

République Algérienne Démocratique et Populaire

Université Abou Bakr Belkaid- Tlemcen

Faculté de Technologie

Département d'Informatique

Mémoire de fin d'études

En vue de l'obtention du Diplôme de Licence en Informatique

Thème :

Application web de gestion d'une agence de voyage (Travel Manager System TMS)

Réalisé par :

- BOUAYAD AGHA Youssouf Mehdi.
- AKBI Leyla.

Encadré par :

- Mme. HALFAOUI Amel.

Présenté le 10 Juin 2014 devant la commission d'examination composée de MM.

- Benamar Abdelkerim (Examineur)
- Belhabi Amel (Examineur)

Année universitaire: 2013-2014

- Résumé -

L'intérêt de ce travail est d'informatiser la gestion d'une agence de voyage, cette informatisation se traduit par la mise en œuvre d'un système le Travel Manager System (TMS) qui gère le fonctionnement de l'agence touristique. L'utilité de ce système est de faciliter la gestion de l'agence pour l'administratif et garder un contact permanent avec le client.

La modélisation du système TMS a été faite par UML en s'appuyant sur le processus UP. La base de données et l'application ont été respectivement créées par le SGBD Oracle et le langage PHP, réputés par leurs puissances, dynamismes et leurs efficacités.

- Remerciements -

En témoignage de notre gratitude et de notre profond respect nous adressons nos vifs et sincères remerciements à :

- *A notre encadreur Madame Halfaoui pour sa disponibilité, ses conseils et son encouragement qui nous ont permis de réaliser ce travail dans les meilleures conditions.*
- *A nos amis El Hadi Mohammed Khedim, Fethi Bendimerad, Anes Bestaoui qui nous ont apporté leurs aides à l'élaboration de ce mémoire, nous leurs devons beaucoup.*
- *Les jurys pour leurs efforts et leurs soins apportés à notre travail.*
- *Aux enseignants de notre université et du département d'informatique.*
- *Nos familles pour leurs soutiens et leurs encouragements au long de toutes ces années.*

- Dédicaces -

Je dédie ce mémoire :

A ma mère qui a toujours cru en moi et en mes capacités, qui a toujours été là pour me motiver, me soutenir, m'aider dans les moments difficiles, à surmonter mes problèmes et veiller sur mon bonheur.

A mon père.

A mon frère, ma sœur et toute ma famille.

A mon fiancé

Et à tous mes ami(e)s.

Leyla AKBI

Je dédie ce mémoire :

Tout d'abord à mes parents qui m'ont apportés tout le confort et les moyens pour avoir le meilleur parcours possible tout en me soutenant tout au long de mon cursus universitaire en me poussant à donner le maximum de moi même

Au reste de la famille

A mes ami(e)s

Mehdi Bouayad Agha

- Table des matières -

- Introduction générale	3
Chapitre1 : Définition du système et du langage de modélisation UML	4
I. Introduction	4
II. Applicationweb	4
II.1. Définition d'une application web.....	4
II.2. Utilisation des applications web	4
II.3. Création d'une application web :	5
III. Présentation d'une Agence de voyage.....	5
III.1. Définition.....	5
III.2. Fonctionnalités d'une agence de voyage	6
IV. Langage de modélisation UML.....	6
IV.1. Définition UML.....	6
IV.2. Les diagrammes UML.....	7
A. Les diagrammes comportementaux	7
B. Les diagrammes structurels	9
V. Processus UP	10
V.1. Définition	10
V.2. Caractéristiques.....	11
➤ Itération et Incrémentation	11
➤ Centré sur l'architecture	11
➤ Piloté par les cas d'utilisation d'UML	11
V.3. Vie du processus unifié	12
VI. Conclusion	13
Chapitre 2 : Modélisation du système TMS	14
I. Introduction	14
II. Description du système.....	14
III. Modélisation du système	15
III.1. Diagramme de cas d'utilisation :	15
A. Identification des acteurs du TMS.....	15
B. Description du scenario	16
III.2. Diagramme de séquences.....	16

A.	Consultation.....	16
B.	Réservation	17
C.	Annulation de la réservation.....	17
III.3.	Diagramme de classes.....	18
IV.	Modèle logique de données TMS	19
V.	Conclusion.....	20
Chapitre 3 : Implémentation du système TMS		21
I.	Introduction	21
II.	Choix du langage de programmation.....	21
II.1.	Définition de PHP	21
II.2.	Définition du langage HTML.....	22
II.3.	Définition de CSS.....	23
III.	Choix de l'outil de développement.....	23
IV.	Choix du SGBD.....	23
IV.1.	Définition d'Oracle.....	24
IV.2.	Les versions d'Oracle	24
IV.3.	Les fonctionnalités d'Oracle	24
IV.4.	Les composants d'Oracle	24
V.	Description de l'application	25
V.1.	Interface administratif	25
V.2.	Interface client	28
VI.	Conclusion.....	30
	- Conclusion générale -.....	31
	- Références bibliographique -	32
	- Table des illustrations -	33
	- Résumé -.....	34
	- Abstract -.....	34
	ملخص.....	34

- Introduction générale

Durant ces dernières années l'informatique s'est imposée d'une manière très impressionnante dans les entreprises, cela est dû à son apport extraordinaire dans le domaine de la gestion des bases de données. En effet, l'informatique désigne l'automatisation du traitement de l'information par un système concret « machine » ou abstrait.

On entend également par « Informatique » l'ensemble des sciences et techniques en rapport avec le traitement de l'information. L'informatique est de plus en plus utilisée dans tous les domaines d'activités y compris celui de la gestion des agences de voyages auquel nous rattacherons d'ailleurs notre étude, et cela pour une meilleure gestion des différents traitements exigés par cette activité pour offrir le meilleur voyage aux clients. Nous avons pu constater, en effet, pendant nos recherches au sein d'une agence touristique qu'une majeure partie des traitements se fait manuellement, ce qui engendre un certain nombre de problèmes tels que la lenteur dans l'accès aux données et le risque de perte d'informations ; donc la meilleure solution pour pallier aux problèmes est l'informatisation afin d'assurer l'accès instantané aux données et une sécurisation de ces dernières, ce qui simplifie le travail administratif.

De ce fait, on a proposé aux responsables de l'agence un logiciel afin de leur concevoir un système d'information automatisé pour la gestion de leurs agences et en même temps une plateforme qui leur permettra d'être en relation permanente avec leurs clients.

Pour la réalisation de cette tâche, notre choix s'est porté sur le Processus Unifié. Le langage de modélisation qu'on a utilisé est UML (Unified Modeling Language), qui est une partie intégrante de la démarche UP.

Notre mémoire est répartie en trois chapitres : le premier chapitre comporte des définitions, présentation du système auteur et présentations du langage UML, le deuxième chapitre présente la phase d'analyse et de conception de notre application « Travel Manager System » en expliquant les diagrammes utilisés et enfin nous terminerons avec le troisième chapitre qui est consacré à la partie réalisation et implémentation de l'application.

Chapitre 1 : Définition du système et du langage de modélisation UML

I. Introduction

Dans ce chapitre nous allons en premier, définir ce que c'est qu'une application web, en second, donner la définition d'une agence de voyage et ses fonctionnalités et pour terminer nous présenterons le langage de modélisation UML (les définitions, les descriptions et l'utilité des différents digrammes) ainsi que le processus UP.

II. Application web

II.1. Définition d'une application web

En informatique, une application web est une application manipulable grâce à un navigateur web. De la même manière que les sites web, une application web est généralement placée sur un serveur et se manipule en actionnant des widgets à l'aide d'un navigateur web, via un réseau informatique (Internet, intranet, réseau local, etc.).

Les messageries web, les systèmes de gestion de contenu, les wikis et les blogs sont des applications web.

Les moteurs de recherches, les logiciels de commerce électronique, les jeux en ligne, les logiciels de forum peuvent être sous forme d'application web.

Des appareils réseaux tels que les routeurs sont parfois équipés d'une application web dans leurs micros logiciels.

Les applications web font partie de l'évolution des usages et de la technologie du Web appelée Web 2.0. [1]

II.2. Utilisation des applications web

La technologie des applications web permet de nombreux usages, les plus populaires sont les moteurs de recherches, le webmail, le commerce électronique et les jeux en ligne.

Un moteur de recherches est une application web qui recherche des documents.

Un webmail est une application web pour recevoir et envoyer du courrier électronique.

Un système de gestion de contenu est une application web qui présente des documents. La présentation des documents est similaire à celle d'un site web, cependant les documents sont générés par le logiciel lors de chaque demande.

Le système de gestion de contenu effectue les traitements nécessaires à la mise en forme et la présentation des documents.

II.3. Création d'une application web :

Les applications web sont souvent créées par des équipes composées à la fois de développeurs et de designers. Le développement nécessite la connaissance des différents langages utilisés dans les technologies du Web : HTML pour la présentation des pages, CSS pour la charte graphique, JavaScript, Java ou ActionScript pour les automatismes exécutés par le client, ainsi qu'un langage tel que Java, PHP, C# ou VBScript pour les automatismes exécutés par le serveur.

Les applications web sont faites d'un ensemble de composants logiciels et de pages "porteuses" ; les composants sont regroupés dans des bibliothèques logicielles (voir *servlets*). Un logiciel serveur web prévu à cet effet (serveur d'applications web) exécute un composant donné lors de la réception de chaque requête. ASP.NET, Websphere, JBoss ou ApacheTomcat sont des logiciels serveurs d'applications web.

Une application web est typiquement utilisée simultanément par plusieurs usagers ; elle est équipée de mécanismes de contrôle d'accès logique, ceux-ci sont basés sur les mécanismes de contrôle d'accès propre au serveur d'application web et au système d'exploitation. Ils utilisent parfois des mécanismes existants tels que l'authentification unique (*Single sign-on*).

Pour les travaux de construction, les ingénieurs utilisent des environnements de développement intégrés qui aident à la fois à la programmation informatique et la conception de site Web tels que Visual Studio ou Eclipse.

III. Présentation d'une Agence de voyage

III.1. Définition

Une **agence de voyages** est une entreprise commerciale qui compose et vend des offres de voyages à ses clients. Elle joue le rôle d'intermédiaire et/ou d'agrégateur de services entre les clients et les différents prestataires sur le marché du tourisme : compagnies aériennes, hôtelières, loueurs de voiture, compagnies d'assurances, etc.[2]

III.2. Fonctionnalités d'une agence de voyage

Une agence de voyage à plusieurs fonctionnalités, parmi ces fonctionnalités on cite :

- Vérifier les prix, la disponibilité pour effectuer les réservations en fonction des choix des clients.
- Créer des offres pour procurer des vacances de rêve et paradisiaque pour le client.
- Recevoir le client et lui donner des conseils judicieux.
- Gérer les clients.
- Formalités pour l'obtention du visa.
- Prise en charge des prestations de transport.

IV. Langage de modélisation UML

Grace à son efficacité et sa modularité par rapport aux autres langages de modélisation, UML est devenu de plus en plus utilisé.

IV.1. Définition UML

UML est un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. En plus de ça, il nous permet de :

- Spécifier de manière précise et complète, et sans ambiguïté.
- Représenter des systèmes entiers.
- Etablir un couplage explicite entre les concepts et les artefacts exécutables.

UML est utilisable pour plusieurs domaines [3] :

- ✓ Systèmes d'information des entreprises.
- ✓ Les Banques et les services financiers.
- ✓ Télécommunications.
- ✓ Défense et aérospatiale.
- ✓ Scientifique.
- ✓ Applications distribuées par le WEB.

IV.2. Les diagrammes UML

Un diagramme donne à l'utilisateur un moyen de visualiser et de manipuler des éléments de modélisation. **UML** définit des diagrammes structurels et comportementaux pour représenter respectivement des vues statiques et dynamiques d'un système. En général, un diagramme représente un aspect ou un point de vue particulier. Les diagrammes peuvent également rassembler des informations liées entre elles.

Au total UML définit neuf types de diagrammes, qui sont classifiés dans le tableau suivant :

Statique (structure)	Dynamique (comportement)
Diagramme de classe	Diagramme de collaboration
Diagramme à objet	Diagramme de séquence
Diagramme de composants	Diagramme état/ transition
Diagramme de déploiement	Diagramme d'activités
	Diagramme de cas d'utilisation.

Figure I.1. Types de diagrammes

A. Les diagrammes comportementaux

Ce sont des diagrammes qui représentent une vue dynamique du système, il existe cinq types de diagrammes comportementaux : [2]

- Les diagrammes de cas d'utilisations

Ils représentent les cas d'utilisations, les acteurs et les relations entre eux.

Les cas d'utilisation (use cases): Ils permettent de définir les limites du système et les relations entre le système et l'environnement. Il se représente par une ellipse contenant le nom du cas (un verbe à l'infinitif)

L'acteur : Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Il se représente par un petit bonhomme avec son nom (i.e. son rôle) inscrit dessous.

- Les diagrammes de collaboration

Les diagrammes de collaboration présentent un ensemble de rôles joués par des objets dans un contexte particulier, ainsi que les liens entre ces objets. Ils montrent également des interactions entre ces objets à travers la représentation d'envois de messages.

- Les diagrammes de séquence

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML, comme les diagrammes de collaboration mais ils ne rendent pas compte du contexte des objets de manière explicite.

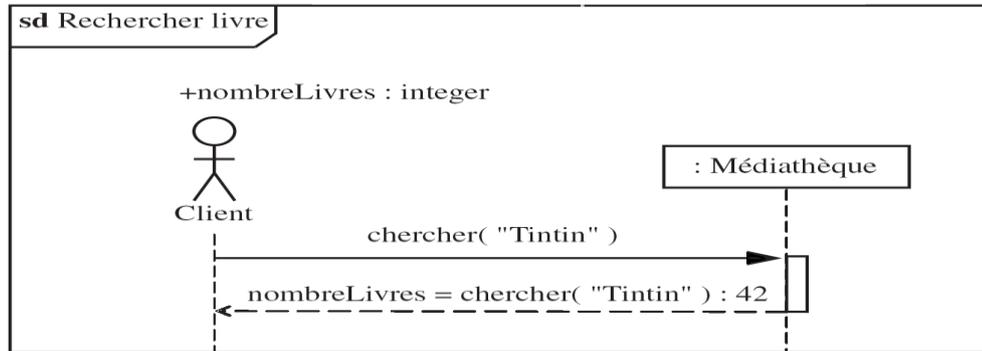


Figure I.2. Exemple d'un diagramme de séquence

- Les diagrammes d'états-transitions

Les diagrammes d'états-transitions visualisent des automates d'états finis, qui représentent le comportement de classes dans la majorité des cas, mais aussi des aspects dynamiques de cas d'utilisation, d'acteurs, de sous-systèmes, d'opérations ou de méthodes. Plus particulièrement, ils décrivent l'évolution des états successifs des éléments auxquels ils sont associés.

- Les diagrammes d'activités

Un diagramme d'activités visualise un graphe d'activités qui modélise le comportement interne d'une méthode (la réalisation d'une opération), d'un cas d'utilisation ou plus généralement d'un processus impliquant l'utilisation d'un ou de plusieurs classificateurs.

B. Les diagrammes structurels

Ce sont des diagrammes qui représentent une vue statique du système, Il existe quatre types de diagrammes structurels :

- Les diagrammes de classes

Le diagramme de classe est généralement considéré comme le plus important dans un développement orienté objet. Il représente l'architecture conceptuelle du système : il décrit les classes que le système utilise, ainsi que leurs liens, que ceux-ci représentent un emboîtement conceptuel (héritage) ou une relation organique (agrégation).

Classes : c'est la description d'un ensemble d'objets qui partage les mêmes attributs, opérations, méthodes, relations et contraintes.

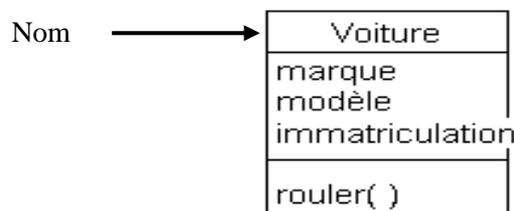


Figure I.3. Exemple d'une classe

Les relations : sont des liaisons entre objets (classes) en modélisation orienté objet. Les 4 relations les plus importantes sont :

➤ Association :

C'est une relation entre deux classes (association binaire) ou plus (association n-aire), qui décrit les connexions structurelle entre leurs instances.

On distingue deux relations de type association :

- **Agrégation :** c'est une association particulière spécifiant une relation 'tout - partie' entre l'agrégat et un composant.
- **Composition :** La composition est une agrégation forte qui lie les cycles de vie entre le composé (ensemble) et les composants (éléments).

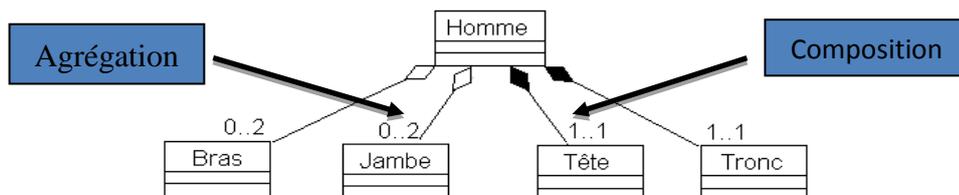


Figure I.4. Exemple des relations d'association

➤ **Dépendance :**

Quand une classe utilise une autre classe, par exemple comme membre ou comme paramètre d'une de ces fonctions, elle "dépend" ainsi de cette classe.

➤ **Généralisation :**

La généralisation décrit une relation entre une classe générale (classe de base ou classe parent) et une classe spécialisée (sous-classe).

➤ **Réalisation :**

Il est parfois souhaitable de modifier une relation dans laquelle une classe représente un élément plus grand qui consiste en un ensemble d'éléments plus petits.

- **Les diagrammes d'objet :**

Un diagramme d'objets est une instance d'un diagramme de classes qui représente des objets (i.e. instances de classes) et leurs liens (i.e. instances de relations) pour donner une vue de l'état du système à un instant donné.

- **Les diagrammes de composants**

Le diagramme de composants décrit les composants du système (fichiers, bibliothèques, Bases de données...) et leurs dépendances dans l'environnement de réalisation d'un point de vue physique, tels qu'ils sont mis en œuvre.

- **Les diagrammes de déploiement**

Un diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels.

V. Processus UP

V.1. Définition

Le Processus Unifié (PU ou UP en anglais pour **Unified Process**) est un processus de développement logiciel, il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel.

L'objectif d'un processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques.[4]

V.2. Caractéristiques

Les principales caractéristiques du processus unifié sont :

- Itération et Incrémentation

L'itération est une répétition d'une séquence d'instructions ou d'une partie de programme un nombre de fois fixé à l'avance ou tant qu'une condition définie n'est pas remplie, dans le but de reprendre un traitement sur des données différentes.

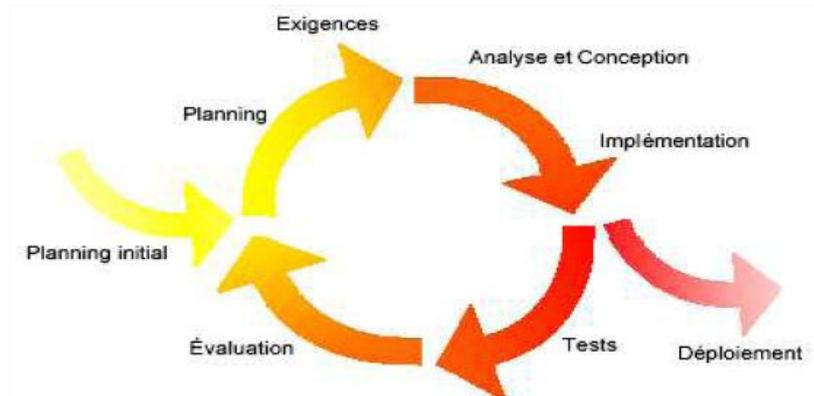


Figure I.5. Schéma d'itération du processus UP

- Centré sur l'architecture

Tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Le processus unifié préconise le modèle des 4+1 vues, de **Philippe Kruchten**, pour représenter l'architecture :

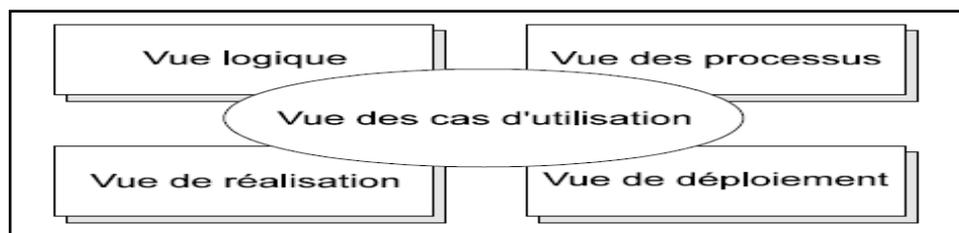


Figure I.6. Le modèle 4+1 vues

- Piloté par les cas d'utilisation d'UML

Le but principal d'un système informatique est de satisfaire les besoins du client.

Le processus de développement sera donc axer sur l'utilisateur.

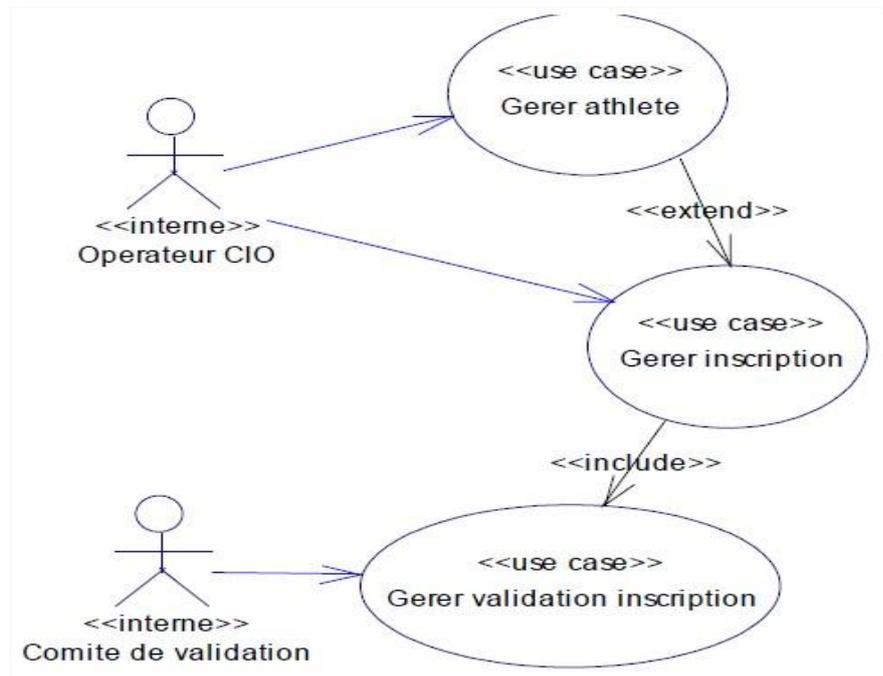


Figure I.7. Exemple d'un diagramme de cas d'utilisation

V.3. Vie du processus unifié

UP gère le processus de développement par deux axes :

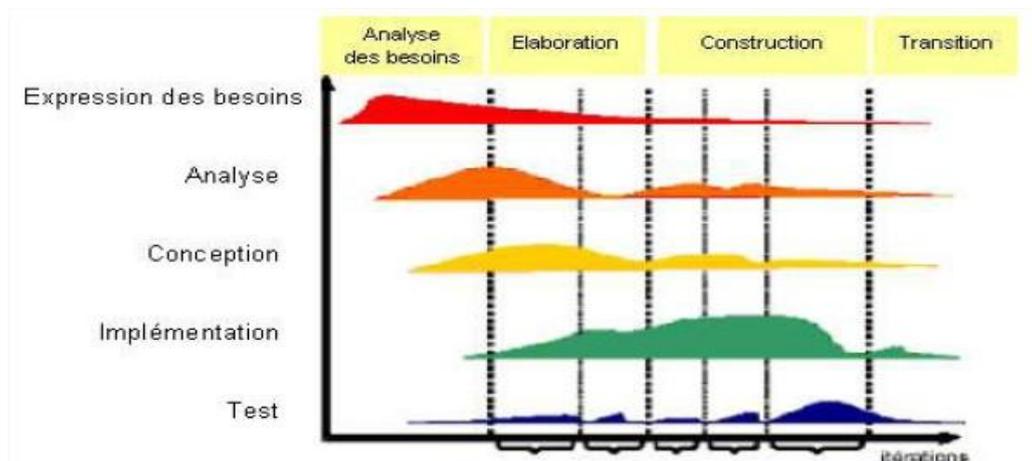


Figure I.8. Cycle de vie du processus UP

L'axe vertical représente les principaux enchaînements d'activités, qui les regroupent selon leurs natures.

L'axe horizontal représente le temps et montre le déroulement du cycle de vie du processus; cette dimension rend compte de l'aspect dynamique du processus qui s'exprime en terme de cycles, de phases, d'itérations et de jalons.

VI. Conclusion

Comme UML n'impose pas de méthode de travail particulière, il peut être intégré à n'importe quel processus de développement logiciel de manière transparente. Parmi les processus cités précédemment, nous avons choisi le processus UP qu'on utilisera pour modéliser notre système. Cette modélisation sera détaillée dans le chapitre suivant.

Chapitre 2 : Modélisation du système TMS

I. Introduction

Dans ce chapitre nous allons faire la conception de notre système, comme nous avons déjà cité dans le chapitre précédent, notre choix s'est porté sur le processus UP, en conséquence, nous allons détailler trois étapes:

- Tout d'abord, nous commencerons par spécifier les besoins de notre système en définissant le diagramme de cas d'utilisations.
- Ensuite les diagrammes de cas d'utilisations vont être détaillés en plusieurs diagrammes de séquences.
- Nous terminerons par représenter le diagramme de classes qui décrit la structure statique de notre système.

II. Description du système

Le Travel Manager System TMS est un système qui consiste à améliorer et faciliter la gestion de l'agence de voyage pour l'administratif et avoir l'accès à l'information au moment opportun, notre système donne aussi la possibilité aux clients de consulter ou réserver les offres disponibles en temps réel. L'intérêt du TMS est de passer d'une gestion manuelle à une gestion automatique utilisable par divers membres en particulier le propriétaire de l'agence et de garder un contact permanent avec le client.

- **Spécification des besoins :**

-Le client peut consulter :

- une offre.
- Un trajet avec une ville de départ, ville d'arrivée, date de départ, date d'arrivée ;
- Hôtels disponibles date d'entrée, date de sortie et la ville.

-Le client peut réserver offre /transport /hôtel pour cela il doit s'identifier ou créer un compte (nom, prénom, date de naissance, adresse, adresse mail, numéro de téléphone,...).

- Proposer une offre avec des critères bien précis en envoyant un message.

- l'administratif gère les informations sur les clients et les offres tout en ayant accès aux statistiques (par date, par destination).

III. Modélisation du système

Pour modéliser le système TMS, on commence par partager ce système en plusieurs cas d'utilisations. Chaque cas d'utilisation correspond à un ou plusieurs scénarios.

III.1. Diagramme de cas d'utilisation :

Les cas d'utilisations et les acteurs TMS sont schématisés dans le diagramme représenté à la Figure II.1.

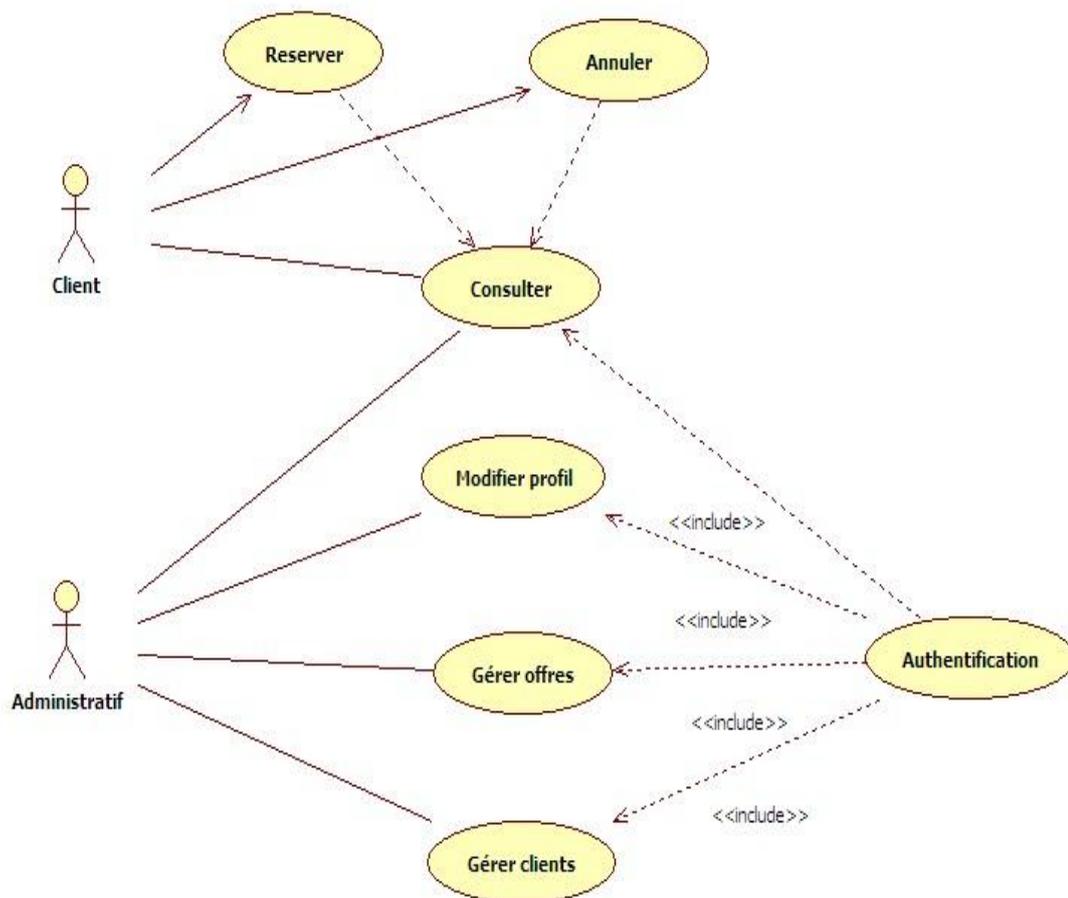


Figure II.1. Diagramme de cas d'utilisation

A. Identification des acteurs du TMS

Notre système comporte deux acteurs, qui sont :

1. L'administratif.
2. Le client.

Qui interagissent entre eux ou avec le système.

B. Description du scenario

- Consulter :

Le client peut consulter :

- une offre.
- Un trajet avec une ville de départ, ville d'arrivée, date de départ, date d'arrivée.
- Hôtels disponibles date d'entrée, date de sortie et la ville.
- Statut de la réservation.

- Réserver :

Le client peut réserver offre /transport /hôtel pour cela il doit s'identifier ou créer un compte (nom, prénom, date de naissance, adresse, adresse mail, numéro de téléphone...).

- Gérer offres / clients :

L'administratif gère les clients et les offres (ajouter, modifier, supprimer) et pour cela faudra qu'il s'identifie.

III.2. Diagramme de séquences

A. Consultation

Dans ce diagramme (Figure II.2) le client consulte les offres et leurs disponibilités

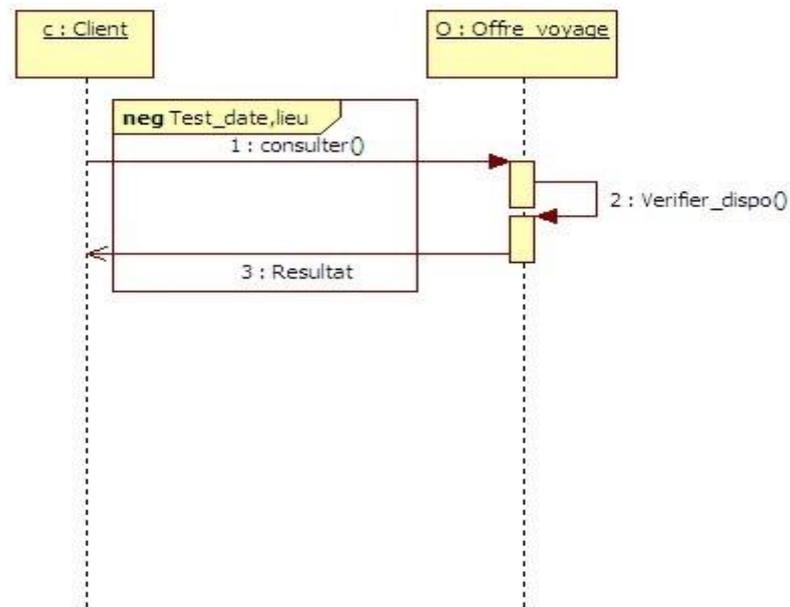


Figure II.2. Diagramme de séquence « consulter offre »

B. Réservation

Dans ce diagramme (Figure II.3) le client peut consulter les offres et leurs disponibilités et faire une réservation, une fois la réservation faite l’administratif envoie un message de confirmation.

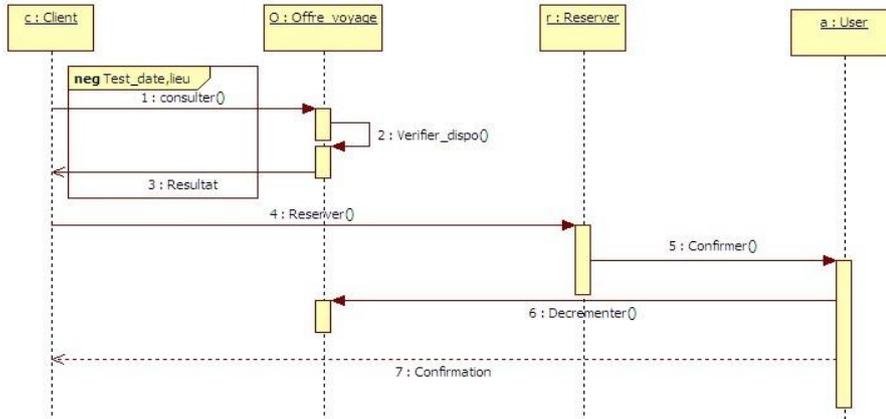


Figure II.3. Diagramme de séquence « réserver offre »

C. Annulation de la réservation

Dans ce diagramme (Figure II.4) le client après avoir fait une réservation, il peut l’annuler, une fois la réservation annulée l’administratif envoie un message de confirmation.

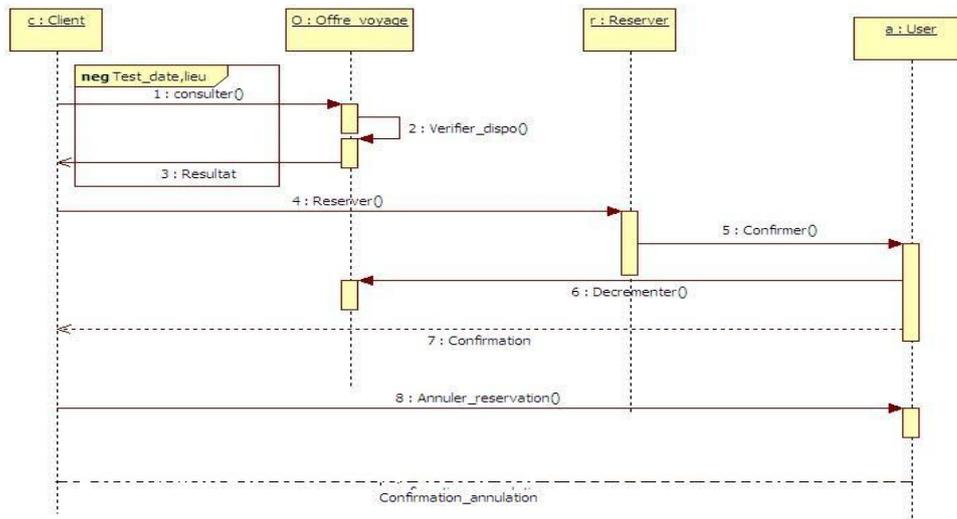


Figure II.4. Diagramme de séquence « Annulation de la réservation »

III.3. Diagramme de classes

Le diagramme de classes identifie les classes de notre système et les associations entre elles. Ce diagramme contient huit classes.

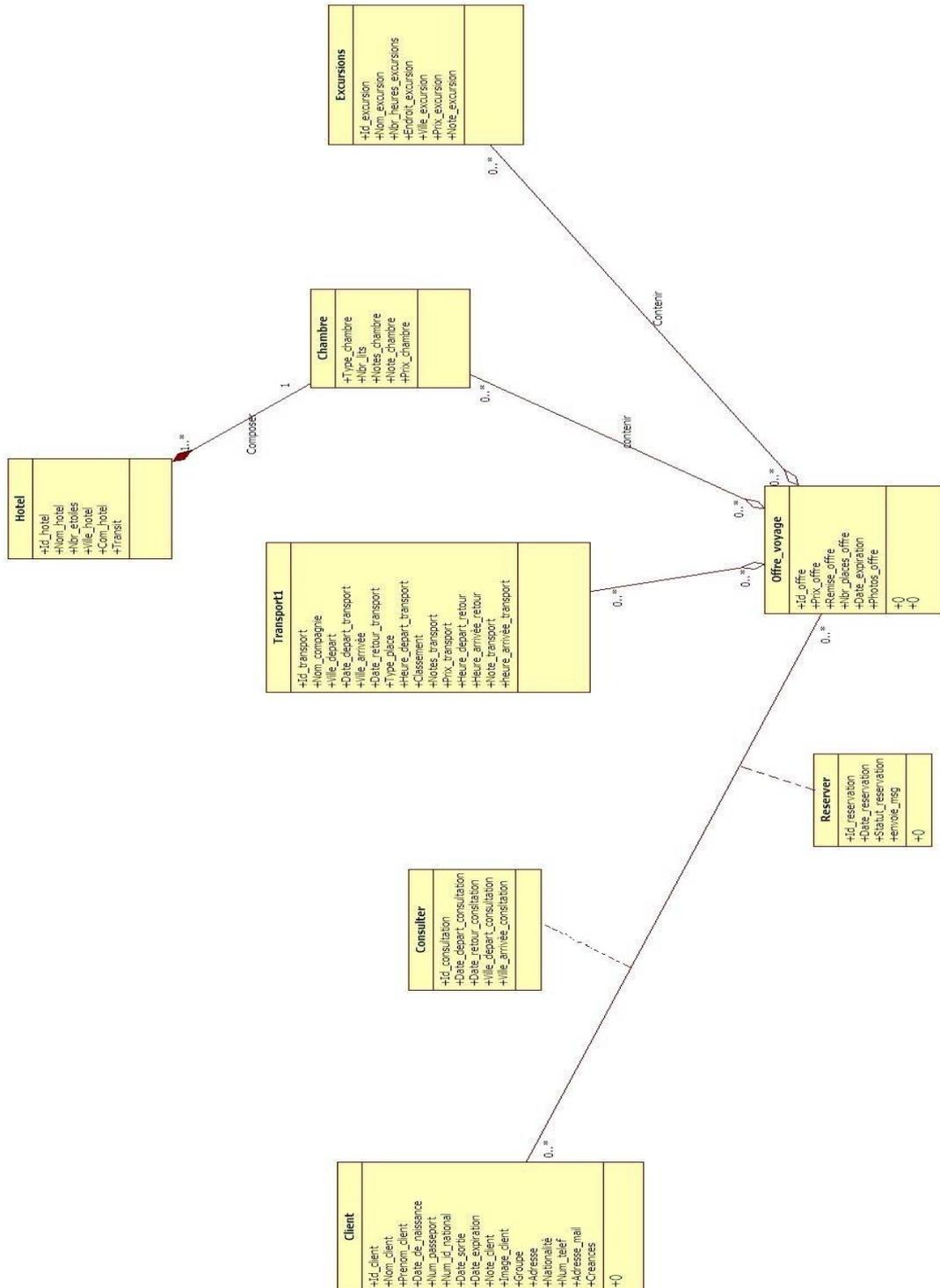


Figure II.5. Diagramme de classes

IV. Modèle logique de données TMS

Le modèle logique présenté ci-dessous a été obtenu en appliquant les règles de passage sur le diagramme de classes (Figure II.5).

Client (ID_CLIENT, NOM_CLIENT, PRENOM_CLIENT, DATE_DE_NAISSANCE, NUM_PASSEPORT, NUM_ID_NATIONAL, DATE_SORTIE, DATE_EXPIRATION, NOTE_CLIENT, IMAGE_CLIENT, GROUPE, NUM_TELEPHONE, ADRESSE_MAIL, CREANCES, NATIONALITÉ_CLIENT, ADRESSE).

Consulter (ID_CONSULTATION, VILLE_DEPART, VILLE_ARRIVEE, DATE_DEPART, DATE_RETOUR, ID_CLIENT*, ID_OFFRE*).

Réserver_offre (ID_RESERVATION, ID_CONSULTATION*, DATE_RESERVATION, STATUT_RESERVATION).

Offre_voyage(ID_OFFRE, PRIX_OFFRE, REMISE_OFFRE, NBR_PLACES_OFFRE, DATE_EXPIRATION_OFFRE, PHOTOS_OFFRE).

Transport (ID_TRANSPORT, ID_OFFRE*, TYPE_TRANSPORT, NOM_COMPAGNIE, TYPE_PLACE, CLASSEMENT_COMPAGNIE, DATE_DEPART_TRANSPORT, DATE_RETOUR_TRANSPORT, HEURE_DEPART_TRANSPORT, HEURE_ARRIVEE_TRANSPORT, HEURE_DEPART_RETOUR_TRANSPORT, HEURE_ARRIVEE_RETOUR_TRANSPORT, COMMENTAIRES_TRANSPORT, VILLE_DEPART_TRANSPORT, VILLE_ARRIVEE_TRANSPORT, PRIX_TRANSPORT).

Excursion (ID_EXCURSION, NOM_EXCURSION, NBR_HEURES_EXCURSION, ENDROIT_EXCURSION, VILLE_EXCURSION, NOTE_EXCURSION, ID_OFFRE*, PRIX_EXCURSION).

Chambre (ID_CHAMBRE, TYPE_CHAMBRE, NBR_LITS, NOTE_CHMABRE, PRIX_CHAMBRE, ID_HOTEL, ID_OFFRE).

Hôtel (ID_HOTEL, NOM_HOTEL, NBR_ETOILES, VILLE_HOTEL, COM_HOTEL, TRANSIT_INCLU).

V. Conclusion

Nous avons réalisé dans ce chapitre la conception UML du système TMS. Cette conception est une étape nécessaire et très importante pour pouvoir créer notre base de données et réaliser l'application, le prochain chapitre détaillera les étapes de l'implémentation de l'application TMS.

Chapitre 3 : Implémentation du système TMS

I. Introduction

Après avoir présenté une étude conceptuelle des données et de traitements dans le chapitre précédent, Nous allons, dans ce chapitre, justifier le choix du langage, donner un bref aperçu sur les outils utilisés, présenter les résultats de notre travail, et finir par une conclusion.

II. Choix du langage de programmation

Pour la réalisation de notre système nous avons choisi PHP un langage de programmation libre qui a été conçu pour créer des sites "vivants" (dynamique) mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté-objet.

Quel que soit le site web que l'on souhaite créer, HTML et CSS sont indispensables, donc nous avons utilisé aussi HTML5 et CSS3, ces langages sont la base de tous les sites web.

II.1. Définition de PHP

PHP est un langage de script utilisé le plus souvent côté serveur : dans cette architecture, le serveur interprète le code PHP des pages web demandées et génère du code (HTML, XHTML, CSS par exemple) et des données (JPEG, GIF, PNG par exemple) pouvant être interprétés et rendus par un navigateur. PHP peut également générer d'autres formats comme le WML, le SVG, le PDF.

Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent développées pour le Web. PHP est le plus souvent couplé à un serveur Apache bien qu'il puisse être installé sur la plupart des serveurs HTTP tel que IIS. Ce couplage permet de récupérer des informations issues d'une base de données, d'un système de fichiers(contenu de fichiers et de l'arborescence) ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées pour une utilisation ultérieure.

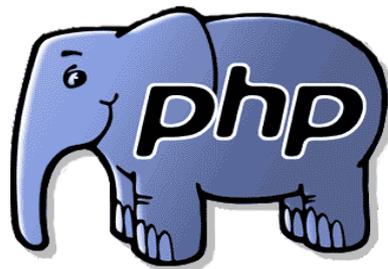
C'est un langage peu typé et souple et donc facile à apprendre par un débutant mais, de ce fait, des failles de sécurité peuvent rapidement apparaître dans les applications. Pragmatique, PHP ne s'encombre pas de théorie et a tendance à choisir le chemin le plus

direct. Néanmoins, le nom des fonctions (ainsi que le passage des arguments) ne respecte pas toujours une logique uniforme, ce qui peut être préjudiciable à l'apprentissage.

Son utilisation commence avec le traitement des formulaires puis par l'accès aux bases de données. L'accès aux bases de données est aisé une fois l'installation des modules correspondant effectuée sur le serveur. La force la plus évidente de ce langage est qu'il a permis au fil du temps la résolution aisée de problèmes autrefois compliqués et est devenu par conséquent un composant incontournable des offres d'hébergements.

Il est multiplate-forme : autant sur Linux qu'avec Windows il permet aisément de reconduire le même code sur un environnement à peu près semblable (prendre en compte les règles d'arborescences de répertoires qui peuvent changer).

Libre, gratuit, simple d'utilisation et d'installation, ce langage nécessite comme tout langage de programmation une bonne compréhension des principales fonctions usuelles ainsi qu'une connaissance aigüe des problèmes de sécurité liés à ce langage. [5]



L'éléPHPant, la mascotte de PHP

II.2. Définition du langage HTML

L'**HypertextMarkupLanguage**, généralement abrégé **HTML**, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web.[6]

II.3. Définition de CSS

Les feuilles de styles (en anglais "Cascading Style Sheets", abrégé CSS) sont un langage qui permet de gérer la présentation d'une page Web. Le langage CSS est une recommandation du World Wide Web Consortium. Les styles permettent de définir des règles appliquées à un ou plusieurs documents HTML. Ces règles portent sur le positionnement des éléments, l'alignement, les polices de caractères, les couleurs, les marges et espacements, les bordures, les images de fond, etc. [7]

III. Choix de l'outil de développement

Nous avons choisi comme outil de développement le **WampServer** qui permet de développer des applications Web dynamiques à l'aide du serveur Apache, du langage de scripts PHP et d'une base de données ORACLE en activant les extensions « oci8 » et « oci8_11g » La plate-forme possède également PHPMyAdmin pour gérer plus facilement les bases de données. Contrairement aux autres solutions, WampServer permet de reproduire fidèlement son serveur de production.[8]

Et **Bootstrap** qui est un ensemble de composants structurés qui sert à créer les fondations et à organiser le code HTML et CSS pour faciliter le travail des programmeurs, que ce soit en termes de productivité ou de simplification de la maintenance, il permet aussi de créer l'architecture d'une page web. [9]

IV. Choix du SGBD

De nombreux SGBD (Système de Gestion de Bases de Données) sont disponibles sur le marché, partant des SGBD gratuits jusqu'à ceux destinés spécialement aux professionnels, comportant de plus nombreuses fonctionnalités, mais plus coûteux. Citons quelques-uns :

MsAccess, SQL Server Oracle &TimesTen&DBLibrary, DB2, Interbase (Delphi) Sybase, SQL Anywhere ,Adabas , MySQL, PostgreSQL, OpenOfficeBase, Monet , SD-SQL-Server...

Nous avons choisi comme SGDB Oracle version express 2.1.0.00.39

IV.1. Définition d'Oracle

Oracle est un SGBD édité par la société du même nom, leader mondial des bases de données.[10]

IV.2. Les versions d'Oracle

Oracle se décline en plusieurs versions

- Oracle Server **Standard**, une version comprenant les outils les plus courants de la solution Oracle. Il ne s'agit pas pour autant d'une version bridée...
- Oracle Server **Enterprise Edition**.

IV.3. Les fonctionnalités d'Oracle

Oracle est un SGBD permettant d'assurer :

- La définition et la manipulation des données
- La cohérence des données
- La confidentialité des données
- L'intégrité des données
- La sauvegarde et la restauration des données
- La gestion des accès concurrents

IV.4. Les composants d'Oracle

Outre la base de données, la solution Oracle est un véritable environnement de travail constitué de nombreux logiciels permettant notamment une administration graphique d'Oracle, de s'interfacer avec des produits divers et d'un assistant de création de bases de données et de configuration de celles-ci.

On peut classer les outils d'Oracle selon diverses catégories :

- Les outils d'administration
- Les outils de développement
- Les outils de communication
- Les outils de génie logiciel
- Les outils d'aide à la décision

V. Description de l'application

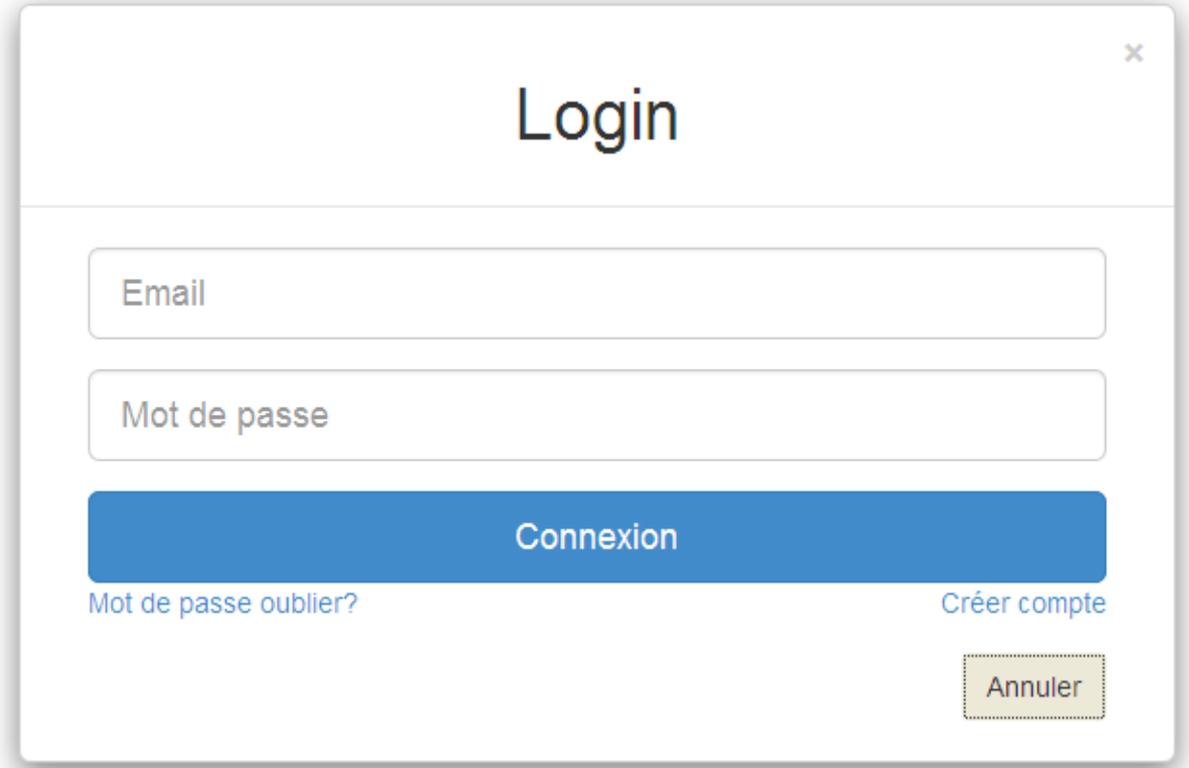
Notre application se décompose en deux parties:

- Partie client.
- Partie administratif.

V.1. Interface administratif

- **L'écran d'identification :**

Dans cet écran l'utilisateur doit s'identifier en donnant son nom d'utilisateur et son mot de passe, afin d'accéder à la l'application TMS, si les informations ne sont pas correctes, le message ('vérifier votre mot de passe') sera affiché.



The screenshot shows a 'Login' window with a title bar containing a close button (X). The main content area has a large 'Login' title. Below the title are two input fields: 'Email' and 'Mot de passe'. A prominent blue button labeled 'Connexion' is centered below the input fields. Underneath the 'Connexion' button, there are two links: 'Mot de passe oublier?' on the left and 'Créer compte' on the right. At the bottom right of the window, there is a button labeled 'Annuler' with a dashed border.

Figure III.1. Ecran d'authentification

- **L'écran principale pour l'administratif:**

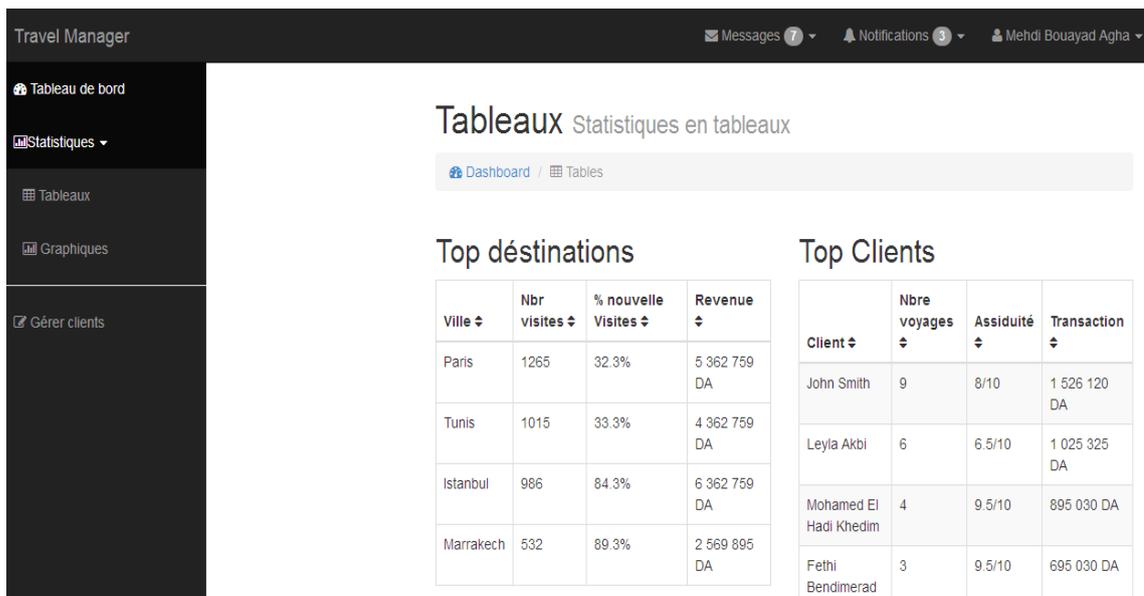
Dans cette fenêtre l'administratif consulte les messages reçus de la part des clients, reçoit des alertes et les informations sur les nouvelles réservations et celles qui ont été annulées, peut aussi faire la gestion des clients et des offres.



Figure III.2. Ecran du tableau de bord

- **L'écran des statistiques :**

Cet écran permet à l'administratif d'avoir une idée sur le nombre de voyages avec des statistiques sous forme de tableaux voir figure III.3 et graphiques figure III.4.



Offres consultés

Offres ↕	Visites ↕	% Nouvelles Visites ↕
Offre 1	1265	32.3%
Offre 2	261	33.3%
Offre 3	665	21.3%
Offre 4	9516	89.3%
Offre 5	23	34.3%
Offre 6	421	60.3%
Offre 7	1233	93.2%

Figure III.3. Statistiques sous forme de tableaux

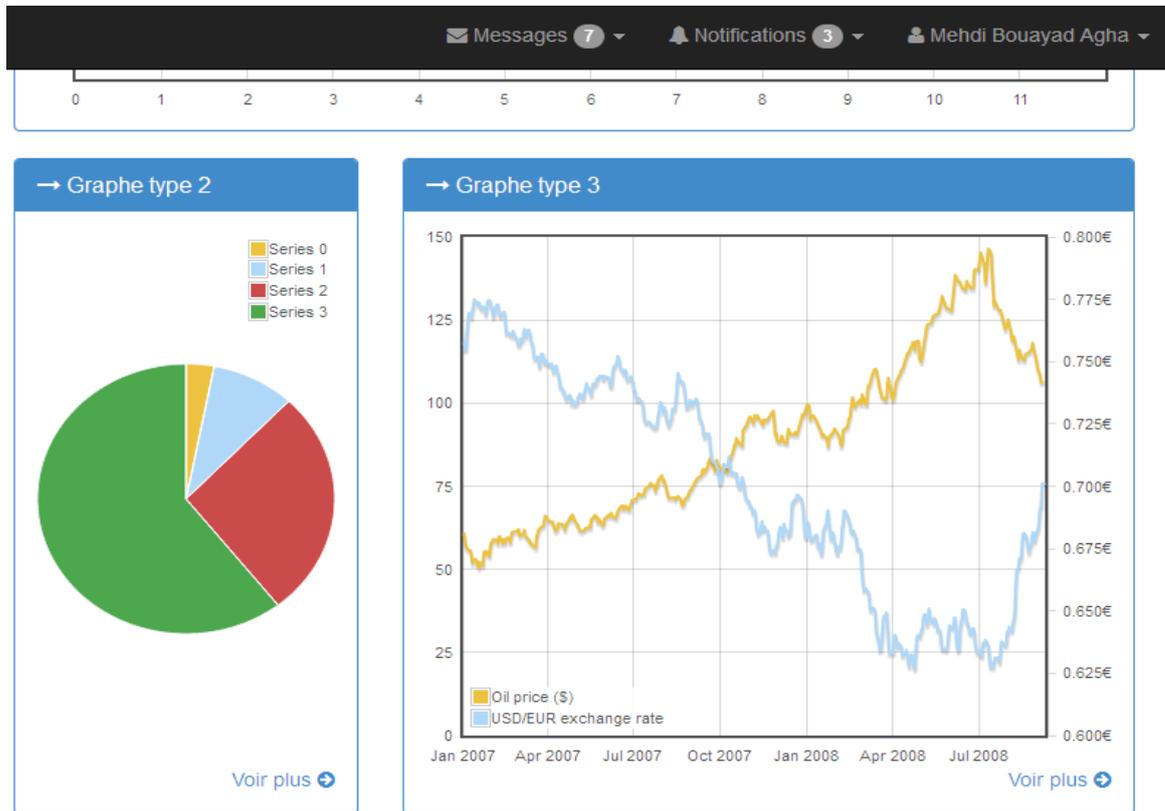


Figure III.4. Statistiques sous forme de graphe

- **L'écran de gestion de client :**

The image shows a web registration form titled "Formulaire d'inscription". The form contains the following fields and elements:

- Two input fields for "Nom*" and "Prénom*".
- An input field for "Email*".
- A date picker for "Date De naissance*".
- Input fields for "Address*", "Adresse 1", and "Adresse 2".
- Input fields for "Ville" and "Pays" (with a dropdown menu labeled "— Select a country —").
- Input fields for "Code postal" and "Phone*".
- A "Photo d'identité*" section with a file selection button "Aucun fichier sélectionné" and a "Choisir fichier" button.
- A dropdown menu for "Numero identité national*".
- Date pickers for "Date sortie passeport*" and "Date expiration passeport*", both in "yyyy-mm-dd" format.
- An input field for "Crédits".
- A large text area for "Des questions/commentaires?".
- A green "Submit" button at the bottom right.

Figure III.5. Formulaire d'inscription

V.2. Interface client

- **page d'accueil :**

Cet écran comporte :

- Une barre de menu se composant de liens envoyant vers les pages :
 - ✓ Accueil.
 - ✓ Offres.
 - ✓ Blog.
 - ✓ Contact.
 - ✓ Recherche.
- Un bloc pour une recherche rapide sur :
 - ✓ les transports
 - ✓ les hôtels.
- Un bloc réservé à l'inscription pour la newsletter des offres.

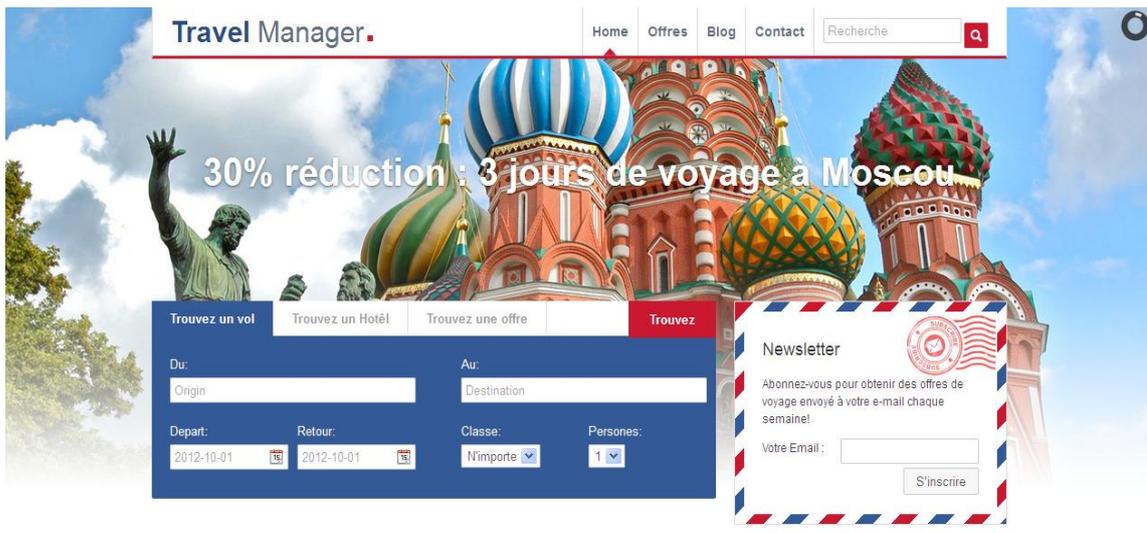


Figure III.6. Ecran de page d'accueil

- Ecran de recherche :

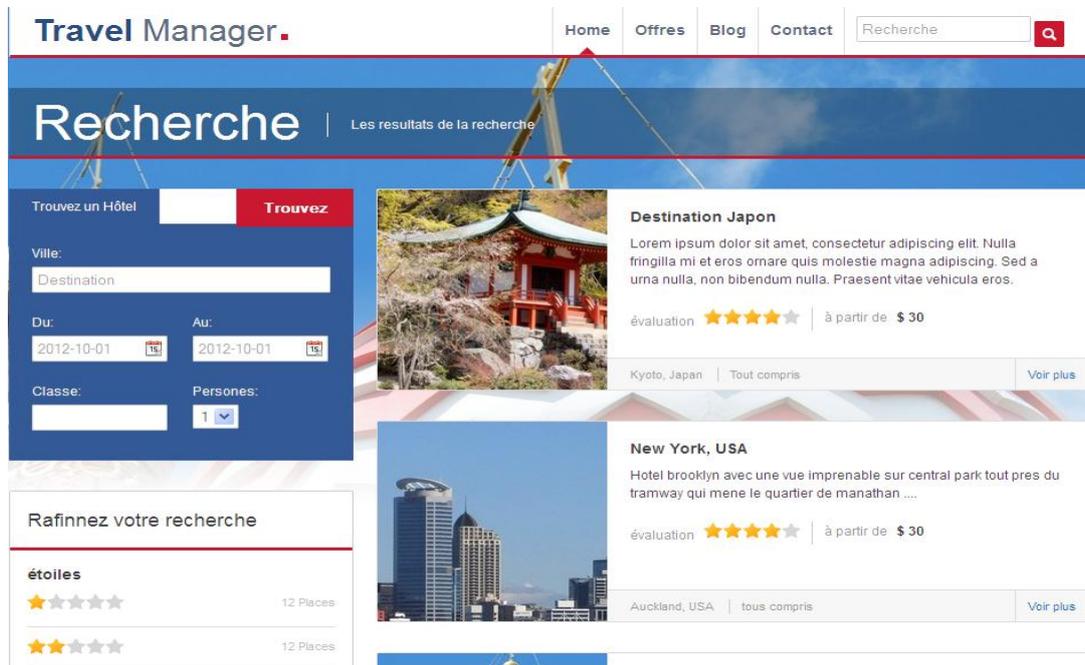


Figure III .7. Ecran de recherche

VI. Conclusion

Dans ce chapitre, Nous avons fait la partie réalisation de notre projet, Nous avons justifié notre choix du langage. Vu le nombre des interfaces composant notre application, on s'est contenté de donner quelques-unes qui nous paraissent les plus importantes.

Nous avons essayé de prendre en considération les principaux besoin de ce projet, en tenant compte des points de vue de l'utilisateur.

- Conclusion générale -

A travers ce projet, nous nous sommes intéressés au fonctionnement d'une agence de voyage. Une étude a été réalisée, en récoltant les informations nécessaires. Le but de ce travail est d'automatiser la gestion de l'agence touristique.

Ce projet a contribué à améliorer nos compétences dans plusieurs domaines. Il nous a permis de nous perfectionner en améliorant nos connaissances en conception et en programmation.

Nous avons appliqué au maximum possible les règles de bases permettant d'avoir une application performante. Nous avons utilisé le langage UML pour modéliser le système en suivant les étapes du processus UP en utilisant Oracle comme SGBD et le langage PHP pour implémenter notre application.

Du fait que PHP est un langage interopérable et flexible, l'application est portable et opérationnel indépendamment de toute plateforme.

Le système TMS ainsi construit, fonctionne correctement et répond aux besoins énoncés, mais peut être amélioré. Nous prévoyons de faire:

- A court terme :
 - Acheter un hébergement dans un serveur pour notre système.
- A moyen terme :
 - Enrichir notre système avec de nouveaux modules tel que le module de gestion de comptabilité.
 - Développer une application mobile sous Android, Windows phone, etc.

- Références bibliographique -

- [1] http://fr.wikipedia.org/wiki/Application_web
- [2] http://fr.wikipedia.org/wiki/Agence_de_voyages
- [3]:Modélisation avec UML- Robert Ogor- mai 2003- www.enst-bretagne.com
- [4] UP: UnifiedProcess - www.Developpez.com
- [5] <http://fr.wikipedia.org/wiki/PHP#Pr.C3.A9sentation>
- [6] http://fr.wikipedia.org/wiki/Hypertext_Markup_Language
- [7] <http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-css-4050/>
- [8] <http://php.developpez.com/telecharger/index/categorie/210/Outils-pour-PHP>
- [9]<http://fr.openclassrooms.com/informatique/cours/prenez-en-main-bootstrap>
- [10] <http://www.commentcamarche.net/contents/702-oracle-introduction-au-sgbd-oracle>

- Table des illustrations -

Figure I.1. Des types de diagramme	7
Figure I.2. Exemple d'un diagramme de séquence	8
Figure I.3. Exemple d'une classe	9
Figure I.4. Exemple des relations d'association.....	9
Figure I.5. Schéma d'itération du processus UP	11
Figure I.6. Le modèle 4+1 vues.....	11
Figure I.7. Exemple d'un diagramme de cas d'utilisation.....	12
Figure I.8. Cycle de vie du processus UP.....	12
Figure II.1. Diagramme de cas d'utilisation.....	15
Figure II.2. Diagramme de séquence « consulter offre»	16
Figure II.3. Diagramme de séquence« réserver offre»	17
Figure II.4. Diagramme de séquence« Annulation de la réservation»	17
Figure II.4. Diagramme de classe	17
Figure III.1. Ecran d'authentification	25
Figure III.2. Ecran du tableau de bord.....	26
Figure III.3. Statistique sous forme de tableau	27
Figure III.4. Statistique sous forme de graphe	27
Figure III.5. Formulaire d'inscription	28
Figure III.6. Ecran de page d'accueil	29
Figure III .7. Ecran de recherche	29

- Résumé -

Dans le cadre de présenter un projet de fin d'étude, des efforts ont été consacrés à la mise en œuvre d'un système qui a pour but de faciliter la gestion d'une agence de voyage. Pour atteindre cet objectif, la modélisation de notre système TMS a été faite par le langage UML en suivant des étapes du processus UP. L'implémentation de l'application a été réalisée sous le langage PHP, Oracle comme SGBD et le WampServer comme outil de développement.

- Abstract -

In the context of presenting a final project study, efforts have been devoted to the implementation of a system to facilitate the management of a travel agency to achieve this goal for the modeling TMS our system was made by following the UML UP steps the process. The implementation of the application was carried out under the PHP, Oracle as DBMS and WampServer language as a tool for development.

ملخص

في إطار تقديم مشروع عن نهاية الدراسة، كانت الجهود المبذولة مخصصة لتنفيذ برنامج مخصص لتسهيل إدارة وكالة سفر؛
فمن أجل دراسة تحليلية باستخدام TMS من أجل تحقيق هذا البرنامج، UML،
بعد هاقمنا بإنشاء قاعدة البيانات باستخدام النظام إدارة قواعد البيانات Oracle؛
وأخير تنفيذ البرنامج بالاعتماد على PHP باستخدام، WampServer .