importants, qui doivent être définies :

- Flow

Le Flow est un processus composé de certaines vues où les données sont stockées dans une conversation, Du point de vue plus technique, un Flow en capsule une séquence réutilisable d'étapes qui peuvent être exécutées dans des contextes différents. Tout d'abord, le Flow décrit les ordres et les exigences quand une vue est publiée, En outre, les actions peuvent être exécutées, entre les exécutions du flow. D'autre part, la conversation contient les données de l'utilisateur.

- Vue

Une Vue au Spring Web Flow est une page unique qui visualise l'information.

- Conversation

Dans les applications web traditionnelles, on a plusieurs scopes : requête, session etc. plusieurs cas d'utilisation dans une application web se composent de plus d'une seule page. Par conséquent, vous avez besoin de plus d'une requête pour stocker les données. Une requête

n'est souvent pas suffisante, et une session est de trop, Dans ce cas, Spring Web Flow introduit le concept d'une conversation qui utilise des différents scopes prédéfinies, définissant les scopes les plus utilisés :

- FlowScope : au début du Flow, le Flowscope est initialisé et détruit une fois le flow se termine, c'est la meilleur façon pour stocker les données qui doivent être présente tout au long du flow.
- ViewScope : le View scope initialise le scope avec les données en entré de la vue, une fois sortie de la vue le scope est détruit, Il est important de savoir que le scope est uniquement accessible à partir de la vue concernée.

✓ **Notion de View State**

Comme son nom le laisse penser, un View state correspond à une page que l'on souhaite afficher. Quand on entre dans un View state, SWF affiche la page correspondante à l'état. Le web flow est alors en pause. Après un certain temps, l'utilisateur génère un événement pour reprendre l'exécution du web flow, un View state est défini comme l'état de démarrage du web flow, l'état de démarrage est le premier état trouvé dans la liste.

Exemple de View-state :

```
<view-state id="saisirProduit" view="/saisirproduit.jsp">
  <transition on="suivant" to="saisirEmail" />
  <transition on="ajouter" to="saisirProduit" />
</view-state>
```

Figure 8 : exemple de View-state

Ce tableau décrit les détails de la figure :

Mot	description
<View-state><View-state>	définit le début et la fin d'une vue.
id	définit un nom particulier pour la vue
View	Définit la vue correspondante
transition	Définit la vue suivante après l'envoi d'une action par l'utilisateur
On	Définit l'action envoyée.
To	Définit la vue suivante après l'envoi d'une action.

Tableau 3 : tableau de View-state

1.1.7. Framework JSF

Java Server Faces est un Framework d'interface utilisateur pour les applications web, basé sur les technologies JSP et Servlets, le but de JSF est d'accroître la productivité des développeurs dans le développement des interfaces utilisateur tout en facilitant leur maintenance

JSF est le fruit de la communauté Java via le JCP. Le développement de JSF suit donc la même procédure que les autres technologies Java comme JSP, Servlets, EJB, etc. Cette procédure consiste pour une version donnée de la technologie, en une phase de spécification puis une phase d'implémentation quasiment parallèle. JSF fait partie de JEE 1.4.

Pourquoi utiliser JSF? Le support de JSF par les éditeurs JEE est obligatoire. Actuellement, les plus grands éditeurs Java annoncent ou proposent une intégration de JSF dans leurs IDE.

JSF permet :

- Une séparation nette entre la couche de présentation et les autres couches,
- Le mapping HTML/Objet,
- Un modèle riche de composants graphiques réutilisables,
- Une gestion de l'état de l'interface entre les différentes requêtes,
- une liaison simple entre les actions côté client de l'utilisateur et le code Java correspondant côté serveur.

1.2. Structuration du projet java avec Maven

Grâce à Maven, on ne définit plus chacune des opérations à réaliser mais la structure même du projet à travers un fichier unique (le pom.xml - Project Object Model). A partir de cette définition centralisée, le système Maven déduira seul l'ensemble des tâches qu'il peut effectuer sur le projet et la manière de les mettre en œuvre. L'utilisation de Maven permet de se passer d'écritures de scripts spécifiques.

Maven propose une configuration par défaut très complète qui permet une standardisation de la structure des projets Java. Au besoin, Maven permet de surcharger les configurations pour s'adapter à plus de situations. De cette manière on disposera d'un processus standardisé et extensible favorisant la capitalisation sur différents projets ainsi que l'efficacité quotidienne des équipes de développement.

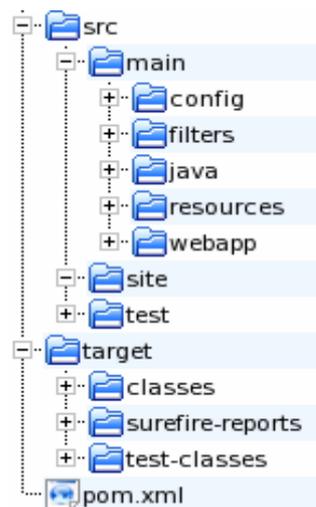


Figure 9 : structure de Maven

1.3.Modélisation UML

1.3.1. Présentation UML

UML n'est pas une méthode (i.e. une description normative des étapes de la modélisation), ses auteurs ont en effet estimé qu'il n'était pas opportun de définir une méthode en raison de la diversité des cas particuliers. Ils ont préféré se borner à définir un langage graphique qui permet de représenter, de communiquer les divers aspects d'un système d'information (aux graphiques sont bien sûr associés des textes qui expliquent leur contenu). UML est donc un métalangage car il fournit les éléments permettant de construire le modèle qui, lui, sera le langage du projet.

Il est impossible de donner une représentation graphique complète d'un logiciel, ou de tout autre système complexe, de même qu'il est impossible de représenter entièrement une statue (à trois dimensions) par des photographies (à deux dimensions). Mais il est possible de donner sur un tel système des vues partielles, analogues chacune à une photographie d'une statue, et dont la juxtaposition donnera une idée utilisable en pratique sans risque d'erreur grave.

Les points forts d'UML

UML est un langage formel et normalisé

- Gain de précision
- Gage de stabilité

- Encourage l'utilisation d'outils

UML est un support de communication performant

- Il cadre l'analyse.
- Il facilite la compréhension de représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel.

Les points faibles d'UML

La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation et l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue

1.3.2. Présentation de l'outil TopCased

TopCased est un acronyme pour Toolkit in Open Source for Critical Applications & Systems Development (boîte à outils open source pour le développement d'applications critiques et de systèmes).

TopCased est un logiciel d'ingénierie assisté par ordinateur. Il contient un IDE basé sur le Framework de la plateforme de développement Eclipse, à laquelle il ajoute des fonctionnalités essentiellement liées à la mise en œuvre de la première branche du cycle en V pour l'ingénierie du logiciel, du matériel ou de systèmes mixtes logiciel/matériel.

S'appuyant principalement sur des langages standardisés pour la modélisation du logiciel (UML, SysML, AADL...), TopCased travaille avec des fichiers XMI. Tous ses standards sont implémentés dans leurs dernières versions stables, soit directement par le projet TopCased, soit par les modules de la dernière version stable de la plate-forme Eclipse. TopCased UML est ainsi le modèleur UML le plus complet parmi les solutions gratuites, et le plus respectueux des standards actuels.

1.3.3. Présentation d'Apache Tomcat.

Apache Tomcat est un conteneur libre de Servlets Java EE. Issu du projet Jakarta, Tomcat est désormais un projet principal de la fondation Apache. Tomcat implémente les spécifications des Servlets et des JSP de Sun Microsystems. Il inclut des outils pour la configuration et la gestion, mais peut également être configuré en éditant des fichiers de

configuration XML. Comme Tomcat inclut un serveur HTTP interne, il est aussi considéré comme un serveur HTTP (web). Tomcat peut être utilisé en autonomie avec son propre serveur web, ou en collaboration avec d'autres comme IIS par exemple.

Le projet Tomcat a été lancé comme implémentation de référence des Servlets par James Duncan Davidson, architecte logiciel chez Sun. Il a contribué à rendre le projet libre et a joué un rôle majeur dans sa donation par Sun à la fondation Apache. Le projet Tomcat fait partie d'un collectif Open Source connu sous le nom de Jakarta.

Il est constitué de composants :

- **Catalina** est le container Servlets, et implémente les spécifications de Sun pour les Servlets et les JSP,
- **Coyote** est le connecteur HTTP: il écoute le trafic entrant, dirige les requêtes au moteur de Tomcat, il traite la requête et renvoie la réponse au client,
- **Jasper** est le moteur JSP. Il envoie les fichiers JSP pour les compiler en tant que Servlets (gérable par Catalina). Il est capable de détecter les modifications des fichiers et de les recompiler à la volée.

Avantages de Tomcat :

- Tomcat est simple, beaucoup plus que les serveurs d'application Open Source,
- Il est donc plus simple d'administrer une instance Tomcat qu'un serveur d'applications complet.
- Il n'occupe que 2 ports sur la machine (8080 et 8009), alors que les autres en prennent une dizaine
 - 8080 : port propre de Tomcat
 - 8009 : port de communication entre Apache et Tomcat (protocole AJP13)

2. Etude et spécification des besoins

Dans cette partie, on va décrire en détail l'ensemble des besoins fonctionnels auxquels devra répondre le système, Les acteurs sont les clients, l'administrateur de l'application d'une part, et le partenaire de l'hôtel d'autre part.

2.1. Spécification des besoins

✓ Côté Client :

- **Réservation dans un hôtel**

Après avoir fait une recherche multicritère sur les hôtels fournis par l'application. Le Client peut réserver un séjour dans l'hôtel en passant par des formulaires administratifs.

- **Consulter une réservation**

Le client peut consulter sa réservation a l'aide d'un identifiant et d'un mot de passe fournis par l'application, il peut annuler, modifier la date de la réservation et même laisser son avis sur l'hôtel.

✓ Côté Partenaire :

- **Inscrire un hôtel**

Le partenaire peut soumettre une demande d'inscription de son produit (hôtel) dans l'application. Pour cela, il sera nécessaire de remplir les coordonnées prescrites du produit afin de publier son hôtel dans l'application.

- **Consulter son hôtel**

Le partenaire est disposé d'un login et mot passe, après authentification, il peut modifier les coordonnées, Lire les avis sur son hôtel, ajouter, modifier et supprimer une chambre.

✓ Côté Administrateur :

- **Gestion des demandes**

L'administrateur s'authentifie pour pouvoir consulter les demandes.il peut les confirmer, les annuler et supprimer carrément un hôtel.

2.2.Diagramme des cas d'utilisation.

Les cas d'utilisation, ayant un sens pour les acteurs. Elles permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique. c'est dans ce but qu'on a réaliser les diagrammes de

cas d'utilisation pour permettre de recueillir, d'analyser et d'organiser les besoins, et de recenser les grandes fonctionnalités d'un système.

Ces diagrammes définissent en détail les différentes interactions entre le système et les acteurs cités, ainsi chaque cas d'utilisation doit répondre à une fonctionnalité donnée.

- **Diagramme de cas d'utilisation du client.**

Cette figure présente les différents besoins cités sur un diagramme de cas d'utilisation. Les associations simples entre l'acteur (client) et ses cas d'utilisation, explique que l'acteur peut les accéder à tout moment, pour les associations « extend », elle traduit que les cas d'utilisation liés est optionnel pour l'acteur et ne peut les utiliser qu'au cours de l'utilisation du cas d'utilisation « consulter réservation » et « rechercher un hôtel ».

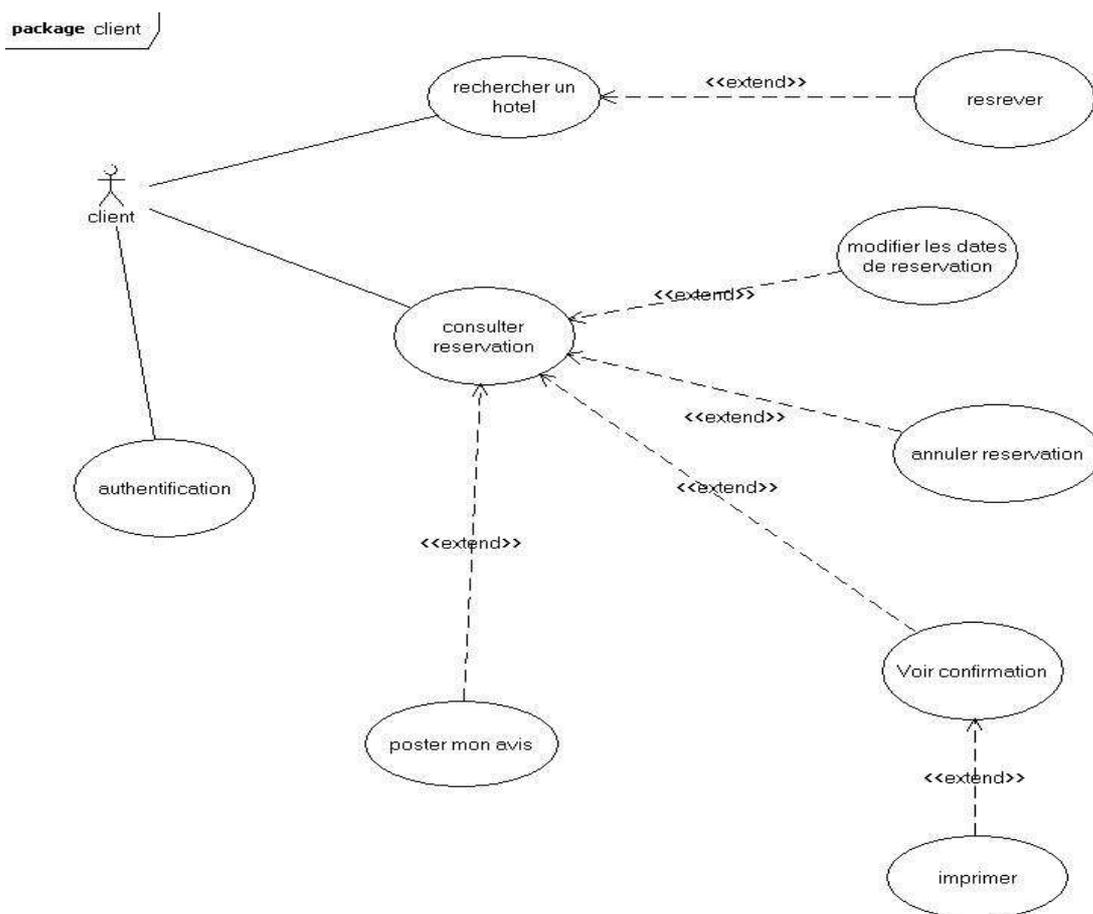


Figure 10 : Diagramme de cas d'utilisation pour le client

▪ **Diagramme de cas d'utilisation du partenaire.**

Après avoir définie les besoins du partenaire, ces derniers sont présentés sous forme d'un diagramme définissant les différents cas d'utilisation contenu dans des bulles, toutes les relations sont associatives explique qu'il ya une relation directe entre l'acteur et les services rendu par le système.

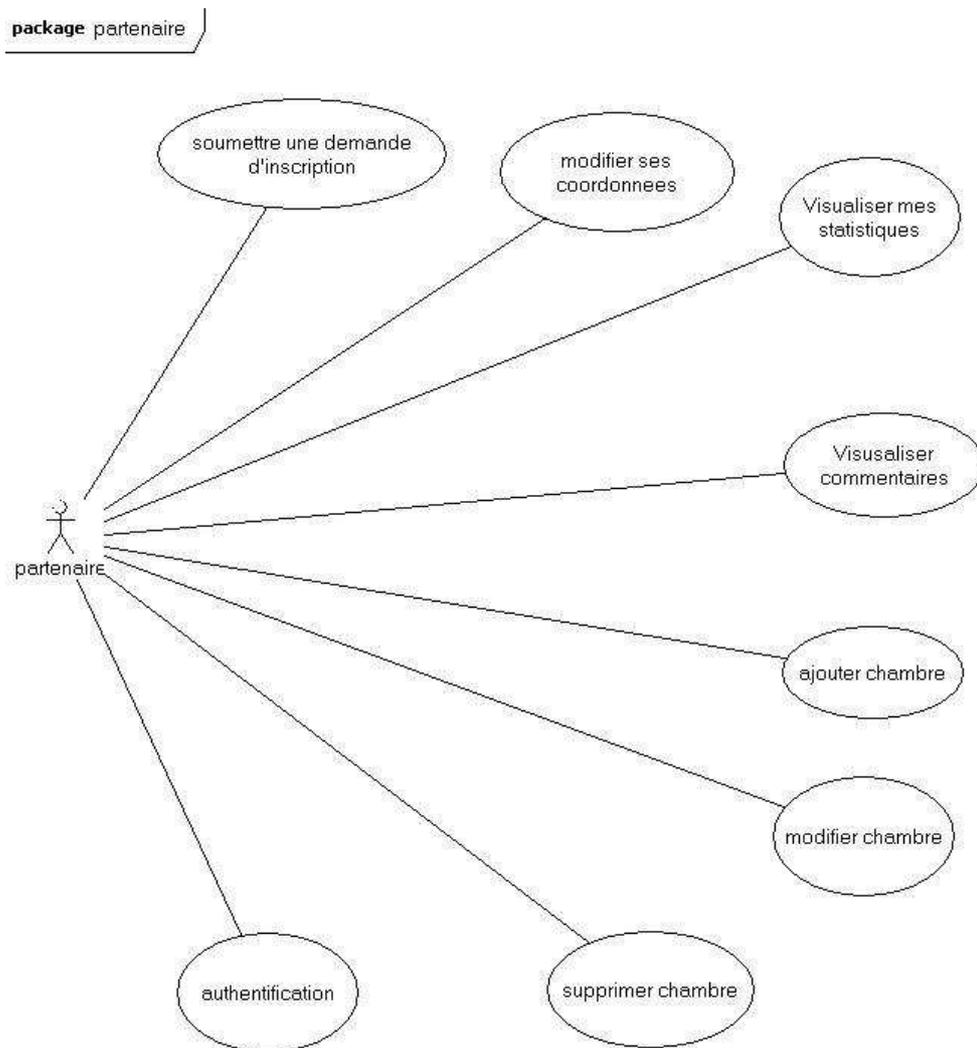


Figure 11 : Diagramme de cas d'utilisation pour le partenaire

▪ **Diagramme de cas d'utilisation d'administrateur.**

Le diagramme suivant présente les différents cas d'utilisation qu'on a pu extraire à partir des spécifications des besoins de cet acteur. Les cas d'utilisation présentés par des bulles indiquent les services rendus par le système à cet acteur.

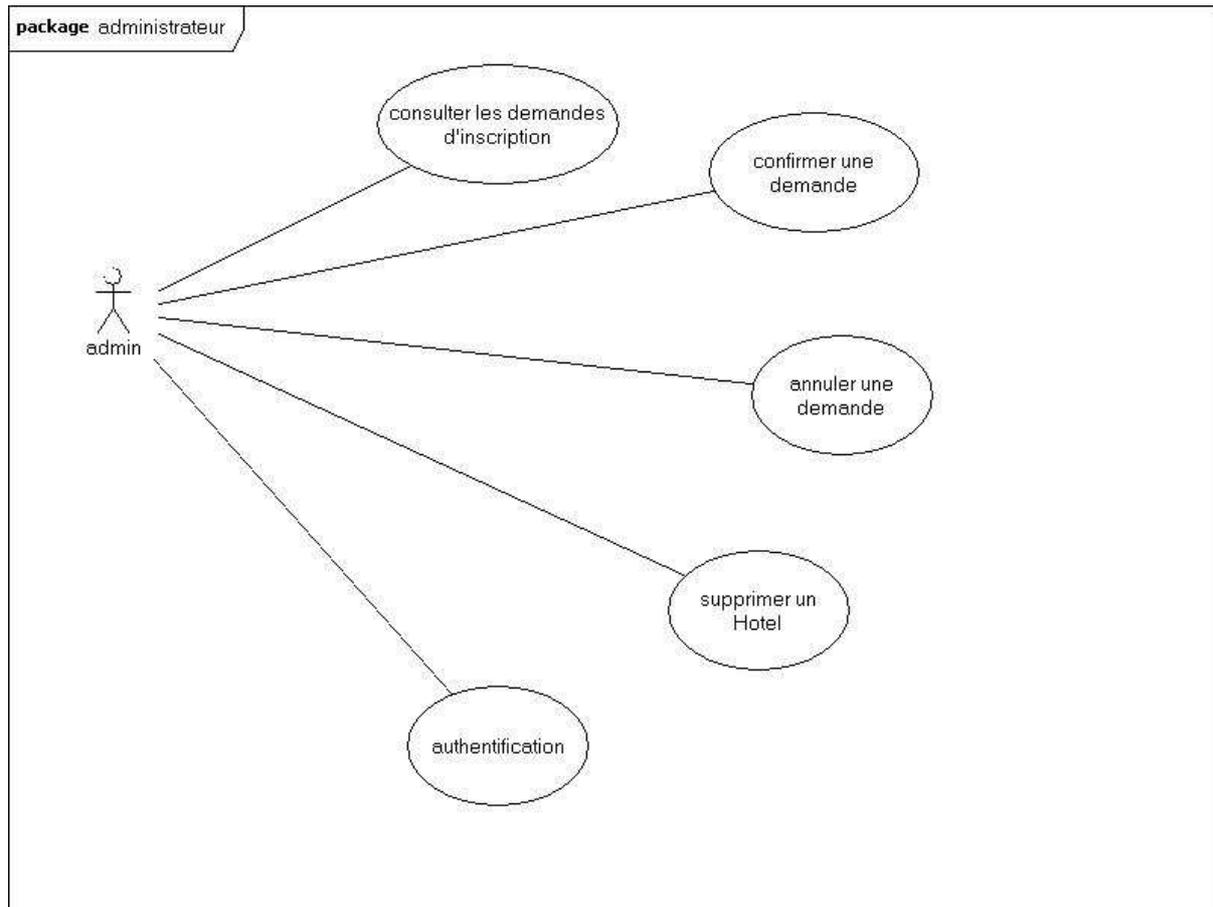


Figure 12 : Diagramme de cas d'utilisation pour l'administrateur

3. Etude et conception

3.1 Diagramme de déploiement

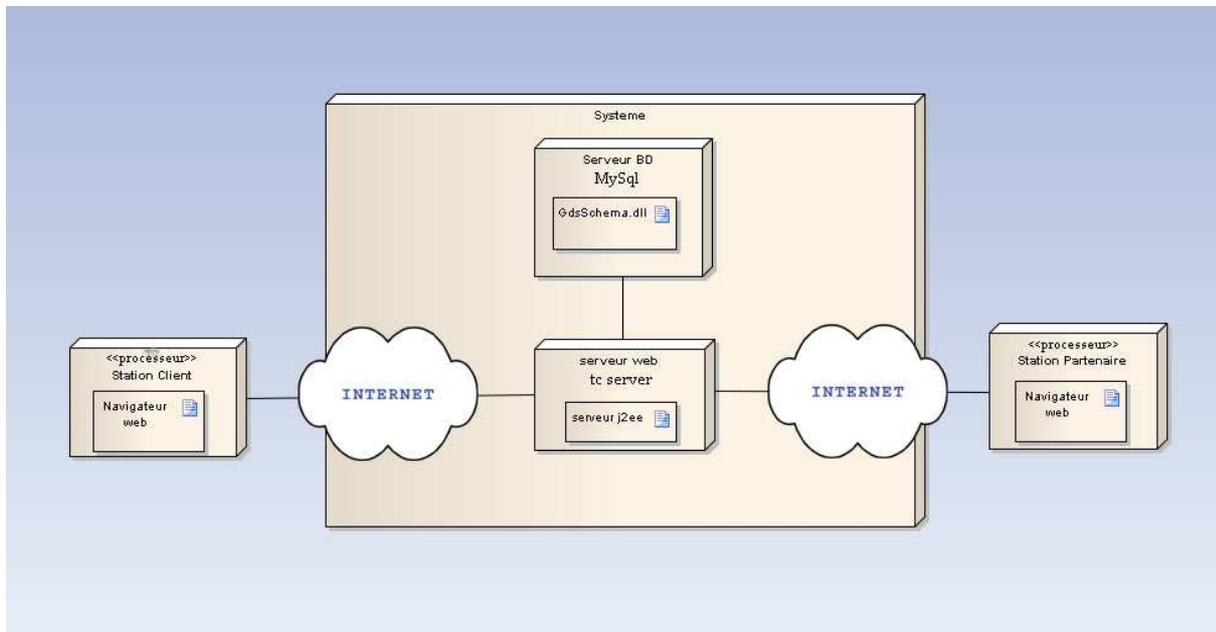


Figure 13 : Diagramme de déploiement

Ce diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels. Chaque ressource étant matérialisée par un nœud, on définit 3 nœud dans ce diagramme, nœud système et deux nœuds processeurs.

- Le nœud système englobe toute l'application, il contient le serveur de la base de données MySQL ainsi que le serveur web Tomcat.
- Le premier processeur s'agit de la station client un des acteurs communiquant avec le système, la liaison utilisé dans ce cas est une liaison à distance garanti par internet. Le processeur contient un artefact « navigateur web », l'interface web de l'application pour utiliser les services rendu par le système.
- Le deuxième processeur concerne le partenaire de l'hôtel, il se connecte au système à l'intermédiaire d'internet, le processeur est manié d'une interface web afin que le partenaire puisse bénéficier des services rendu par le système.

Ce diagramme nous a permis de mieux comprendre la disposition physique des matérielles qui composent le système ainsi que la relation entre le système et ses interlocuteurs.

3.2 Analyse du domaine

Considéré comme le plus important de la modélisation orientée objet, le diagramme de classes en montre la structure interne et fournit une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation. Le diagramme de classes modélise les concepts du domaine dans le cadre de l'implémentation de l'application.

Après l'analyse et la spécification des besoins, on a pu réaliser un diagramme de domaine illustrant les différentes tables qui interagissent dans le système afin de les persister dans la base de données.

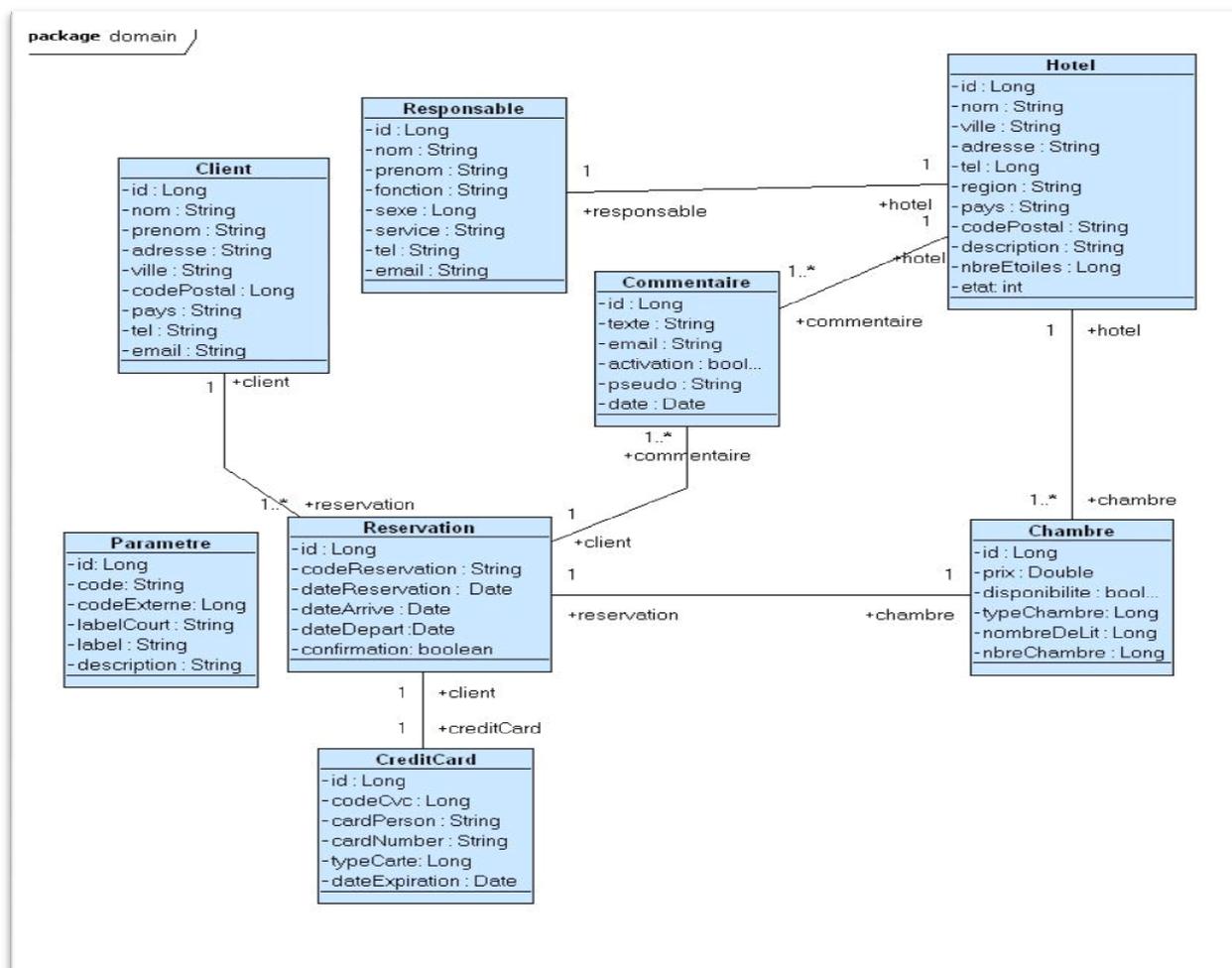


Figure 15 : Diagramme de classe

Les tableaux suivants définissent les caractéristiques de chaque classe ainsi que leurs types.

✓ Classe Hôtel

Cette classe définit tous les attributs nécessaires pour caractériser un hôtel.

<u>Attribut</u>	<u>Caractéristique</u>	<u>Type</u>
id	Identifiant de l'hôtel	Long
Nom	Nom de l'hôtel	String
Adresse	L'adresse de l'hôtel	String
ville	La ville de l'hôtel	String
téléphone	Le téléphone de l'hôtel	Long
rue	La rue où se situe l'hôtel	String
région	La région où se situe l'hôtel	String
pays	Le pays de l'hôtel	String
Code postal	Le code postal de la ville de l'hôtel	String
Nombre étoile	Le niveau de service de l'hôtel par étoile	Int
description	Une description générale sur l'hôtel	String
état	Mentionne si l'hôtel est confirmé par l'administrateur	Int

Tableau 4 : Classe Hôtel

✓ Classe Chambre

Cette classe définit les caractéristiques de chambres pour un hôtel donné.

<u>Attribut</u>	<u>Caractéristique</u>	<u>Type</u>
id	Identifiant d'un type de chambre	Long
Prix	Prix d'un type de chambre	double
disponibilité	Disponibilité d'un type de chambre	booléen
type Chambre	Le nom d'un type de chambre	Long
nombre Lit	Le nombre de lit pour un type de chambre	Long
nbreChambre	Le nombre de chambre pour un type donné	Long

Tableau 5 : Classe Chambre

✓ Classe Responsable

Cette classe définit tous les attributs nécessaires pour caractériser un responsable d'hôtel.

<u>Attribut</u>	<u>Caractéristique</u>	<u>Type</u>
id	Identifiant du responsable	Long
Nom	Nom du responsable	String
Prénom	prénom du responsable	String
Fonction	La fonction du responsable	String
téléphone	Le téléphone du responsable	Long
Service	Le service où il appartient le responsable	String
Sexe	Le Sexe du responsable	String
Email	L'email du responsable	String

Tableau 6 : Classe Responsable

✓ Classe Client

Cette classe définit tous les attributs nécessaires pour caractériser un client.

<u>Attribut</u>	<u>Caractéristique</u>	<u>Type</u>
id	Identifiant du client	Long
Nom	Nom du client	String
Prénom	prénom du client	String
Adresse	L'adresse du client	String
Email	Email du client	String
Ville	La ville du client	String
Code postal	Code postal de la ville du client	Long
pays	Le pays du client	String
téléphone	Le téléphone personnel du client	String

Tableau 7 : Classe Client

✓ Classe commentaire

Cette classe présente les attributs nécessaires pour définir un commentaire.

<u>Attribut</u>	<u>Caractéristique</u>	<u>Type</u>
id	Identifiant d'un commentaire	Long
Text	Texte d'un commentaire	String
Email	Email du propriétaire du commentaire	String
Activation	L'état du commentaire	booléen
pseudo	Pseudo du propriétaire du commentaire	String
Date	Date de rédaction du commentaire	Date

Tableau 8 : Classe Commentaire

✓ Classe carte crédit

Cette classe présente les attributs nécessaires pour définir une carte crédit.

<u>Attribut</u>	<u>Caractéristique</u>	<u>Type</u>
id	Identifiant de la carte crédit	Long
Code CVC	Code CVC appartenant à la carte crédit	Long
cardPerson	Propriétaire de la carte crédit	String
cardNumber	Numéro de la carte de crédit	String
typeCarte	Type de la carte crédit	Long
dateExpiration	Date d'expiration de la carte crédit	Date

Tableau 9 : Classe Carte Crédit

✓ Classe Réservation

La classe réservation contient tous les attributs nécessaires pour définir une réservation.

<u>Attribut</u>	<u>Caractéristique</u>	<u>Type</u>
id	Identifiant de la réservation	Long
codeReservation	Code confidentiel de la réservation	String
dateReservation	La date de la réservation	Date
dateArrive	La date de l'arrivée à l'hôtel	String
dateDepart	La date de départ de l'hôtel	Long
Confirmation	L'état de confirmation de la réservation	Date

Tableau 10 : Classe Réservation

✓ Classe paramètre

Cette classe a été implémentée pour faire référence aux différents types de classe exemples (type de chambre, type de carte de crédit, etc.)

<u>Attribut</u>	<u>Caractéristique</u>	<u>Type</u>
id	Identifiant d'un paramètre	Long
Code	Ce code fait référence à la table correspondante	String
codeExterne	Identifiant d'un type de table	Long
labelCourt	Abréviation d'un type de table	String
Label	Nom d'un type de table	String
Description	Description d'un type de table	String

Tableau 11 : Classe paramètre

Chapitre 3

Réalisation de l'application

Introduction

Dans ce chapitre on va détailler les objectifs de l'application déjà cités accompagné de figures illustratives.

1. IHMs Client

1.1. Réserver une chambre dans un hôtel.

- Recherche multicritère.

Pour que le Client puisse réserver dans un hôtel, l'application lui offre un moteur de recherche multicritère, basé sur un champ destination (nom de l'hôtel, pays, ville, région), une date d'arrivée et une date de départ. Le champ destination doit contenir en moins deux motifs et évidemment la date d'arrivée doit être supérieure ou égale à la date de départ.

- Liste des hôtels disponibles.

Après avoir effectué la recherche. L'application liste une série d'hôtels disponibles selon les dates saisies et le champ destination.



The screenshot shows the 'Fit-Booking' application interface. At the top right, there are links for 'ADMINISTRATION' and 'MA RÉSERVATION'. Below the header, there is a navigation menu with 'ACCUEIL', 'A PROPOS DE FIT-BOOKING', 'REJOIGNEZ-NOUS', and 'CONTACT'. The main content area displays a table of available hotels.

Nom	Ville	Action
jnane palace	fes	Voir
karouiyne palace	casablanca	Voir

At the bottom left of the table, there is a 'Retour' link.

Figure 16 : Vue Hôtels

- Voir les informations d'un hôtel donné.

Pour visualiser un hôtel, on clique sur l'action Voir, on passe alors à une page avec une description détaillée et des photos de l'hôtel, plus un tableau de chambres disponibles entre les dates saisies, ce tableau contient les noms des types de chambres, nombres de lits, le nombre de chambres et les tarifs correspondants à une nuitée par type de chambre en dirham marocain.

ACCEUIL
A PROPOS DE FIT-BOOKING
REJOIGNEZ-NOUS
CONTACT



jnane palace - fes

Description:

Le Jnan Palace Fès vous accueille au cœur d'un grand parc, en plein cœur de Fès. Il dispose d'une piscine extérieure, de courts de tennis, d'une salle de remise en forme et d'un sauna. Des massages vous sont également proposés sur place. L'accès sans fil (Wi-Fi) à Internet et le parking sont deux services fournis gratuitement. L'hôtel propose également un service de navettes de et vers l'aéroport. Le Royal Golf de Fez, qui jouit d'un parcours de 18 trous, vous attend non loin de l'hôtel.

date depart :06/18/2010
date Arrivee :06/11/2010

Disponibilité Des chambres

Type chambre	Nombre Lit	Nombre de chambres	Prix	Reservation
Simple vue jardin	1	2 disponibles	85 MAD	RESERVER
Simple vue piscine	2	3 disponibles	95 MAD	RESERVER
Duplex vue jardin	1	4 disponibles	120 MAD	RESERVER

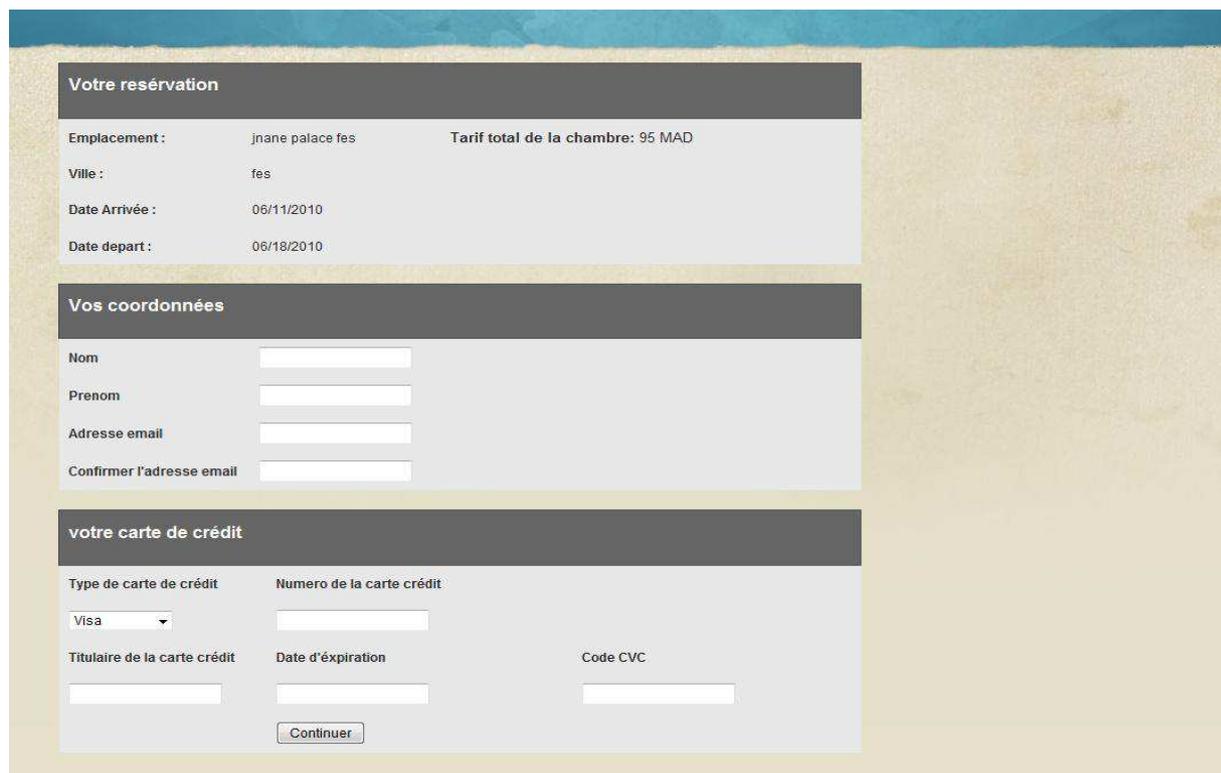
Figure 17 : Vue d'un Hôtel

- Saisie des coordonnées du Client.

En choisissant l'hôtel et la chambre désirée, le client est amené à remplir 2 formulaires :

- Un formulaire relatif aux informations personnelles du client (nom, prénom et adresse email)
- Un formulaire relatif à la carte de crédit Utilisé comme garantie pour l'hôtel.

Afin d'aider le client a bien remplir ses formulaires, les champs saisis doivent être valides et conformes aux normes JQUERY. On notera que la fonctionnalité du paiement sécurisé n'est pas encore traitée dans l'application.



The screenshot shows a web form with three main sections:

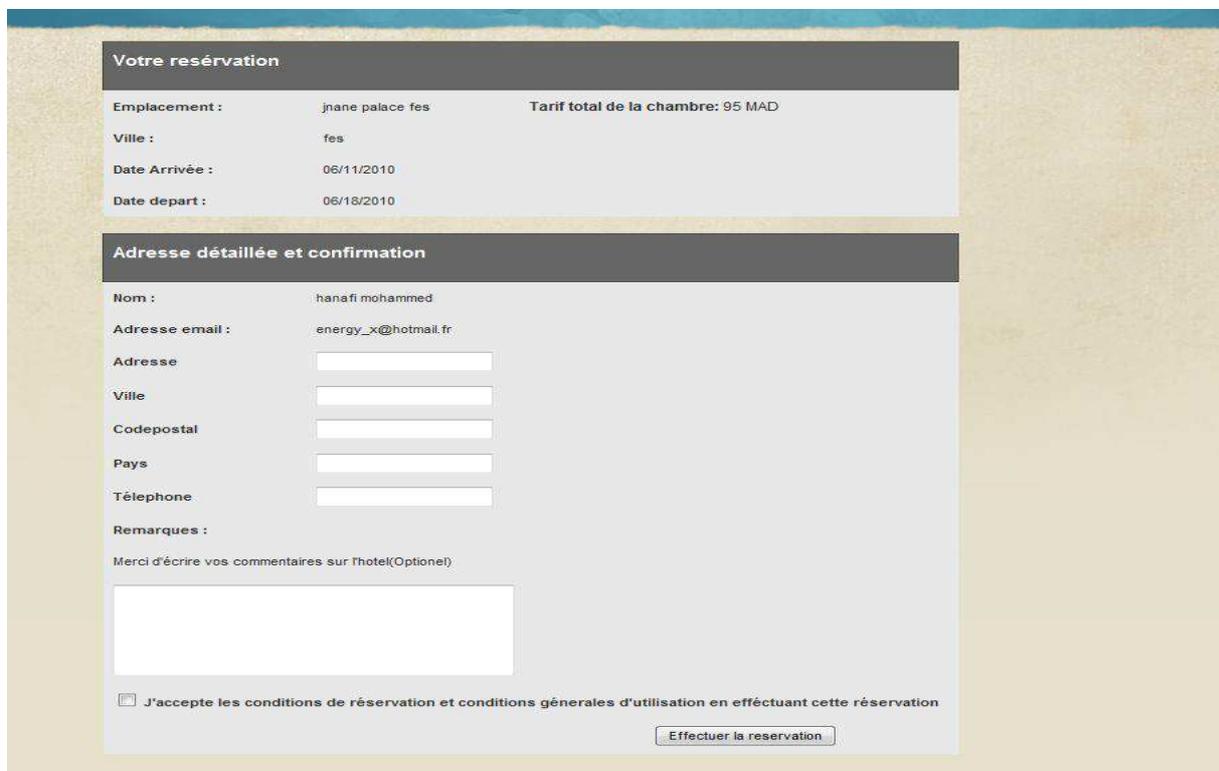
- Votre réservation**: A summary table with the following data:

Emplacement :	jnane palace fes	Tarif total de la chambre: 95 MAD
Ville :	fes	
Date Arrivée :	06/11/2010	
Date depart :	06/18/2010	
- Vos coordonnées**: A form with four input fields:
 - Nom
 - Prenom
 - Adresse email
 - Confirmer l'adresse email
- votre carte de crédit**: A form with several input fields:
 - Type de carte de crédit: A dropdown menu showing "Visa".
 - Numero de la carte crédit: A long input field.
 - Titulaire de la carte crédit: An input field.
 - Date d'expiration: An input field.
 - Code CVC: An input field.
 - A "Continuer" button at the bottom.

Figure 18 : Vue de saisie de coordonnées

- Saisie des coordonnées détaillées du client.

Après avoir accepté la carte de crédit par l'application. Le client peut maintenant saisir ses coordonnées personnelles pour permettre à l'hôtel de rester en contact avec lui. À la fin de cette étape, le client coche qu'il est d'accord sur les conditions générales et clique sur le bouton pour effectuer sa réservation au sein de cette hôtel.



Voire réservation

Emplacement : jnane palace fes Tarif total de la chambre: 95 MAD
 Ville : fes
 Date Arrivée : 06/11/2010
 Date depart : 06/18/2010

Adresse détaillée et confirmation

Nom : hanafi mohammed
 Adresse email : energy_x@hotmail.fr
 Adresse :
 Ville :
 Codepostal :
 Pays :
 Téléphone :
 Remarques :
 Merci d'écrire vos commentaires sur l'hotel(Optional)

J'accepte les conditions de réservation et conditions générales d'utilisation en effectuant cette réservation

Effectuer la reservation

Figure 19 : Vue de Saisie de coordonnées détaillées du client

- Page de confirmation de la réservation.

Cette page contient les différentes informations sur la réservation effectuée, Des informations sur la chambre, client, type de paiement, l'application fournit au client un code confidentiel utilisé comme mot de passe pour le Compte de la réservation. Tous ces détails sont envoyés par email au client.



Confirmation de réservation	
Merci Pour Votre réservation	
Email	energy_x@hotmail.fr
Code Confidentiel	MNOTmK
Cette réservation a été confirmée le	06/07/2010
Votre Nom	hanafi mohammed
Date Arrivée	06/11/2010
Date depart	06/18/2010
Tarif total de la chambre:	95 MAD

Une copie de Votre confirmation de réservation a été envoyée a l'adresse email suivante : **energy_x@hotmail.fr**.
Si vous n'avez pas reçu cette copie, merci de vérifier votre dossier Spam ou Courriel indésirable, elle y peut-etre bloquée.

pour Consulter, modifier ou annuler Votre reservation cliquez sur lien suivant :
<http://localhost:8081/gds/main/client?execution=e6s1>

Carte de crédit / Garantie / Information sur le paiement

Paiement

Vous avez confirmé et garanti votre réservation avec votre carte bancaire.

fit-Booking.com ne facture aucun montant sur votre carte de crédit.

Votre carte de crédit sert uniquement de garantie pour la réservation. L'hotel se réserve le droit d'effectuer une pré-autorisation de la carte de crédit avant la date d'arrivée.

Figure 20 : Vue de confirmation de réservation

1.2. Modifier ou annuler la réservation.

- Authentification.

Pour que le client puisse visualiser sa réservation, il doit se manifester de son adresse email et le code confidentiel fourni par l'application.



Compte De réservation

Email:

Mot de passe:

Figure 21 : Vue d'authentification du client

- Visualiser la Réservation.

Après authentification, le client visualise sa réservation, présentée sous forme d'un tableau contenant toutes les informations nécessaires. Cependant le client peut annuler sa réservation.

Votre Réservation
JNANE PALACE - FES

Type de chambre	Date Arrivée	Date Depart	Nom du Client	Tarif
Simple vue piscine	06/11/2010	06/18/2010	hanafi mohammed	95.0 MAD

[MODIFIER LES DATES](#)
[ANNULER LA RÉSERVATION](#)

Figure 22 : Vue de visualisation de la réservation

- Annuler la Réservation.

Le Client aura la possibilité d'annuler sa réservation. Tant que cette action est soumise le client ne peut plus activer sa réservation. et le Compte sera détruit par le partenaire directement.

Fit-Booking

ADMINISTRATION MA RÉSERVATION

ACCEUIL | A PROPOS DE FIT-BOOKING | REJOIGNEZ-NOUS | CONTACT

Cette Réservation a été annullée

ACCEUIL | FIT-BOOKING | REJOIGNEZ-NOUS | CONTACT

Fit-Booking

Menara Bureaux
10 Av. Aïlal Ben Abdellah Fès
Tel. : (+212) 535 622 430
Fax : (+212) 535 622 429

Figure 23 : Vue d'annulation de la réservation

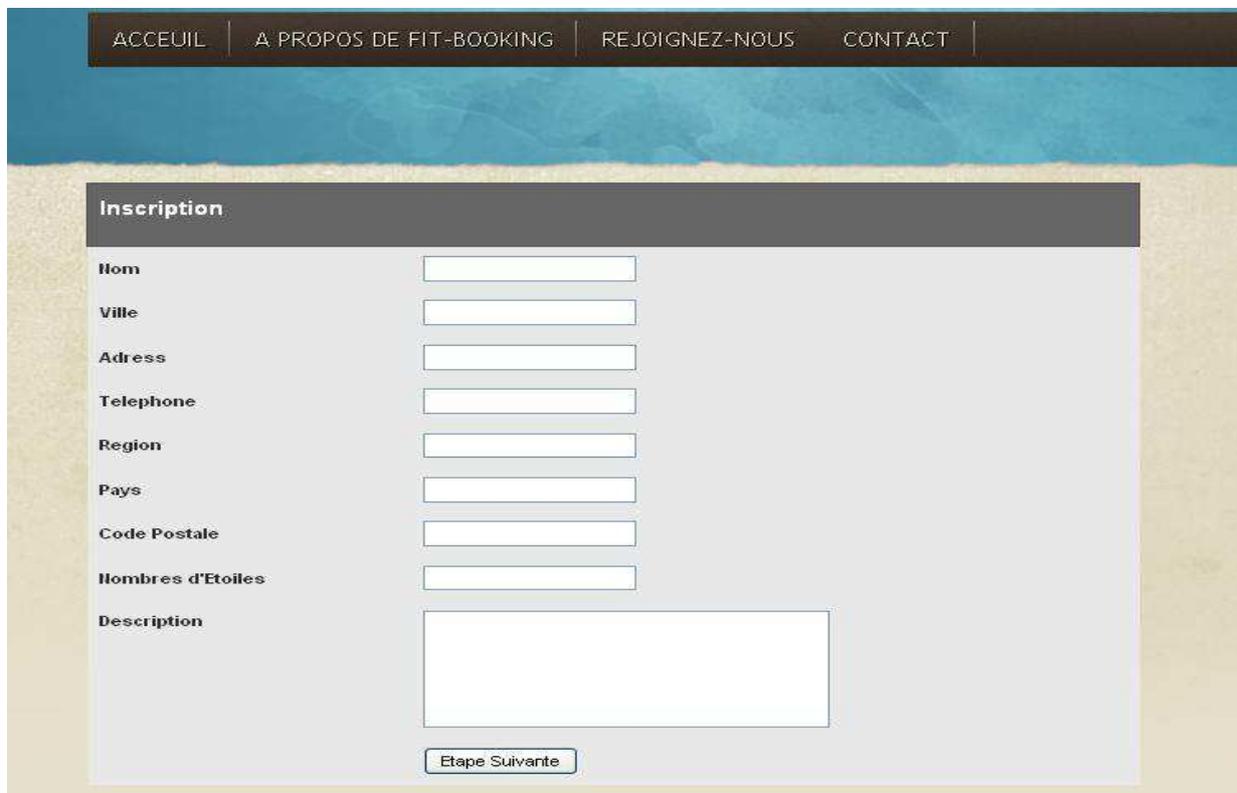
Dans le but de permettre au Client de réserver dans les hôtels, l'application Fit-Booking gère un côté partenaire afin d'inscrire les hôtels et publier ses différentes propriétés.

2. IHMS Partenaire

Inscription dans le site

- Saisie des coordonnées d'hôtel.

L'application permet aux partenaires d'hôtels d'inscrire leurs produits afin de les publier. Pour Cela, le partenaire sera amené à remplir un premier Formulaire par des informations sur l'hôtel (Nom, ville, pays, nombre d'étoiles, téléphone...).

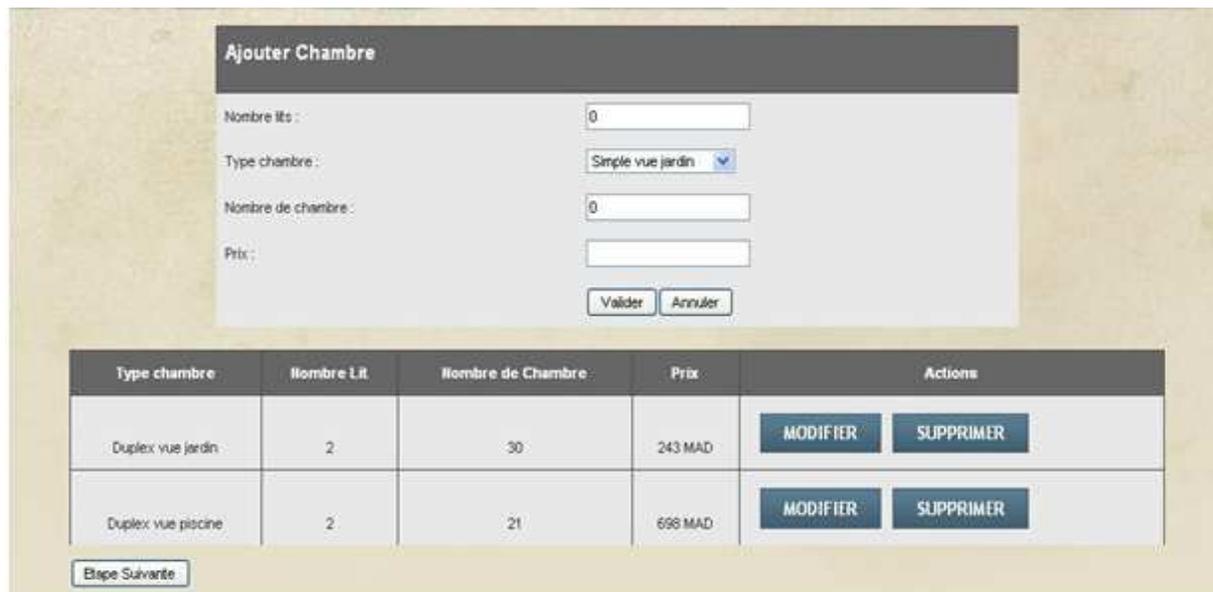


The screenshot shows a web interface for hotel registration. At the top, there is a navigation bar with links: ACCEUIL, A PROPOS DE FIT-BOOKING, REJOIGNEZ-NOUS, and CONTACT. Below this is a large blue decorative banner. The main content area is titled 'Inscription' and contains a form with the following fields: Nom, Ville, Adress, Telephone, Region, Pays, Code Postale, Nombres d'Etoiles, and Description. Each field has a corresponding input box. At the bottom of the form is a button labeled 'Etape Suivante'.

Figure 24 : Vue de Saisie des coordonnées d'hôtel

- Insertion des chambres

Après identification de l'hôtel, Le partenaire passe à un deuxième formulaire où il insère les spécifications des types de chambre, la vue propose aussi un tableau en dessous du formulaire mentionné contenant des actions permettant de modifier et de supprimer afin de faciliter la gestion.



Type chambre	Nombre Lit	Nombre de Chambre	Prix	Actions
Duplex vue jardin	2	30	243 MAD	<input type="button" value="MODIFIER"/> <input type="button" value="SUPPRIMER"/>
Duplex vue piscine	2	21	696 MAD	<input type="button" value="MODIFIER"/> <input type="button" value="SUPPRIMER"/>

Figure 25 : Vue d'insertion des chambres d'un Hôtel

- Formulaire du responsable

Afin de rester en contact avec l'application et ses nouveautés, le partenaire doit confier la tâche de gestion de son hôtel à un responsable. Pour cela, il remplit une fiche détaillée de son responsable (voir la figure 18).

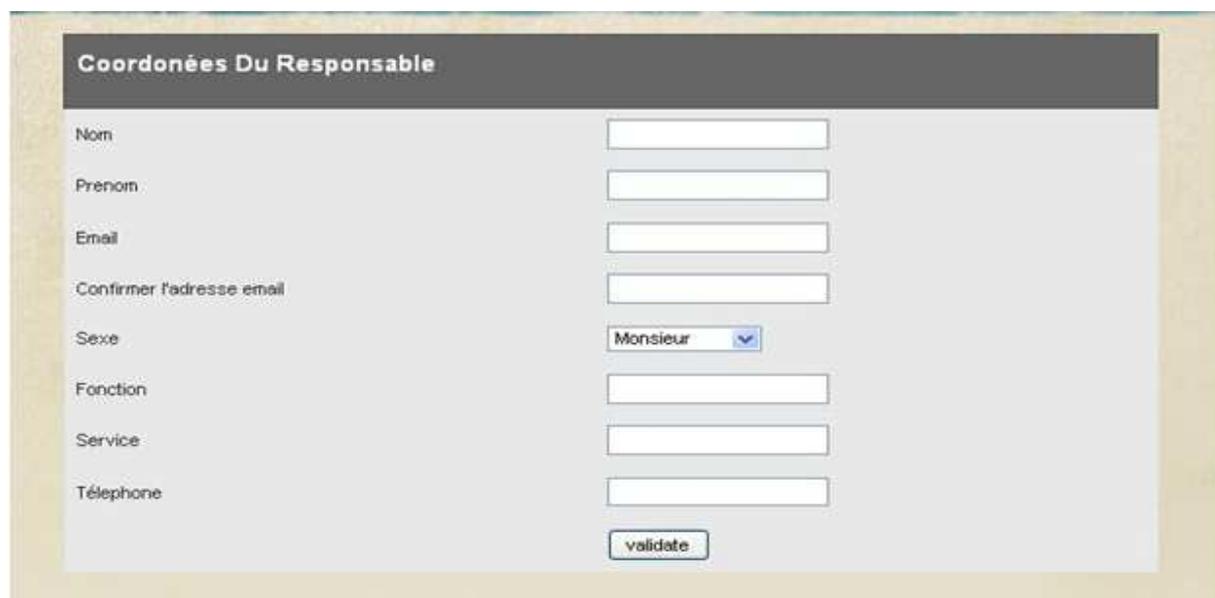


Figure 26 : Vue du formulaire du Responsable

- confirmation d'inscription

À la fin de l'inscription. Une vue de confirmation récapitule toutes les informations déjà saisie, un email d'attente de confirmation de réservation est envoyé a l'adresse email du responsable, dès que l'inscription sera confirmée par l'administrateur de l'application, le responsable de l'hôtel sera averti par un email d'acceptation de la réservation.



Figure 27 : Vue de confirmation d'inscription

3. IHMS Administrateur

Gestion des demandes

- Page d'authentification.

Cette partie concerne l'administrateur de l'application, pour se connecter à ses propres fonctionnalités, il doit s'authentifier grâce a son login et mot de passe.

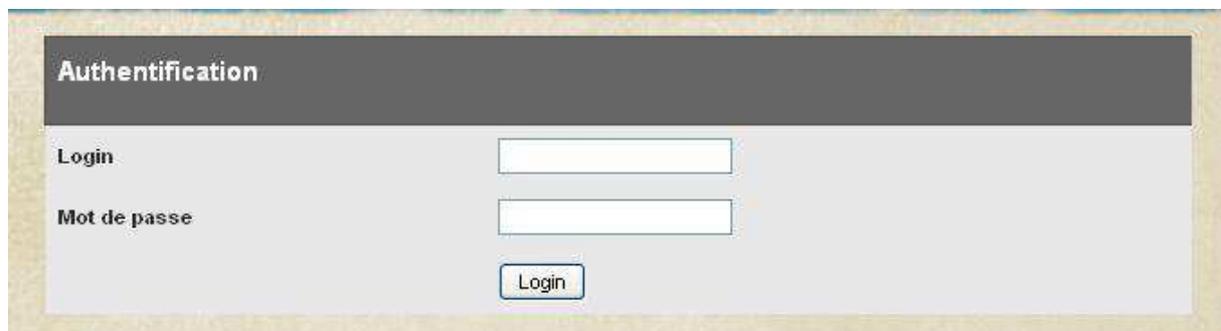


Figure 28 : vue d'authentification de l'administrateur

- Page des hôtels (nouveaux inscrits).

L'application met à la disposition d'administrateur après authentification une liste des hôtels non confirmés (nouveaux inscrits), avec la possibilité de visualiser, confirmer ou rejeter l'hôtel.



Nom	Ville	Pays	Nombre d'étoiles	Telephone	Actions
Royale Mirage	fes	maroc	5	05 35534356	VISUALISER CONFIRMER REJETER
Jnane Fes	fes	MAROC	2	05 35636783	VISUALISER CONFIRMER REJETER

Figure 29 : Vue des nouveaux inscrits

- Page de visualisation d'Hôtel

Afin qu'il puisse bénéficier des fonctionnalités de l'application, l'administrateur visualise les informations de l'hôtel pour les vérifier, l'administrateur garde toujours le choix de confirmer ou de rejeter l'inscription.



INFORMATIONS HOTEL	
Nom d' hotel	Royale Mirage
Pays et Ville	maroc fes
Adresse	centre ville rue hassan 2
Code Postale	30110
Telephone	05 35534356
Nombre d'étoiles	5
Description	un hotel qui vous garantie un service de haut qualité

INFORMATIONS RESPONSABLE	
Nom Prenom du responsable	Filali Ismail
Email	ism_21_ail@live.fr
Fonction	directeur
Service	reservation
Telephone	06 64434647

CONFIRMER REJETER

Figure 30 : Vue de visualisation d'Hôtel

Conclusion

Notre projet consistait à concevoir et développer un "GDS" au sein de la société G-fit.

La réalisation de ce projet était faite en trois grandes phases. Dans une première, une étude générale a été menée durant laquelle on a présenté la société d'accueil et le cadre global du projet. La deuxième phase a été consacrée à la présentation des outils et les contraintes du développement, une étude des besoins du projet complétée par une modélisation UML. La troisième phase a consisté en une étude technique qui comprend la mise en œuvre de l'application, ayant pour but de fournir une image prête des parties à coder de l'application.

Nous avons pu, au cours de cette période de stage, satisfaire les besoins de nos clients et développer les cas d'utilisations importants du projet, il reste quelques points non complétés

à citer :

- Le paiement sécurisé par carte bancaire,
- Modifier les dates de réservations,
- Laisser des remarques sur l'hôtel,
- Interface partenaire pour la gestion des réservations.

En perspective, nous pourrions tout d'abord généraliser cette application à l'échelle internationale, et d'autre part généraliser l'application pour l'ensemble du secteur touristique à savoir la restauration, le transport, etc.

Durant ce projet, nous avons pu raffiner nos capacités d'abstraction et de conception ainsi que notre méthodologie de travail. Nous avons aussi développé nos connaissances métiers. Par ailleurs, ce travail nous a permis d'acquérir une double compétence, technique et une culture de bonnes pratiques.

Bibliographies

[Spring Web Flow 2 Web Development, 09] *Spring Web Flow 2 Web Development*, Birmingham, Packt Publishing Ltd, 2009, 271.

[Spring par la pratique, 07] *Spring par la pratique*, Paris, Edition Eyrolles, 2007, 541.

[Spring MVC par l'exemple, 06] serge tahe, « Cours » *Spring MVC par l'exemple*, 2006.

[Persistence Java 5 par la pratique, 07] serge tahe, « Cours » *Persistence Java par la pratique*, 2007.

Webographie

<http://dev.mysql.com/doc/refman/5.0/fr/what-is.html>

http://www.jtips.info/index.php?title=Spring_Web_Flow

<http://laurent-piechocki.Developpez.com/uml/tutoriel>

http://www.jtips.info/index.php?title=Spring_Web_Flow

<http://java.sun.com/javaee/>