

# Mini-Projet

# Développement d'une application pour la gestion d'une agence de voyage

1

## Liste des acronymes

JEE	Jakarta Entreprise (Java EE).
MVC	Modèle Vue Controle.
SGBD	Système de Gestion de Base de Données.
SQL	Structured Query Langauge.
POO	Programation Orientée Objet.
XML	eXtensible Markup Langauge.
HTTP	Hyper Text Transfert Protocol.
MySQL	My Structured Query Language.
HTML	Hypertext Markup Langauge.
CSS	Cascading Style Sheets.
BDD	Base de Données.
JDBC	Java DataBase Connectivity.
JSP	Java Server Pages.

# Résumé

Notre rapport se concentre sur l'analyse, la conception et la réalisation d'une application web son but est la gestion d'une agence de voyage. Ce travail nous a permis l'acquisition d'une expérience enrichissante dans le monde du web.

Le cahier de charges, qui est essentiel dans tel genre de projets, est déterminé en premier temps afin de pouvoir préciser les tâches à satisfaire durant la réalisation de ce projet.

Plusieurs langages et technologies sont utilisés pour construire ce site web dynamique fonctionnel à savoir J2EE, JavaScript,... . Ainsi la dynamité des pages nécessite le développement d'un système de gestion de contenu basé sur une base de données schématisée et manipulé par le SGBD MySQL.

A la fin du projet, nous avons pu créer des pages respectant le contexte général déterminé au niveau du cahier de charges. Par ailleurs, la mise en place du site web a tenu une grande importance à la clarifier dans la représentation du site pour les utilisateurs tout en leurs offrant les fonctionnalités nécessaires pour la gestion.

## Mots clés :

J2EE, JSP, Servlet, MVC, JavaScript, CSS, SGBD, MySQL.

# Table de matières

Liste des acronymes .....	3
Résumé .....	4
Liste des figures .....	6
Introduction générale .....	7
Chapitre 1 : contexte générale du projets .....	8
Chapitre 2 : Analyse et Conception .....	9
Chapitre 3 : environnement et outils de développement .....	14
Mise en œuvre du projet .....	18
Démonstration .....	19
Conclusion générale .....	29
Bibliographies .....	30

## Liste des figures

Figure 1 : diagramme de cas d'utilisation .....	9
---	---

<b>Figure 2 : diagramme de classe.....</b>	<b>10</b>
<b>Figure 3 : MCD .....</b>	<b>11</b>
<b>Figure 4 : Architecture générale de MVC.....</b>	<b>16</b>
<b>Figure 5 : Architecture générale de mon Projet.....</b>	<b>18</b>

# Introduction générale

## **Présentation des étapes du projet :**

Le projet comporte trois chapitres. Le premier chapitre présente le contexte général et description du projet, et centrée sur l'organisation de l'agence du voyage et son partie administratif.

Le second chapitre est consacré à la conception générale puis une conception détaillé en se basant sur les diagrammes du langage de modélisation de l'UML.

Finalement, le dernier chapitre présente les différents outils et langages nécessaires à la réalisation de cette application en faisant référence au résultat final de notre application selon des captures d'écrans prises depuis notre site web.

En guise de conclusion nous donnerons ainsi un récapitulatif de tous nos efforts loués ainsi que de futures perspectives.

# **CHAPITRE 1 : Contexte général du projet.**

Les principales rubriques et fonctionnalités attendues sont les suivantes :

- ❖ présentation de l'agence.
- ❖ présentation des services proposés par l'agence (circuits accompagnés, voyages en individuel, sélection d'hébergements, voyages 4 thèmes, etc.).
- ❖ recherche de circuits par critères (destination, type de voyage, date de départ, durée, budget).
- ❖ sélection de voyages via un panier électronique.
- ❖ gestion d'un compte administrateur (login et mot de passe) avec suivi de commande et ajout de nouveaux voyages (destination, type de voyage (accompagnés ou individuel), dates, prix, thème, hébergement).
- ❖ formulaire de contact.

# CHAPITRE 2 : Analyse et Conception

## le diagramme de cas d'utilisation:

Le diagramme de cas d'utilisation est un diagramme UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel.

### Identification des acteurs :

Acteurs	Rôles
Visiteur	<ul style="list-style-type: none"> <li>- Contacter l'administration</li> <li>- S'inscrire</li> </ul>
Client	<ul style="list-style-type: none"> <li>- S'inscrire.</li> <li>- Se connecter.</li> <li>- Chercher un voyage.</li> <li>- Gérer son profil.</li> <li>- Gérer son panier.</li> <li>- Participer dans un voyage.</li> </ul>
Administrateur	<ul style="list-style-type: none"> <li>- S'inscrire.</li> <li>- Gérer les clients.</li> <li>- Gérer les paniers.</li> <li>- Gérer les voyages.</li> <li>- Gérer les thèmes.</li> <li>- Gérer les Types.</li> <li>- Gérer les Circuits accompagnés.</li> <li>- Gérer les messages.</li> </ul>

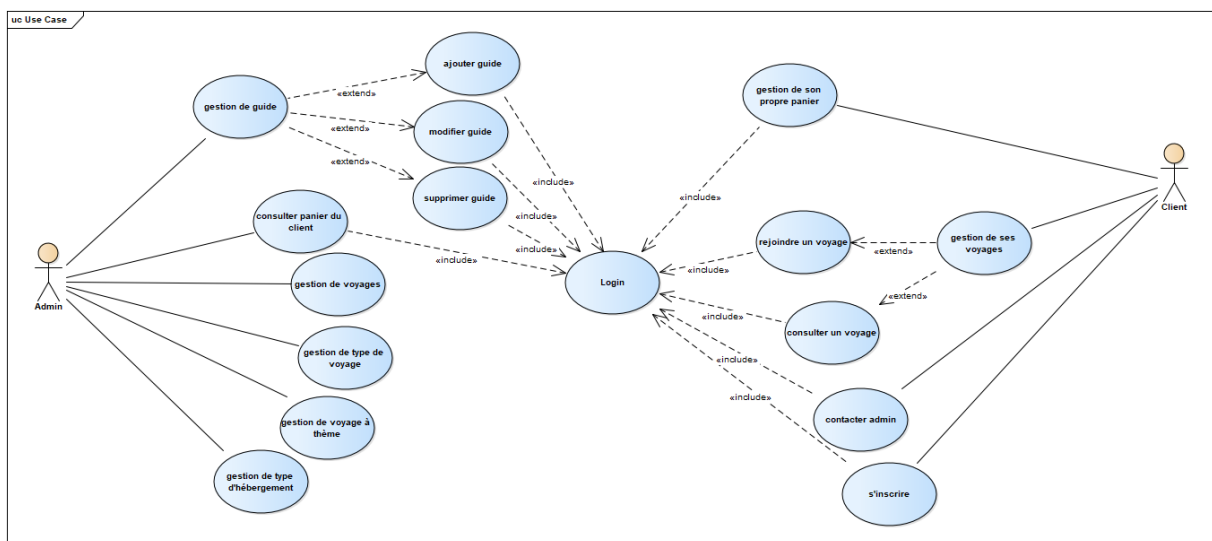


Figure 1 : diagramme de cas d'utilisation



## le diagramme de classe:

Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre elles. De même qu'une classe décrit un ensemble d'objets, une association décrit un ensemble de liens, les objets sont des instances de classes et les liens sont des instances de relations.

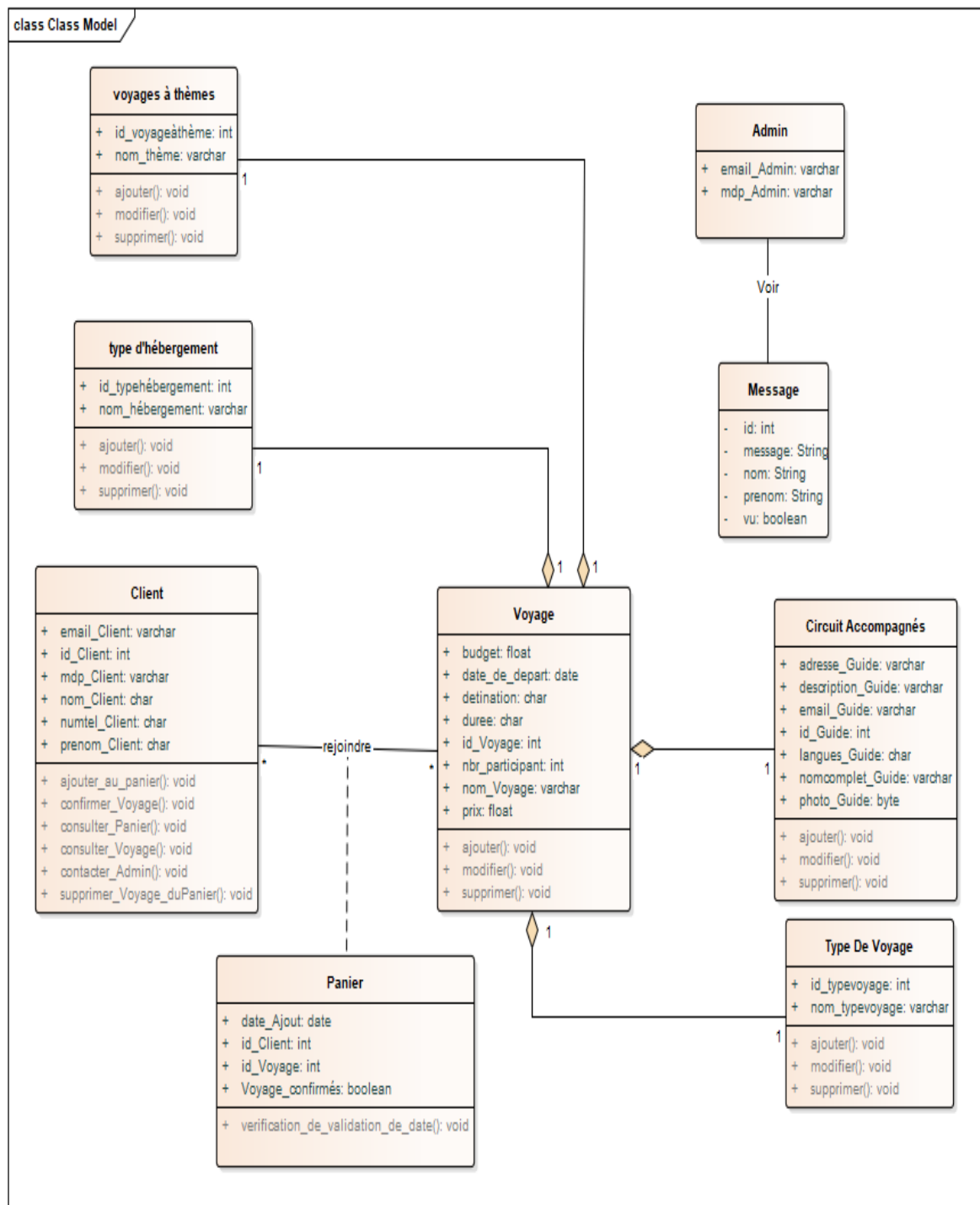


Figure 2 : diagramme classe

## Le MCD:

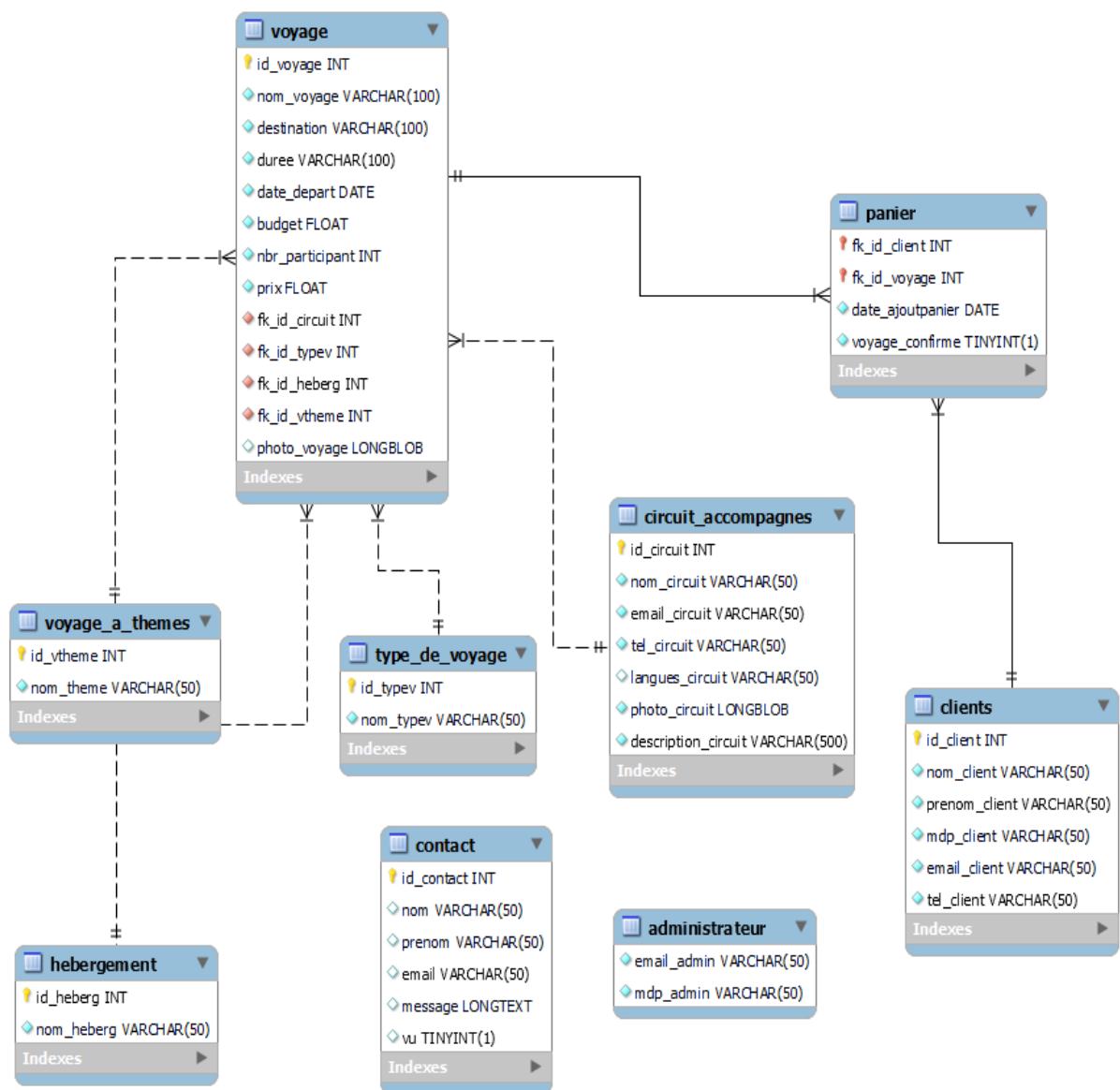


Figure 3 : MCD

## **Le MLD :**

- + clients (id\_client, nom\_client, prenom\_client, mdp\_client, email\_client, tel\_client)
- + circuit\_accompagnes (id\_circuit, nom\_circuit, email\_circuit, adresse\_circuit, langues\_circuit, photo\_circuit, description\_circuit)
- + type\_de\_voyage (id\_typev, nom\_typev)
- + voyages\_a\_themes (id\_vtheme, nom\_vtheme)
- + hebergement (id\_heberg, nom\_heberg)
- + administrateur (email\_admin, mdp\_admin)
- + voyage (id\_voyage, nom\_voyage, destination, duree, date\_depart, budget, nbr\_participant, prix, #id\_circuit, # id\_typev, # id\_heberg, # id\_vtheme)
- + panier (#id\_client,#id\_voyage, date\_ajoutpanier, voyage\_confirme)
- + contact(idcontact,nom,prenom,email,message,vu)

### **Conclusion Partielle:**

Dans ce chapitre, nous avons présenté notre étude conceptuelle du système. La vue fonctionnelle a été illustrée par le diagramme de cas d'utilisation. Ensuite, la vue statique, réalisée par le diagramme de classe qui nous a permis de définir la structure du système et de dégager les différentes entités du composant puis le MCD (modèle conceptuel de base de données) et enfin le MLD (modèle logique des données). Dans le chapitre suivant nous allons présenter des études comparatives, afin de justifier le choix des technologies utilisées.

# **CHAPITRE 3 : Environnement et outils de développement :**

## **Introduction:**

Après avoir achevé l'étape conception de notre application web, nous allons entamer dans ce chapitre la partie réalisation et implémentation dans laquelle on s'assure que le système est prêt pour être exploité par les utilisateurs finaux.

## **choix du langage UML :**

## **Environnement de développement :**



## **Eclipse :**

Eclipse est un IDE, Integrated Development Environment (EDI environnement de développement intégré en français), c'est-à-dire un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation. Il est développé par IBM, est gratuit et disponible pour la plupart des systèmes d'exploitation.



## **Apache Tomcat :**

Apache Tomcat, souvent appelé Tomcat Server, est un conteneur de servlets Java open source développé par Apache Software Foundation (ASF). Tomcat implémente plusieurs spécifications Java EE, y compris Java Servlet, JavaServer Pages (JSP) et WebSocket, et fournit un environnement de serveur web http « Java pur » dans lequel le code Java peut s'exécuter.



## **MySQL:**

Est un système de gestion de bases de données relationnelles (SGBDR). Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server



### **Enterprise Architect:**

Est un logiciel de modélisation et de conception UML, édité par Sparx Systems. Qui couvre, par ses fonctionnalités, l'ensemble des étapes du cycle de conception d'application, il nous permettra de créer les différents diagrammes pendant la conception.

### **Technologies:**

#### **Plateforme Java EE :**

Jakarta Enterprise Edition (anciennement Java EE). Il fait quant à lui référence à une extension de la plate-forme standard. Autrement dit, la plate-forme Java EE construite sur le langage Java et la plateforme Java SE, et elle ajoute un grand nombre de bibliothèques remplissant tout un tas de fonctionnalités que la plate-forme standard ne remplit pas d'origine. L'objectif majeur de Java EE est de faciliter le développement d'applications web : robustes et distribuées déployées et exécutées sur un serveur d'applications.

#### **JSTL :**

JSTL est l'acronyme de Java server page Standard Tag Library. C'est un ensemble de tags personnalisés développé sous la JSR 052 qui propose des fonctionnalités souvent rencontrées dans les JSP. : Elle étend la spécification JSP en ajoutant une bibliothèque de balises pour les tâches courantes, comme le travail sur des fichiers XML, l'exécution conditionnelle, les boucles et l'internationalisation.

#### **XML :**

XML est un langage de balisage qui définit un ensemble de règles pour l'encodage des documents dans un format de qui est à la fois lisible par l'homme et lisible par machine.

#### **Servlets :**

Une "servlet" est une classe Java qui permet de créer dynamiquement des données au sein d'un serveur HTTP. Ces données sont le plus généralement présentées au format HTML, mais elles peuvent également être au format XML ou tout autre format destiné aux navigateurs web.

## JSP (Java Server Pages) :

Une technologie Java qui permet la génération de pages web dynamiques en ajoutant du code java dans des pages HTML interprétés par le serveur. (10)

### le modèle MVC:

Mon projet consiste à concevoir et réaliser une application web dynamique en se basant sur le modèle MVC constitué de trois parties suivants :

**Modèle** : Il gère les données de l'application. Son rôle principal est de récupérer les informations à partir de la base de données, et les envoyer au contrôleur qui fait le traitement. Les données peuvent être stockées dans la BD, dans ce cas le modèle contient des requêtes SQL. On peut aussi chercher des données dans des fichiers, alors le modèle contient les opérations d'ouverture, de lecture et d'écriture de fichiers.

**Vue** : C'est la partie qui gère l'affichage. Elle ne fait presque aucun calcul et se limite à récupérer des variables pour savoir ce qu'elle doit afficher.

**Contrôleur** : Cette partie gère la logique du code qui prend des décisions. C'est l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données afin de les analyser pour prendre des décisions et renvoyer le texte vers la vue.

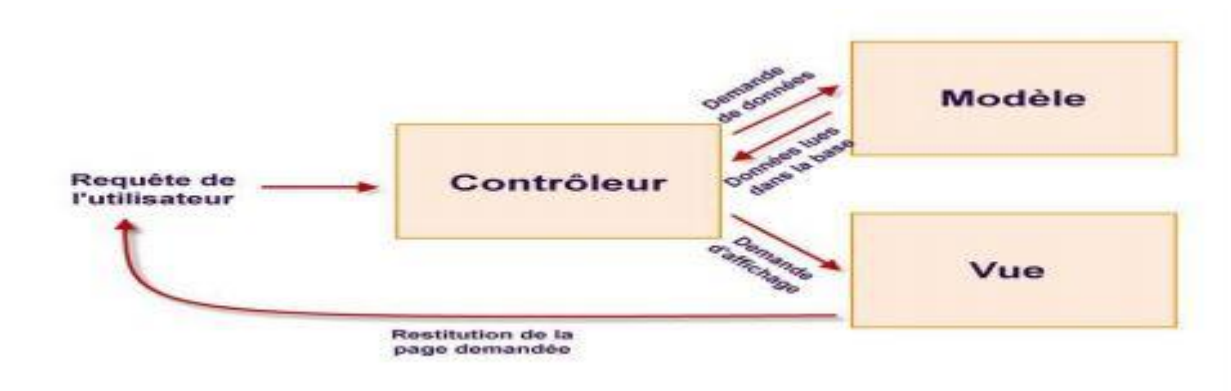


Figure 4 : Architecture générale de MVC

## JSP et Servlets :

Les JSP permettent d'introduire du code Java dans des tags prédéfinis à l'intérieur d'une page HTML. La technologie JSP mélange la puissance de Java côté serveur et la facilité de mise en page d'HTML côté client. Concrètement, les JSP sont basées sur les servlets. Au premier appel de la page JSP, le moteur de JSP génère et compile automatiquement une servlet qui permet la génération de la page web. Le code HTML est repris intégralement dans la servlet.

Le code Java est inséré dans la servlet. La servlet générée est compilée et sauvegardée puis elle est exécutée. Les appels suivants de la JSP sont beaucoup plus rapides car la servlet, conservée par le serveur, est directement exécutée. Il y a plusieurs manières de combiner les technologies JSP, les beans/EJB et les servlets en fonction des besoins pour développer des applications web. Comme le code de la servlet est généré dynamiquement, les JSP sont relativement difficiles à déboguer. Cette approche possède plusieurs avantages :

- L'utilisation de Java par les JSP permet une indépendance de la plateforme d'exécution mais aussi du serveur web utilisé.

La séparation des traitements et de la présentation: la page web peut être écrite par un designer et les tags Java peuvent être ajoutés ensuite par le développeur. Les traitements peuvent être réalisés par des composants réutilisables (des Java beans).

Les JSP sont basées sur les servlets : tout ce qui est fait par une servlet pour la génération de pages dynamiques peut être fait avec une JSP.



# Mise en œuvre du projet

Architecture générale de notre logiciel:

La figure ci-dessous montre l'architecture qui a été mise en place dans le cadre de ce projet, cette architecture est largement admise comme la plus efficace et généralisable à n'importe quel projet Web.

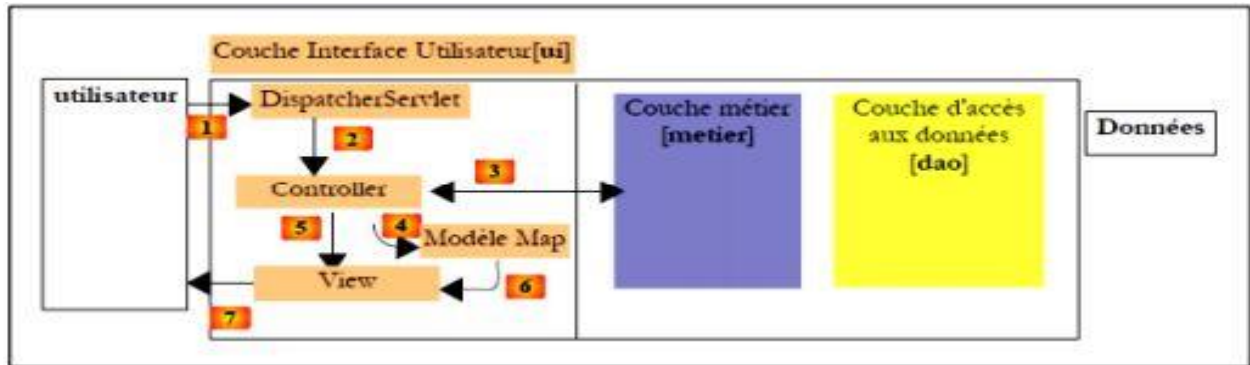


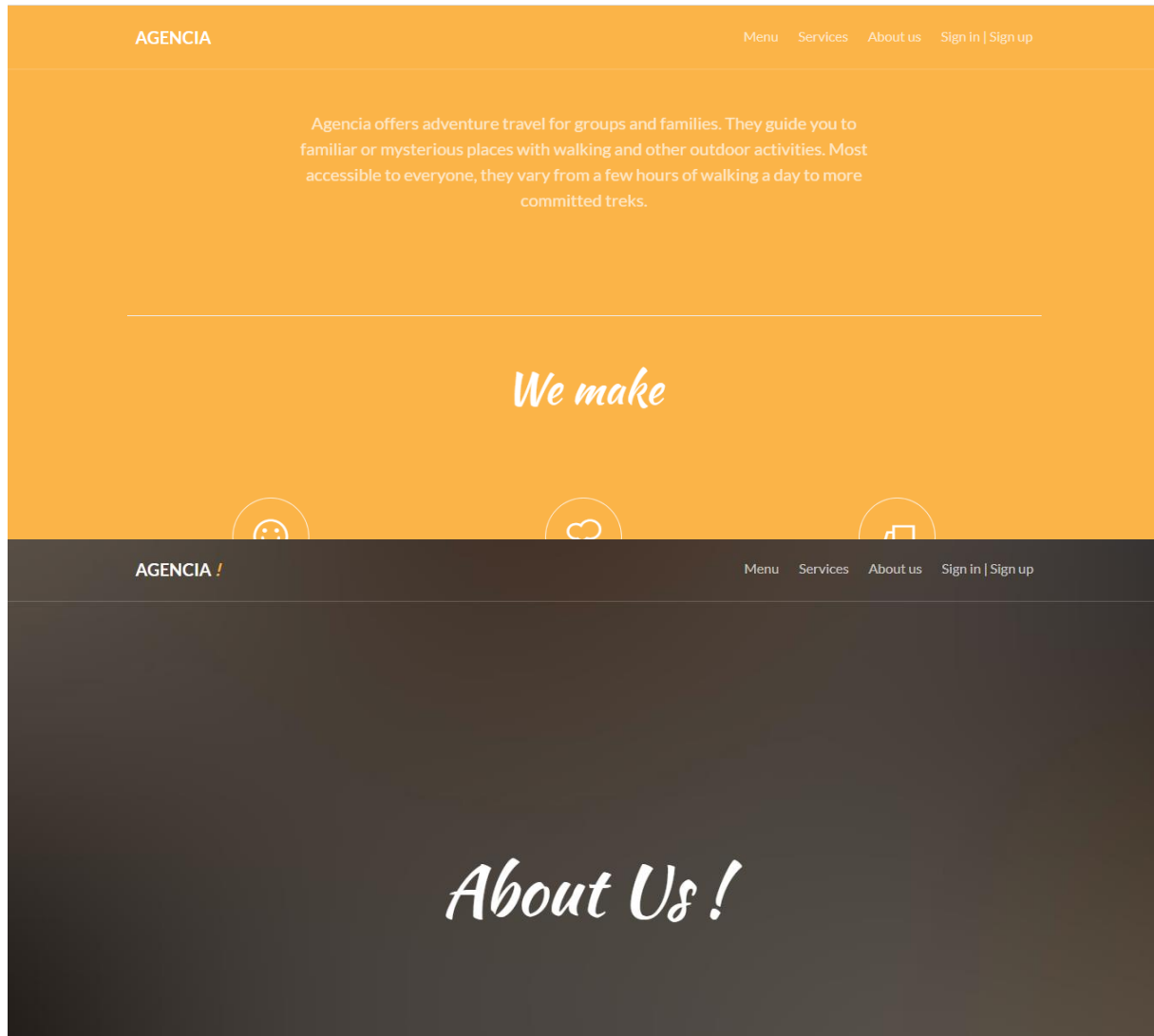
Figure 5 : Architecture générale de Mon logiciel

La principale caractéristique de cette architecture est la séparation des préoccupations (données, Service et Présentation) grâce à la séparation stricte des couches applicatives. En effet on peut observer les trois couches de l'application :

- Couche DAO : permet les accès à la base de données.
- Couche Métier : contient l'ensemble du code service de l'application, elle organise les accès à la couche DAO.
- Couche Présentation: cette couche est la couche d'entrée dans l'application du point de vue d'utilisateur.

# Démonstration:

## Partie Visiteur :




### Get In Touch

Write us something

Send Message

### Contact Information

 Bachelor UMI  
Meknes, Maroc

 +1235 2355 98

 info@yoursite.com

 FreeHTML5.co



## All Our Lodging

Lodging

Auberge

Lodging

Camping

Lodging

Hôtel

Lodging

Hôtel de luxe

Lodging

Motel

Lodging

Tente

## Partie Client :

## Registration

First Name

Last Name

Email

Password


Phone Number

REGISTER

You Have An Account? [Sign In](#)

# Login

Email

 Type your email

Password

 Type your password

LOGIN

You Don't Have An Account?

SIGN UP

# All Our Trips

Try Us Once You Will Enjoy And Be Surprised



## Voyage Vers Agadir

To Agadir

Period of : Week-end



## Voyage Vers Essaouira

To Essaouira

Period of : 1 semaine



## Voyage Vers Ouarzazat

To Ouarzazat

Period of : 2 semaine

# Have A Look!

## Search Table

Destination

Destination ▼

Type de voyage

Type ▼

Date de depart

tt.mm.jjjj 

duree

duree ▼

budget

budget ▼

Search Now

## Details About the trip



Nom Du Voyage

Voyage

Date De Depart

20.01.2022

Budget

1500.0

Prix

1200.0

Type De Voyage

couple

Nom De Circuit

ellena

Theme De Voyage

Découverte

Hebergement

gite

\*Vous Pouvez emmener un binôme

Ajouter Au Panier



*Enjoy Our Trip !*

## *Your Personnal Data*

First Name

hakim

Last Name

Ossor

Email

hakim@gmail.com

Phone

0767574829

Edit

# Voyage Confirmés



## Voyage Vers Agadir

To Agadir

Period of : Week-end

Will begin at 2022-01-20

Number of participants : 10

## Partie administrateur :

Agencia !

Welcome, Administrateur

GENERAL

- Home
- Messages
- Gestion de clients
- Gestion de Voyage
- Circuit Accompagnés
- Type d'hebergements
- Theme de Voyages
- Type de Voyages

7 Total Clients

6 Total Voyages

6 Total Circuits

12 Total Themes

7 Total Types

10 Total Heber...

Gestion Clients

Chercher... Go!

Gestion

Num	Nom	Prenom	Email	téléphone	Panier	Supprimer
1	ntest1	pctest1	emailtest1@gmail.com	0611111111		
2	ntest2	pctest2	emailtest2@gmail.com	0622222222		
3	ntest3	pctest3	emailtest3@gmail.com	0633333333		
4	ayoub	ouhdidou	ayoub@gmail.com	0606060606		
5	Houria	Qarchi	qarchi@gmail.com	064323760		

1		Ayoub Ouhtidou	ouhtidou2002@gmail.com	0600000000			
2		Ayoub Ouhtidou	ouhtidou2@gmail.com	0625737979			
3		Fatima Alaoui	fatima@gmail.com	0634256754			
4		Amine Gaiz	amine@gmail.com	0643458493			
5		Amal berada	amal@gmail.com	0648392843			
6		ali mahmoud	ali@gmail.com	0632456832			

Gestion de clients

Gestion de Voyage

Circuit Accompagnés

Type d'hebergements

Theme de Voyages

Type de Voyages

Welcome,  
Administrateur

GENERAL

Home

Messages

Message non vue

Message vue

Gestion de clients

Gestion de Voyage

RéinitialiserAjouter

Gestion

Num	Type de Voyages	Supprimer
1	en couple	
2	en famille	
3	en groupe	
4	Individuel	

Gestion Message

Num	Nom complet	Email	Message	Supprimer
1	daba daba	daba@gmail.com		
2	safi safi	safi@gmail.com		
3	test khif test khif	test@gmail.com		

28

# Conclusion générale

En Guise de conclusion, nous devons avouer que cette expérience a été très enrichissante, on a pu approfondir nos connaissances théoriques tout en appuyant sur la pratique de tout langages et outils de programmation acquis (Java, UML, SQL...) et sur la maîtrise de l'architecture JEE. On a eu certes quelques difficultés durant la conception et la réalisation pratique mais à la fin on a réussi à surmonter ces difficultés et atteindre notre but et qui est La création d'une application web fonctionnelle et qui remplit toutes les tâches demandées dans le cahier de charge et plus.

OUHDIDOU Ayoub et KHELLA Ranya.

# **Bibliographie**

- Pr Said Benhlima : cours JEE
- Pr Ali Oubelkacem : cours UML
- <http://www.eclipse.org/documentation/>
- <https://www.mysql.com/fr/>
- <https://colorlib.com/>
- <https://www.w3schools.com/sql/>
- Openclassrooms : developpez-des-sites-web-avec-java-ee