

RAPPORT DU STAGE DU FIN D'ETUDES

Présenté en vue de l'obtention du
Diplôme National de Licence Appliquée en Sciences et Technologies

Mention : Informatique

Spécialité : Systèmes Informatiques et Logiciels

Par

Mohamed Yosri YAHYAOUI

**Conception et développement d'une application de gestion
d'un tour opérateur**

Encadrant professionnel

Monsieur Aouadi Karim

Encadrant académique

Monsieur El-Ayeb Faycel

Réalisé au sein de Global Web Engineering

Année Universitaire 2014/2015

Encadrant Entreprise

Signature et cachet :

Encadrant ISI

Signature :

RAPPORT DU STAGE DU FIN D'ETUDES

Présenté en vue de l'obtention du
Diplôme National de Licence Appliquée en Sciences et Technologies

Mention : Informatique

Spécialité : Systèmes Informatiques et Logiciels

Par

Mohamed Yosri YAHYAOU

**Conception et développement d'une application de gestion
d'un tour opérateur**

**Encadrant professionnel
Encadrant académique**

**Monsieur Aouadi Karim
Monsieur El-Ayeb Faycel**

Réalisé au sein de Global Web Engineering

Année Universitaire 2014/2015

Remerciements

J'adresse mes remerciements aux personnes qui m'ont aidé dans la réalisation de ce projet.

En premier lieu, je remercie Mr. Karim Aouadi, encadreur de la société Global Web Engineering, qui m'a aidé dans les difficiles tâches, et pour le temps qu'ils m'ont consacré.

En deuxième lieu je remercie Mr. El Ayel Faycel, encadreur de l'Institut Supérieur de l'Informatique, qui m'a donné de son temps pour accomplir la réalisation de ce projet, et pour ces conseils qui ont de plusieurs intérêts pour moi.

Je tiens aussi à remercier tous les membres du jury pour avoir accepté de juger mon travail.

Enfin, je tiens à exprimer mon amitié et mon respect profonds envers tous mes collègues de l'ISI.

Dédicace

*À mes très chers parents qui sont toujours là pour moi tout au long
de mes études et qui m'ont donné un magnifique modèle de labeur et
de persévérance.*

À mon frère Mahdi pour son soutien moral.

*À mes amis Mzoughi Ahmed, Aycub Habibi et Ayachi Sadok
par leurs encouragements.*

À tous mes amis dans la jeune chambre internationale de Tunis.

Liste des figures

Figure 1: Diagramme de cas d'utilisation globale.....	15
Figure 2:Diagramme de cas d'utilisation détaillé du client.....	16
Figure 3:Diagramme de cas d'utilisation détaillé du fournisseur.....	17
Figure 4 :Diagramme de cas d'utilisation détaillé de l'administrateur.....	18
Figure 5:Diagramme de cas d'utilisation pour raffiner la fonctionnalité de la présentation des données du client.....	18
Figure 6:Diagramme de cas d'utilisation pour affiner l'achat du client	19
Figure 7:Diagramme de cas d'utilisation pour manipuler les données client.....	19
Figure 8:Diagramme de cas d'utilisation concernant les paiements des factures.....	20
Figure 9:Diagramme de cas d'utilisation pour raffiner la fonctionnalité du gestion des réservations.....	20
Figure 10:Diagramme de cas d'utilisation pour raffiner la fonctionnalité du gestion des factures avec les clients.....	21
Figure 11:Diagramme de cas d'utilisation pour raffiner la fonctionnalité du gestion des prestations des fournisseurs.....	22
Figure 12:Diagramme de cas d'utilisation pour raffiner la fonctionnalité du gestion des encaissements, soldes dus et relances.....	23
Figure 13:Diagramme de cas d'utilisation pour raffiner la fonctionnalité gestion de la caisse.....	24
Figure 16:Diagramme de séquence pour la gestion des enquêtes.....	27
Figure 17:Diagramme de cas d'utilisation pour raffiner la fonctionnalité pour la gestion des factures des fournisseurs.....	28
Figure 18:Diagramme de cas d'utilisation pour la gestion des statistiques et du tableaux de bord.....	28
Figure 19 :Diagramme de séquence pour la gestion des clients.....	29
Figure 20 : Diagramme de séquence pour la réalisation des achats.....	29
Figure 21 :Diagramme de séquence pour les paiements des factures clients.....	30
Figure 22 : Diagramme de séquence pour la gestion des prestations des fournisseurs.	30
Figure 23 : Diagramme de séquence pour la gestion des prestations des fournisseurs.	30
Figure 24 : Positionnement de Java Entreprise Edition.....	37
Figure 25 : Liaison entre une application Java et MySQL.....	38
Figure 26 : Menu de WampServer.....	39
Figure 27 : Diagramme de paquetage.....	33
Figure 28 : Contenu de fichier glassfish-web.xml(1).....	47
Figure 29 : Contenu de fichier glassfish-web.xml(2).....	47
Figure 30 : Présentation d'hibernate.....	49
Figure 31 : Le contenu de fichier hibernate.cfg.xml.....	50
Figure 32 : Configuration de mapping d'hibernate.....	50
Figure 33 : Les fichiers DAO générés par hibernate.....	50
Figure 34 : Contenu du web.xml (1).....	52
Figure 35 : Contenu du web.xml (2).....	52

Figure 36 : Capture sur les fichiers Git.....	55
Figure 37 : Exemple d'une classe de test unitaire.....	56
Figure 38 : Serveurs de Google.....	56
Figure 39 : Existence de plugin GAE dans Eclipse.....	57
Figure 40 : Identifiant accès à l'application.....	58
Figure 41 : Déploiement de cette application.....	58
Figure 42 : Erreurs survenus lors de déploiement.....	58
Figure 43 : Dispositif webmarketing complet.....	59
Figure 44 : Test d'indexation d'un site Internet.....	58
Figure 45 : Schéma de stratégie de community manager.....	60
Figure 46 : Exemple d'un tableau de pilotage.....	60
Figure 47 : Logo de la solution MailChimp.....	61
Figure 48 : Page d'authentification.....	57
Figure 49 : Type des acteurs dans l'application.....	57
Figure 50 : Formulaire d'inscription pour les clients.....	58
Figure 51 : Formulaire pour les habitudes des voyageurs.....	59
Figure 52 : Page d'accueil pour les clients.....	59
Figure 53 : Offres proposées.....	60

Liste des tableaux

Tableau 1 : Différence entre une application web et une application desktop.....	64
Tableau 2 : Liste des versions d'Eclipse.....	39
Tableau 3 : Les outils proposés par Eclipse.....	39
Tableau 4 : Liste des IDE correspond au Glassfish.....	45

Sommaire

Remerciements.....	5
Dédicace.....	6
Liste des figures.....	7
Liste des tableaux.....	9
Introduction générale.....	12
Chapitre 1 : Analyse et spécification des besoins.....	14
1.1. Introduction.....	14
1.2. Cadre de stage.....	14
1.3. Présentation de GWE.....	14
1.4. Sujet de stage.....	15
1.5. Etude de l'existant.....	15
1.6. Difficultés liées aux spécifications des besoins.....	16
1.7. Avantages et inconvénients.....	16
1.8. Solution proposée.....	16
1.9. Modèle de cycle de vie d'une application.....	17
1.10. Diagramme de Gantt théorique.....	17
1.11. Les besoins fonctionnels et non fonctionnels.....	17
1.12. conclusion.....	18
Chapitre 2 : Conception.....	20
2.1 Introduction.....	20
2.2 Présentation d'UML.....	20
2.3 Présentation de Power AMC Designer.....	21
2.4 Use case globale.....	21
2.5 Use case détaillé.....	22
2.5.1 Use case client.....	22

2.5.2 Use case fournisseur.....	23
2.5.3 Use case Administrateur.....	23
2.6 Raffinement de la conception.....	24
2.7 Diagramme des séquences.....	30
2.8 Diagrammes des classes.....	34
2.9 Diagramme de package.....	36
2.10 Patrons de conception.....	37
2.11 Conclusion.....	37
Chapitre 3 : Réalisation et présentation des interfaces.....	38
3.1 Introduction.....	38
3.2 Environnement de travail.....	38
3.2.1 Eclipse.....	38
3.2.2 Mysql.....	41
3.2.3 WampServer.....	42
3.2.4 Glassfish.....	43
3.2.5 Framework Hibernate	46
3.2.6 Jsp, jsf, servlets.....	49
3.2.7 EGit.....	52
3.2.8 Test unitaire : JUnit.....	54
3.2.9 Essai avec Cloud Google.....	54
3.3 Présentation des interfaces.....	57
3.4 Diagramme de Gantt réel.....	59
3.5 Introduction.....	59
Chapite 4 : Dépolement et Webmarketing.....	62
4.1 Introduction.....	62
4.2 Webmarketing.....	62
4.3 Référencement.....	63
4.4 Community management.....	65
4.5 E-mailing.....	66
4.6 Conclusion.....	66
Conclusion générale.....	67
Annexe.....	68

Bibliographie et nethographie.....	70
------------------------------------	----

Introduction générale

Dans l'optique de poursuivre mes études, dans le domaine des systèmes informatique et logiciels dans l'Institut Supérieure de l'Informatique (ISI Ariana) et en entrant dans la vie professionnelle, j'ai eu l'opportunité de s'intégrer dans l'entreprise **Global Web Engineering**, pour préparer le projet de fin d'études.

Je suis passionné depuis longtemps par le domaine du web avec son explosion dans les dernières années et ses applications informatique. Pour cela, le sujet qui m'a été proposé, de la part de cette entreprise, est une occasion pour mettre en œuvre mes capacités.

Les objectifs de ce stage sont les suivants :

- Découvrir la vie professionnelle.
- Utiliser mes capacités et les enrichir.
- Présenter et promouvoir les réalisations effectuées et identifier des pistes de réflexion.

On a donc intégré dans cette entreprise pour préparer un projet qui consiste à concevoir et développer une application web pour la gestion d'un tour opérateur.

On a eu la mission de préciser les fonctionnalités pour gérer un tour opérateur comme : la gestion des données (fournisseurs, clients, quelques contact,...), intégration du système décisionnelle, gestion des factures, e-mailing et d'autres fonctionnalités qu'on les précisera dans la suite.

Dans un premier temps, on va analyser les besoins concernant cette application, en indiquant : les besoins fonctionnels et non fonctionnels, une étude de l'existant et des solutions pour la réalisation. On exploite cette partie pour présenter l'entreprise, le cadre du stage et décrire de façon détaillée le sujet.

Ensuite, on va décrire la phase de conception, en présentant des diagrammes du langage UML comme : les diagrammes de cas d'utilisation générale et détaillé en précisant les acteurs impliqués dans cette application, le diagramme de classes, les diagrammes des séquences, diagramme de package, etc.

Enfin, on va décrire la phase de la réalisation et du codage de l'application en indiquant les outils informatiques utilisés tout au long du développement. On va présenter les interfaces inclus dans l'application et on va indiquer des techniques pour réaliser la Ereputation de cette application.

Chapitre 1

Analyse et spécification des besoins

Chapitre 1

Analyse et spécification des besoins

1.1. Introduction

Il s'agit d'une étape cruciale dans la réalisation d'une application donnée. Le futur de logiciel dépend beaucoup de cette phase, elle nous permet le développement d'une application suffisante. Pour cela, le client et le développeur doivent être en étroite relation, voire avoir un intermédiaire entre eux s'il le faut.

Les développeurs doivent transformer les idées floues en une spécification précise de besoins, souhaits et exigences exprimés par une communauté d'utilisateurs. Ainsi, ils définies une relation entre un système et son environnement.

Pour arriver à nos objectifs, il nous faut prendre connaissance de :

- L'étude de la faisabilité : domaine de l'application, l'état actuel de l'environnement du futur système, les ressources, etc.
- L'analyse et la définition des besoins : permet de trouver un commun accord entre les spécialistes et les utilisateurs.

1.2 Cadre de stage

C'est un stage de 4 mois, entre 01/02/2015 à 30/05/2015, à effectuer dans une boîte de développement.

On a pris la mission de développer une application web pour gérer un tour opérateur.

1.3 Présentation de Global Web Engineering

Global Web Engineering (GWE) est une société d'ingénierie informatique en Tunisie. Il offre divers services aux entreprises et aux particuliers dans le domaine de développement web telque:

- Site vitrine, site clé en main, site Joomla.
- Développement des sites par des systèmes de gestion du contenu CMS comme Wordpress, Drupal, etc.
- Création des composants et modules web, installation des scripts, migration et refonte des sites web.
- Développement des sites spécifiques (HTML, CSS, PHP).

- Conception et développement des applications web en intégrant une solution d'entreprise comme le Business Intelligence, ERP, CRM, etc.
- Développement des applications de e-commerce, des applications de la gestion, etc.
- E-mailing, référencement, gestion de la communauté (webmarketing).

1.4 Sujet de stage

Le sujet de stage initial tel qu'il a été défini dans la convention de stage est l'intitulé: « Conception et développement d'une application JEE de gestion d'un tour opérateur (voir l'annexe) ».

1.5 Etude de l'existant

En bénéficiant de l'explosion des technologies du web, les développeurs ne cessent de conquérir ce domaine pour satisfaire la communauté en offrant des solutions aux divers problèmes quotidiens, dont la construction des applications web comme une solution d'entreprise pour la gestion d'un tour opérateur.

Notre application montre son importance par la couverture des nombreux domaines comme:

- Domaine de la gestion du système d'information.
- Domaine du marketing par l'intégration du système du business intelligence(BI).
- Domaine de la comptabilité en intégrant la gestion des factures et les ventes réalisées par le tour opérateur.

Une autre importance dans notre application est qu'elle peut être hébergée à distance par l'utilisation des réseaux comme l'Internet, Intranet (application web interne), extranet où on peut utiliser la technologie du Cloud Computing (voir l'annexe). La technologie utilisée pour héberger l'application se diffère suivant le besoin du client.

Comme tout logiciel, on doit faire face à deux types de complexité :

1. Complexité **essentielle** : peut être dégagée au niveau de la spécification des besoins, celle du problème à informatiser..
2. Complexité **accidentelle**: désigne la complexité introduite dans des programmes informatiques. Ceci est rencontré à la phase de la réalisation et du codage. On trouve des exemples dans ce type de complexité comme : le choix du langage, du middleware, du système d'exploitation, etc.[1]

Les applications traitant ce problème ne sont pas assez variées et nombreuses. Parmi les applications trouvées du ce type, elles utilisent des technologies traditionnelles et un peu critique.

D'une part, elles se basent sur des applications desktop (local) qui n'exploitent pas les avantages proposés par les technologies du web récents. D'autre part, les applications réalisées pour gérer un tour opérateur ne sont pas innovantes, car ils mettent des fonctionnalités traditionnelles comme la gestion des informations. Ils n'intègrent pas des autres solutions telles que le business intelligence (voir l'annexe) ou la technologie du Cloud computing. D'autant plus, la majorité des applications de ce genre utilisent des produits qui sont propriétaires non gratuits. C'est dans ce but qu'on va développer une application complète et fonctionnelle, en utilisant des produits gratuits et open source (Exemple : Eclipse). En outre, On doit innover des autres fonctionnalités. Parmi les meilleures applications trouvées en 2015 : TrekkSoft, TourWriter, WebResManager, PeekPro, etc.

1.6 Difficultés liées aux spécifications des besoins

On a trouvé des difficultés liées à la compréhension de quelques fonctionnalités proposées dans le cahier de charge. En effet, on trouve des mots spécifiques au de domaine de marketing, de comptabilité et de gestion. Alors, on a consacré plus du temps pour bien comprendre et bien analyser les besoins.

1.7 Avantages et inconvénients

Les applications de voyagistes trouvés on des avantages et inconvénients :

Avantages:

- Existence des applications sur le web.
- Implication des fonctionnalités principales pour gérer un tour opérateur.

Inconvénients:

- Intégration d'un système décisionnelle.

1.8 Solutions proposées

Après une étude approfondie de l'existant, plusieurs limites des applications de gestion du tour opérateur présente sur scène ont surgi. Il manque ainsi une application innovante avec l'utilisation des outils open source.

De ce fait, on propose de concevoir et de réaliser une application web de la gestion du tour opérateur avec l'intégration d'un système du business intelligence et l'ajout de quelques fonctionnalités pour gérer l'entreprise.

Dans cette application, on va exploiter la nouvelle technologie massivement utilisée de nos jours, c'est le Cloud computing par l'utilisation de la plateforme du Cloud offerte par Google est le **Google App Engine** ([voir l'annexe](#)).

Tout au long du développement de l'application, on va utiliser que des outils gratuits et open source comme : Eclipse, Serveur Tomcat, etc.

1.9 Modèle de cycle de vie d'une application

On a utilisé le modèle de vie pour cette application est le modèle en V car il demeure actuellement le cycle de vie le plus connu et certainement le plus utilisé.

1.10 Diagramme de Gantt théorique

Le diagramme de Gantt théorique qui définit la période de chaque phase : spécification des besoins, conception et réalisation.

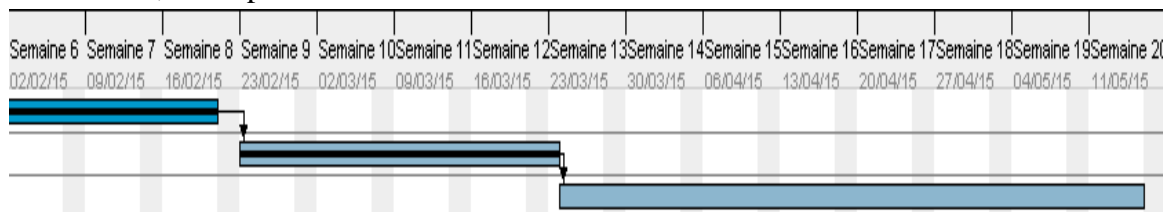


Figure 54 : Diagramme de Gantt théorique

1.11 Les Besoins fonctionnelles et les besoins non fonctionnelles

Les Besoins fonctionnelles :

Il s'agit des fonctionnalités à assurer par l'application. Ce sont les besoins spécifiant le comportement d'entrée/ sortie. L'application doit permettre de:

- Gérer des données concernant les clients et/ou prospects (adresse, nom, prénom, identifiant unique, contact, etc.).
- Gérer l'historique des relations commerciales.
- Gérer des habitudes de voyage.
- Gérer des données concernant les fournisseurs en précisant leurs adresses, type des fournisseurs (TO, hôtels, réceptifs,...).
- Gérer et suivre des devis et ventes.
- Gérer des encaissements, soldes dus et relances.
- Gérer des réservations fournisseurs, contrôles des achats et marges.
- Gérer des factures (Saisir des factures, suivi des échéances, ...).
- Gérer les informations sur les banques (adresse, numéro du compte,...).
- Gérer des enquêtes de satisfaction.
- Intégrer d'un système de la newsletter (Emailing).
- Gérer les remises.
- Editer des documents (Excel, PDF,...).
- Editer des tableaux du bord.
- Editer des statistiques.
- Gérer les paiements des factures (partielle ou totale), ventilations sur les dossiers clients, vérification des marges,...

- Virer entre différents comptes.
- Suivre des opérations de caisse.

Les Besoins non fonctionnelles :

Il s'agit des fonctionnalités qui caractérisent le système. Se sont des besoins liés à la performance et le type de conception. Ces besoins peuvent concerner les contraintes d'implémentation.

Dans le cadre de ce travail, l'application devra être extensible, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.

Cette application doit être capable de :

- Adopter d'une interface du paramétrage conviviale.
- Produire des sauvegardes régulières.
- Faciliter la maintenance dans la conception et l'implémentation, par l'utilisation des modèles de conception (Design Pattern).
- Gérer des comptes pour les acteurs qui interagissent avec l'application.

1.12 Conclusion

L'étude préalable appelée ingénierie des exigences ou analyse et spécification des besoins, constitue une phase capitale, toute la suite du projet dépend d'elle, elle doit être faite avec beaucoup de rigueur et plus d'attention pour que le projet réussit avec un grand succès.

Dans ce chapitre, on a analysé l'existant, puis on a fait les critiques, on a proposé une approche de solution qui consiste à concevoir et à développer une application qui facilitera les services énumérés précédemment. Enfin, on a lister les différents besoins fonctionnelles et les besoins non fonctionnelles.

Chapitre 2

Conception

Chapitre 2

Conception

2.1. Introduction

La plupart des langages sont orientés objets. Le passage de la programmation fonctionnelle à l'orienté objet n'est pas facile. L'un de souci était d'avoir une idée globale en avance de ce qu'on doit programmer.

L'algorithmique qui était utilisé dans la programmation fonctionnelle ne pourrait pas suffire à lui seul. Le besoin d'avoir des méthodes ou langages pour la modélisation des langages orientés objet est nécessaire. Ainsi plusieurs méthodes ou langages ont vu le jour. En occurrence UML qui nous a permis de faire la conception de notre application.

De nos jours, UML2 possède treize diagrammes qui sont classés en deux catégories (dynamique et statique).

Pour ce faire, on a commencé par les diagrammes de cas d'utilisation (Use Case) qui permettent de donner une vue globale de l'application. Pas seulement pour un client non avisé qui aura l'idée de sa future application mais aussi pour le développeur qui s'en sert pour le développement des interfaces.

En deuxième lieu, on fait raffiner la conception en présentant la chronologie des opérations par les diagrammes des séquences.

Enfin, on présente les diagrammes statiques qui sont ceux des classes et la modélisation des packages.

2.2. Présentation UML

En informatique **UML** (de l'anglais *Unified Modeling Language*), ou Langage de modélisation unifié, est un langage de modélisation graphique à base de pictogrammes. Il est utilisé en développement logiciel, et en conception orientée objet. UML est couramment utilisé dans les projets logiciels.

UML est l'accomplissement de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard défini par l'Object Management Group (OMG). La dernière version diffusée par l'OMG est UML 2.5 bêta 2 depuis septembre 2013.

2.3. Présentation du Power AMC Designer

PowerAMC propose différentes techniques de modélisation, chacune accessible aux informaticiens de tout niveau, parmi elles : Merise, UML, Data Warehouse, et processus métiers. Simple d'utilisation, personnalisable et dotée d'une interface intuitive, cette application optimise les productivités individuelle et collective. Elle intègre en outre des fonctions de génération de code pour plus de 45 bases de données et divers langages de programmation.

2.4. Use case global

Dans cette partie, on va préciser les fonctionnalités dans un diagramme global de cas d'utilisation.

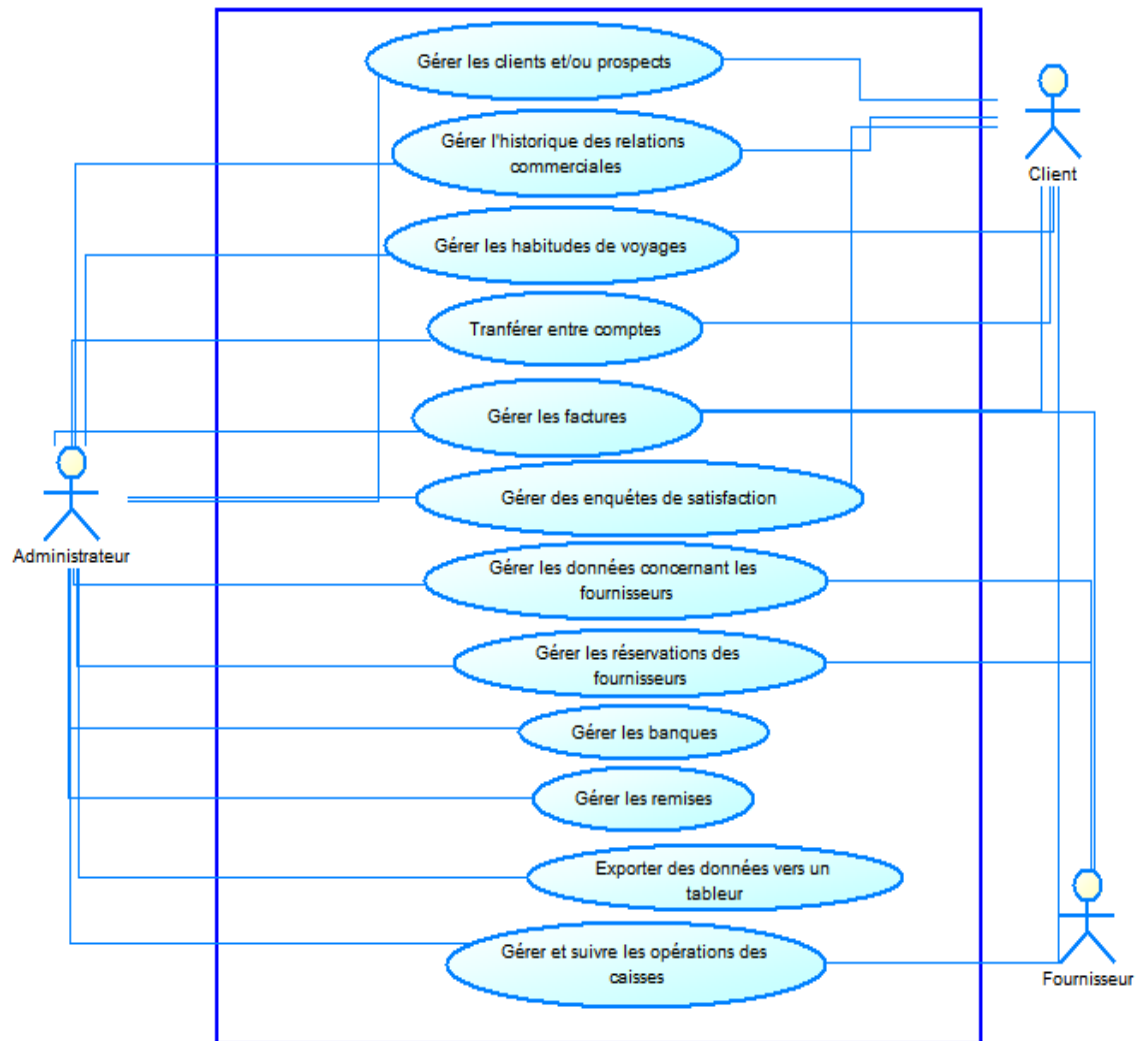


Figure 2:Diagramme de cas d'utilisation globale

2.5. Use case détaillé

Dans cette partie, on va faire préciser les différentes fonctionnalités pour chaque acteur impliqué dans le système.

2.5.1. Use case client

Dans ce diagramme, on présente de façon détaillée les fonctionnalités assurées par le client.

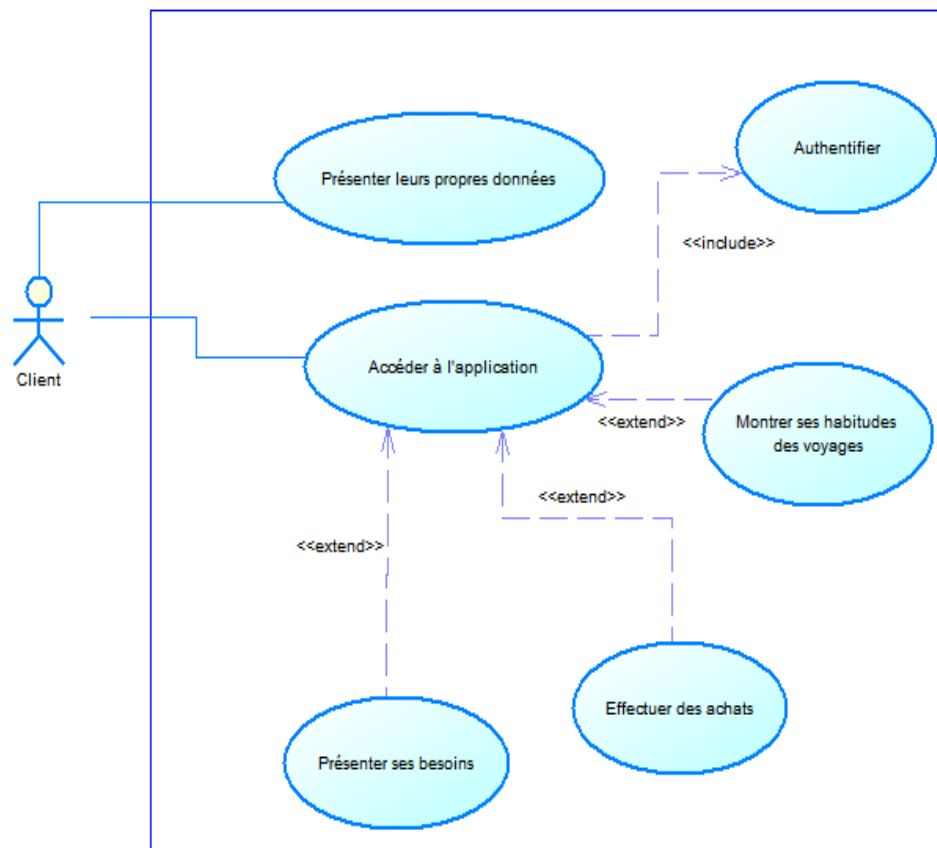


Figure 2:Diagramme de cas détaillé du client

2.5.2. Use case fournisseur

Dans ce diagramme de cas d'utilisation, on va lister les fonctionnalités assurées par le fournisseur.

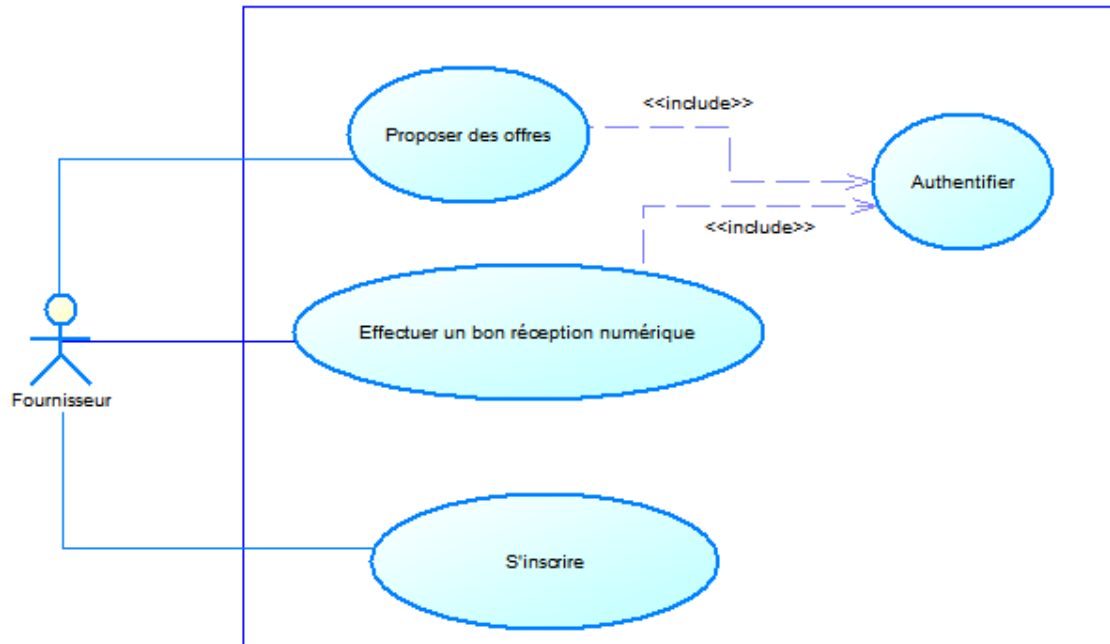


Figure 3:Diagramme de cas d'utilisation détaillé du fournisseur

2.5.3. Use case administrateur

Dans ce diagramme, on va citer les fonctionnalités de la part d'administrateur.

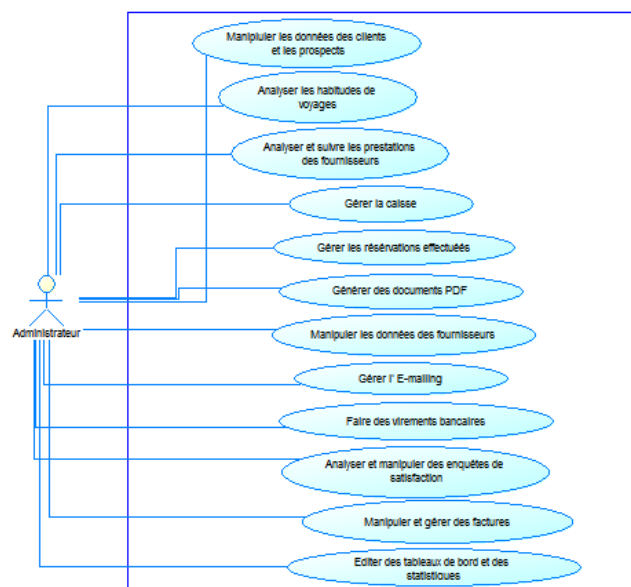


Figure 4: Diagramme de cas d'utilisation détaillée de l'administrateur

2.6 Raffinement de la conception

-Raffinement de la présentation des données concernant le client :

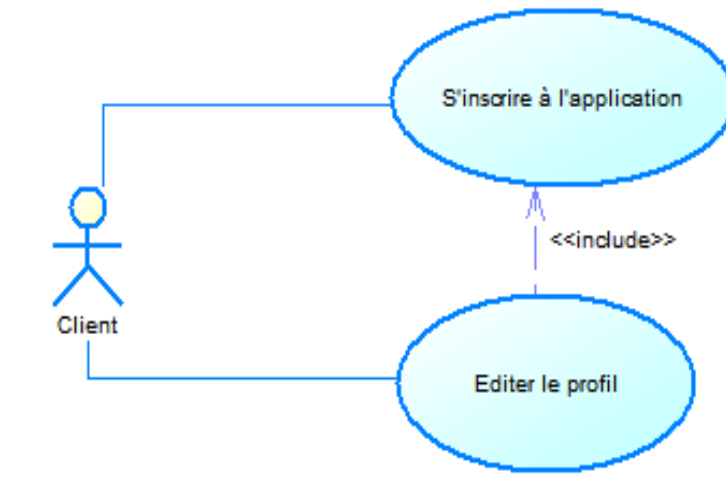


Figure 5:Diagramme de cas d'utilisation pour raffiner la fonctionnalité de la présentation des données des clients

-Raffinement des achats réalisé par le client

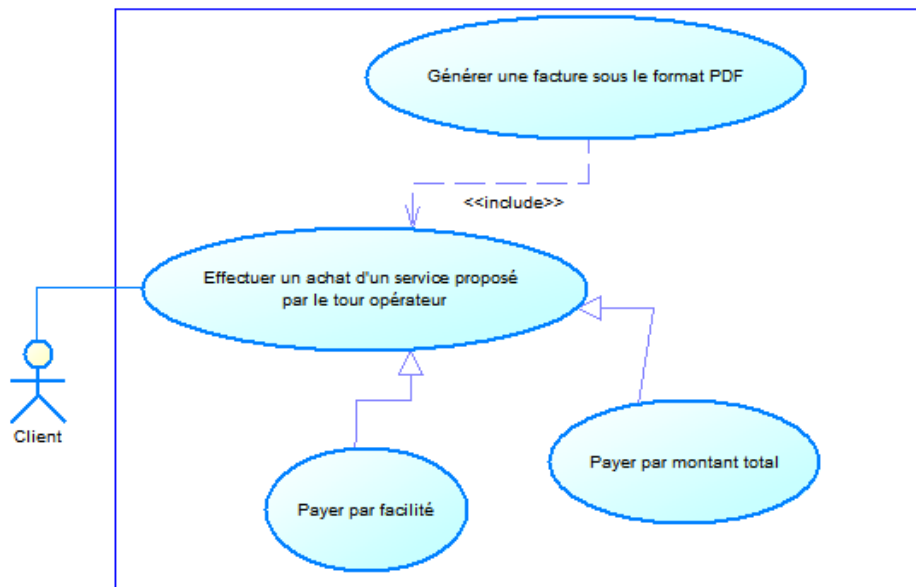


Figure 6:Diagramme de cas d'utilisation pour raffiner la fonctionnalité de l'achat du client

-Manipulations des données client :

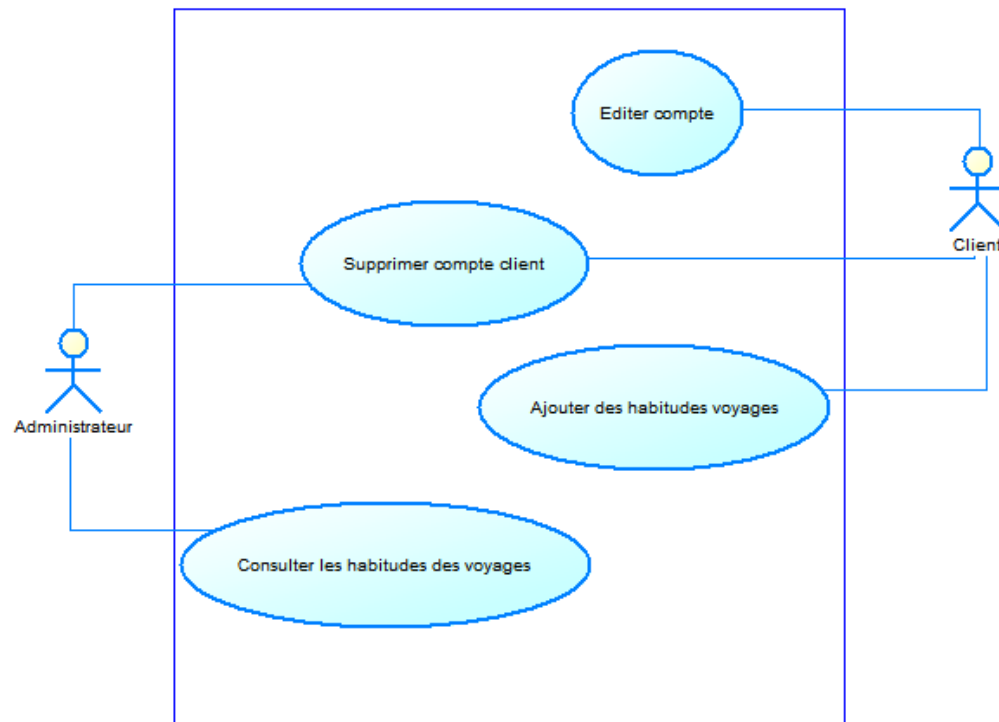


Figure 7:Diagramme de cas d'utilisation pour manipuler des données client

-Paieiment des factures :

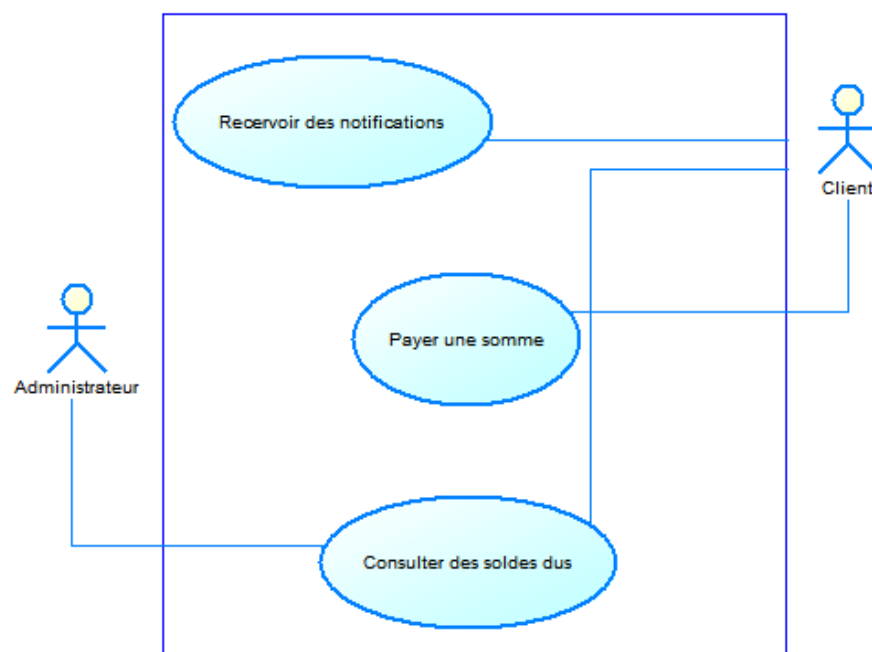


Figure 8:Diagramme de cas d'utilisation pour payer des factures

-Raffinement de la fonctionnalité de gestion des réservations :

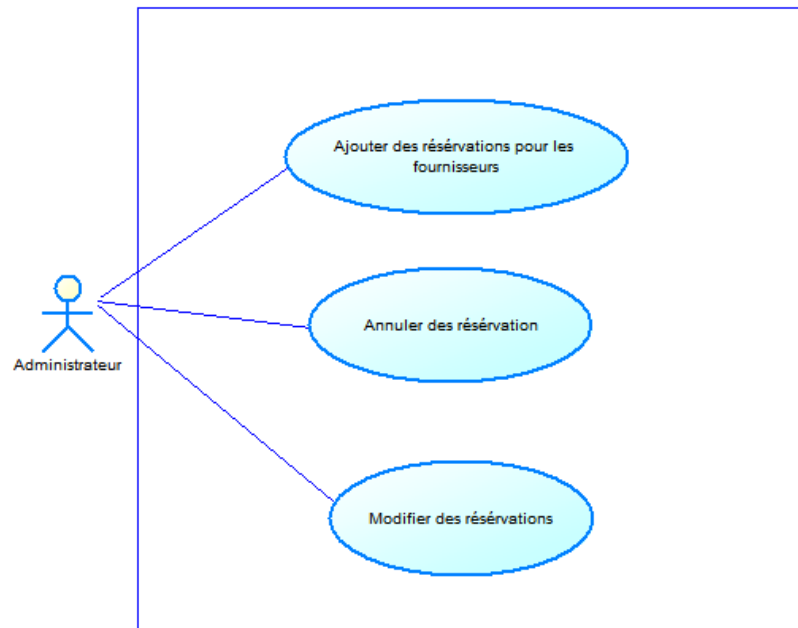


Figure 9:Diagramme de cas d'utilisation pour raffiner la fonctionnalité de la gestion des réservations

-Raffinement de la fonctionnalité de la gestion des factures avec les clients:

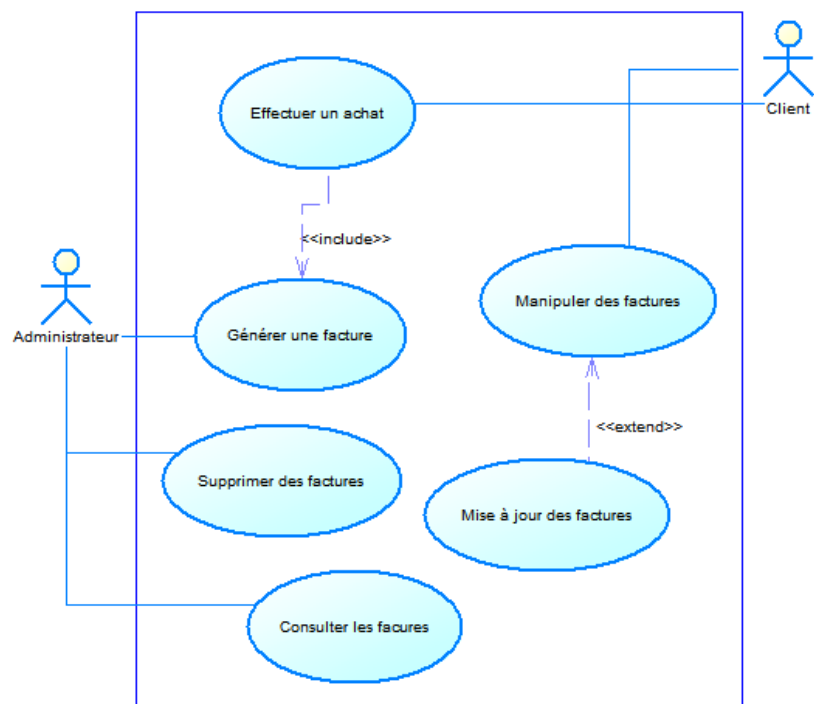


Figure 10:Diagramme de cas d'utilisation pour raffiner la fonctionnalité de la gestion des contrats

-Raffinement sur la fonctionnalité des prestations des fournisseurs :

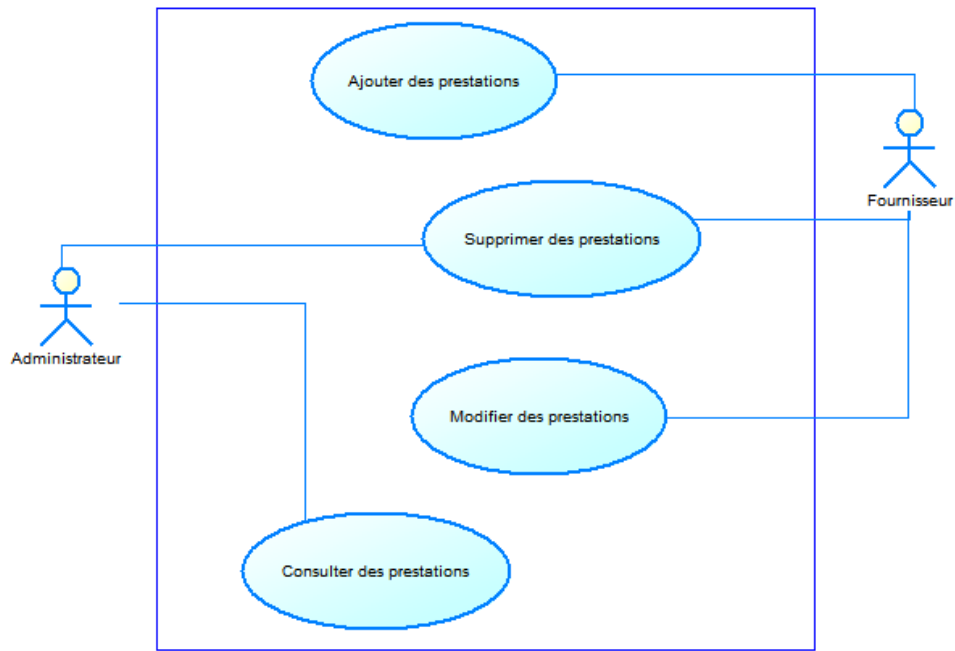


Figure 11:Diagramme de cas d'utilisation pour raffiner la fonctionnalité des prestations des fournisseurs

-Raffinement sur la fonctionnalité du la gestion des encaissements, soldes dus, et relances:

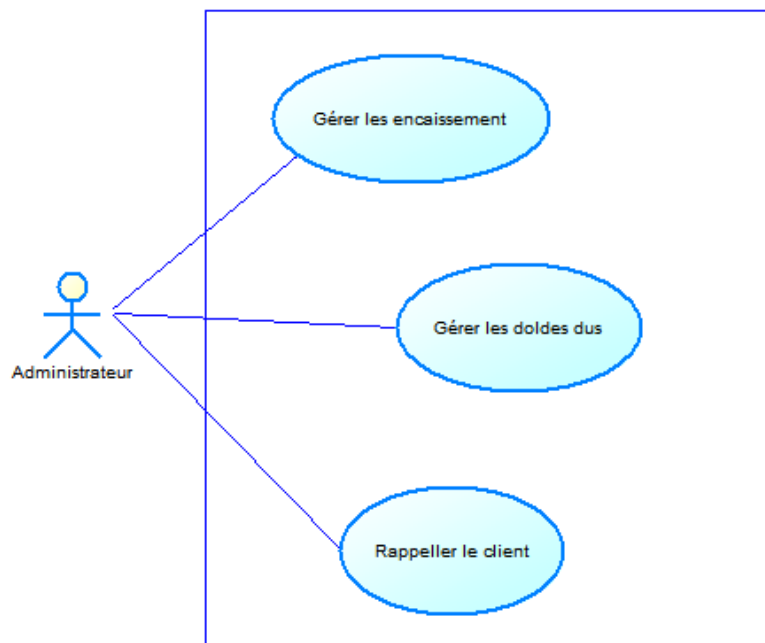


Figure 12:Diagramme de cas d'utilisation pour raffiner la fonctionnalité de la gestion de l'encaissement, soldes dus et relances

-Raffinement sur la fonctionnalité de la gestion de la caisse:

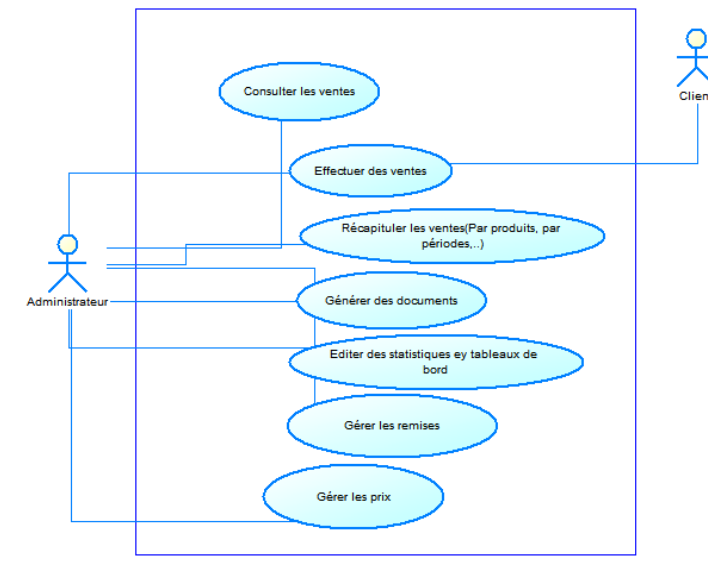


Figure 13:Diagramme de cas d'utilisation pour raffiner la fonctionnalité de la gestion de la caisse

-Raffinement sur la fonctionnalité de la gestion des documents :

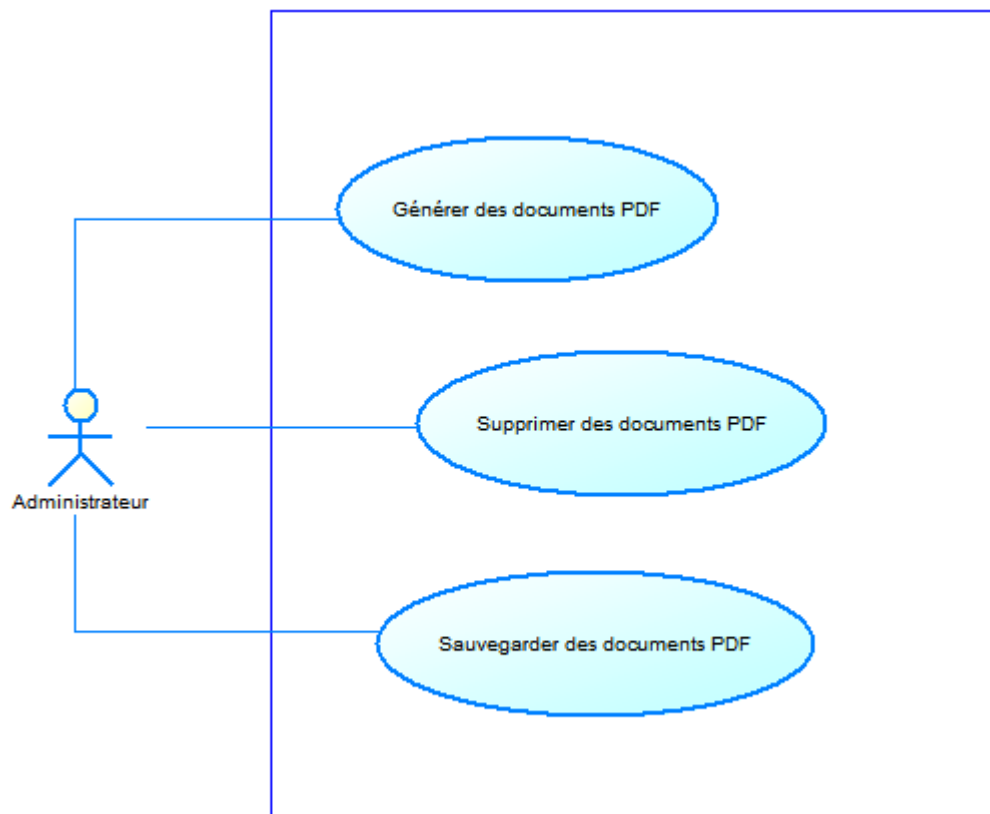


Figure 14:Diagramme de cas d'utilisation pour raffiner la fonctionnalité de la gestion des documents

-Raffinement sur la fonctionnalité de la gestion d'e-mailing :

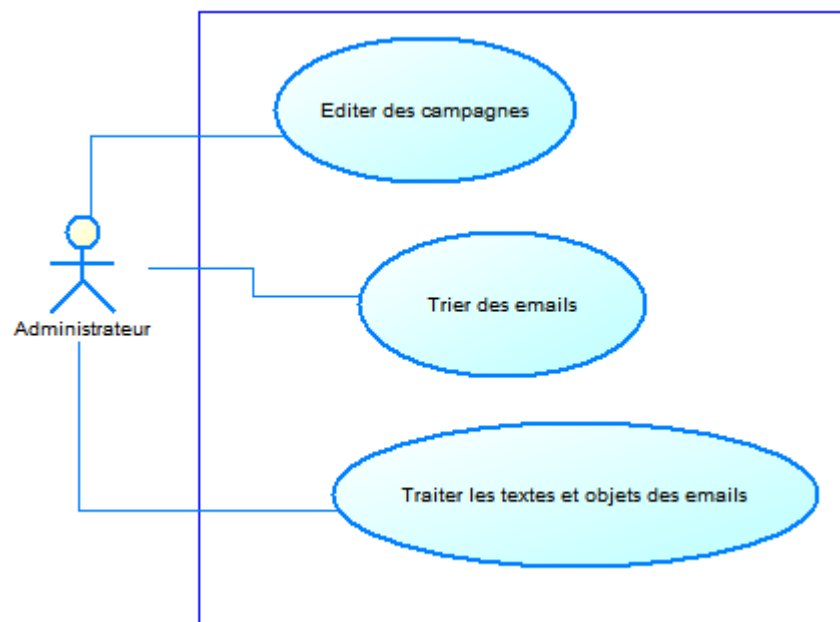


Figure 15:Diagramme de cas d'utilisation pour raffiner la fonctionnalité de la gestion des documents

-Raffinement sur la fonctionnalité de la gestion des enquêtes de satisfactions :

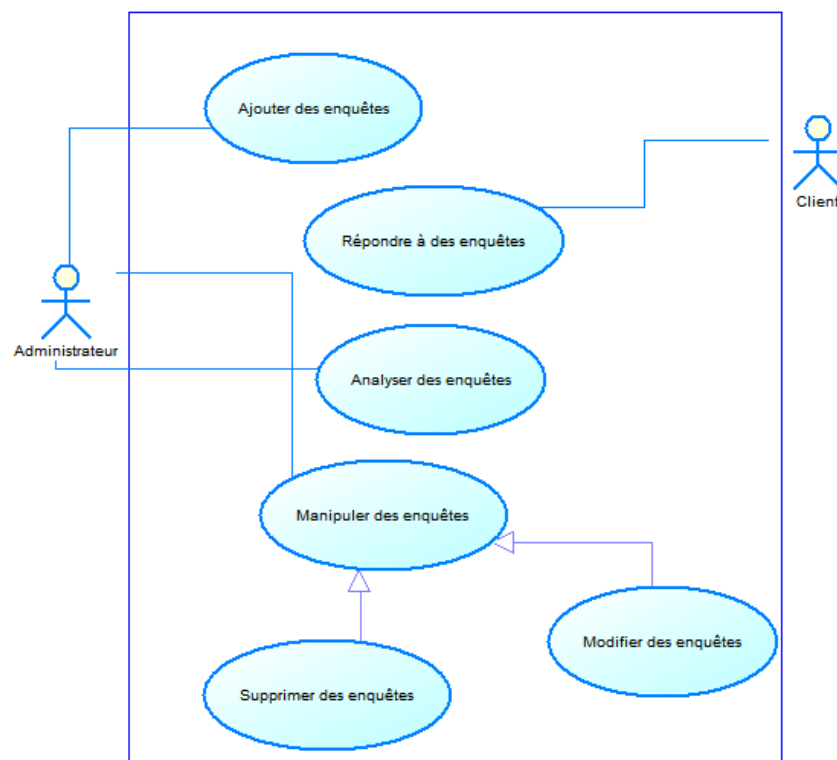


Figure 16:Diagramme de cas d'utilisation pour raffiner la fonctionnalité de la gestion des enquêtes

-Raffinement sur la fonctionnalité de la gestion des factures des fournisseurs :

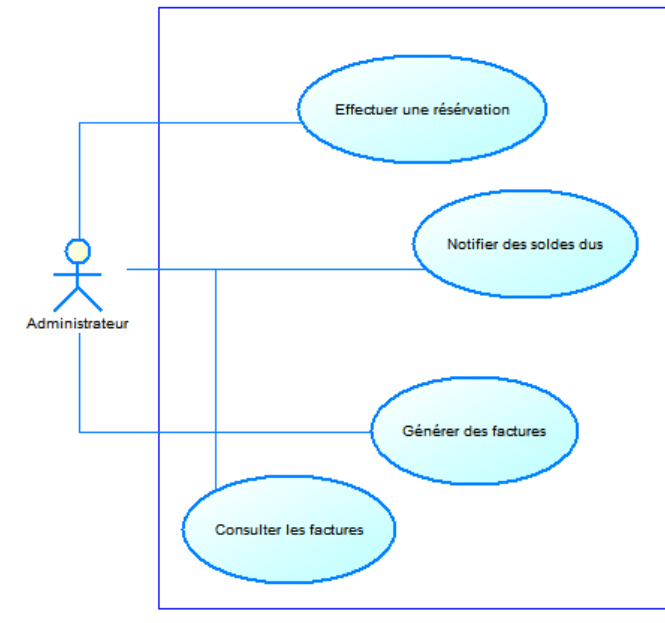


Figure 17:Diagramme de cas d'utilisation pour raffiner la fonctionnalité de la gestion de la facture des fournisseurs

-Raffinement sur la fonctionnalité de l'édition des tableaux de bord et des statistiques :

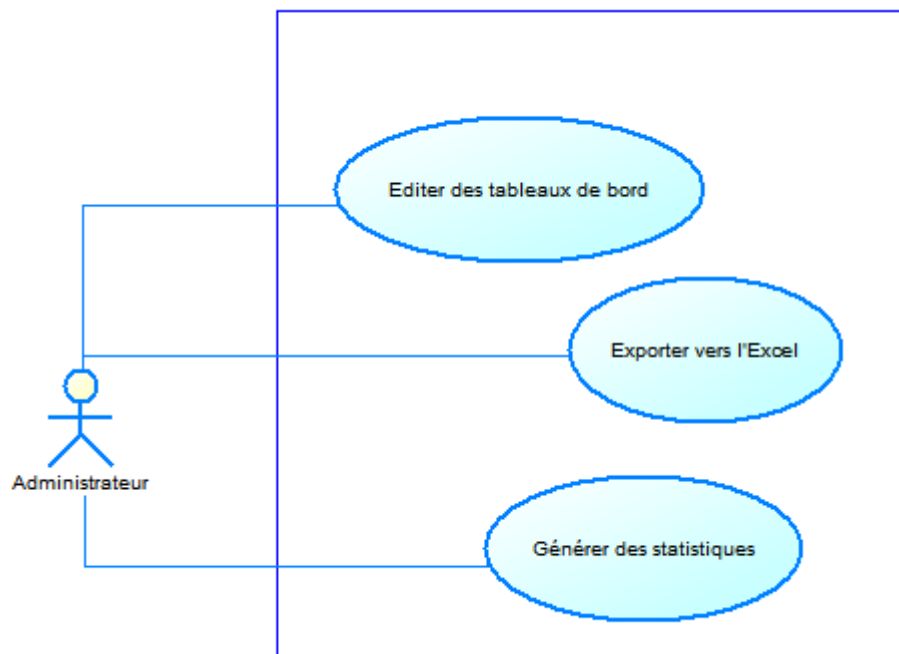


Figure 18:Diagramme de cas d'utilisation pour raffiner la fonctionnalité de l'édition des statistiques et des tableaux de bord

2.7. Les Diagrammes des séquences

Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages.

On n'y décrit pas le contexte ou l'état des objets, la représentation se concentre sur l'expression des interactions.

Les diagrammes de séquences peuvent servir à illustrer un cas d'utilisation. L'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical du diagramme, le temps s'écoule "de haut en bas" de cet axe.

Diagramme de séquence pour la gestion des clients :

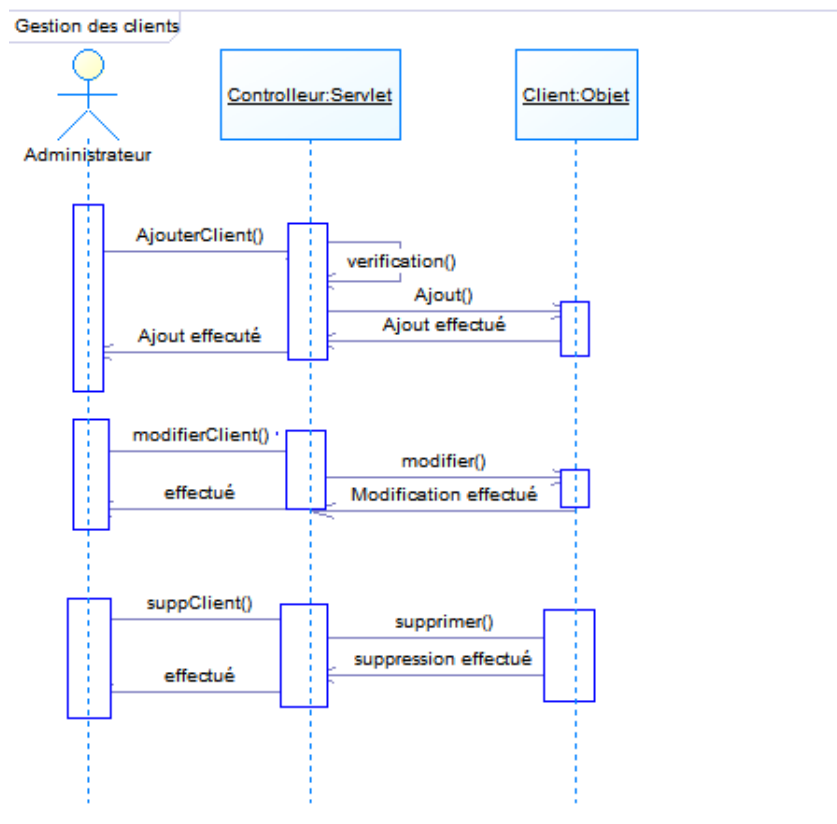


Figure 19:Diagramme de séquence pour la gestion des clients

Diagramme de séquence pour la réalisation des achats:

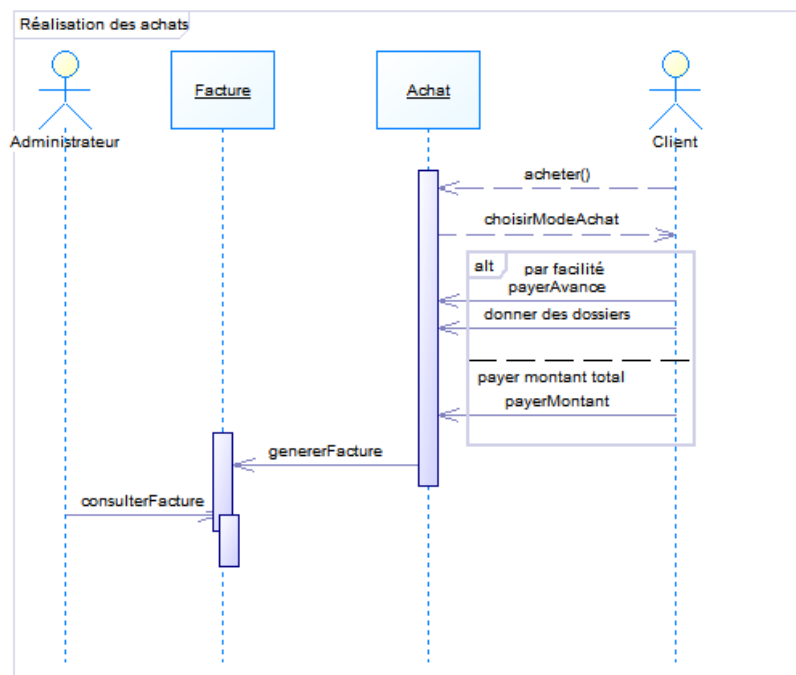


Figure 20:Diagramme de séquence pour la réalisation des achats

Diagramme de séquence pour le paiement des factures :

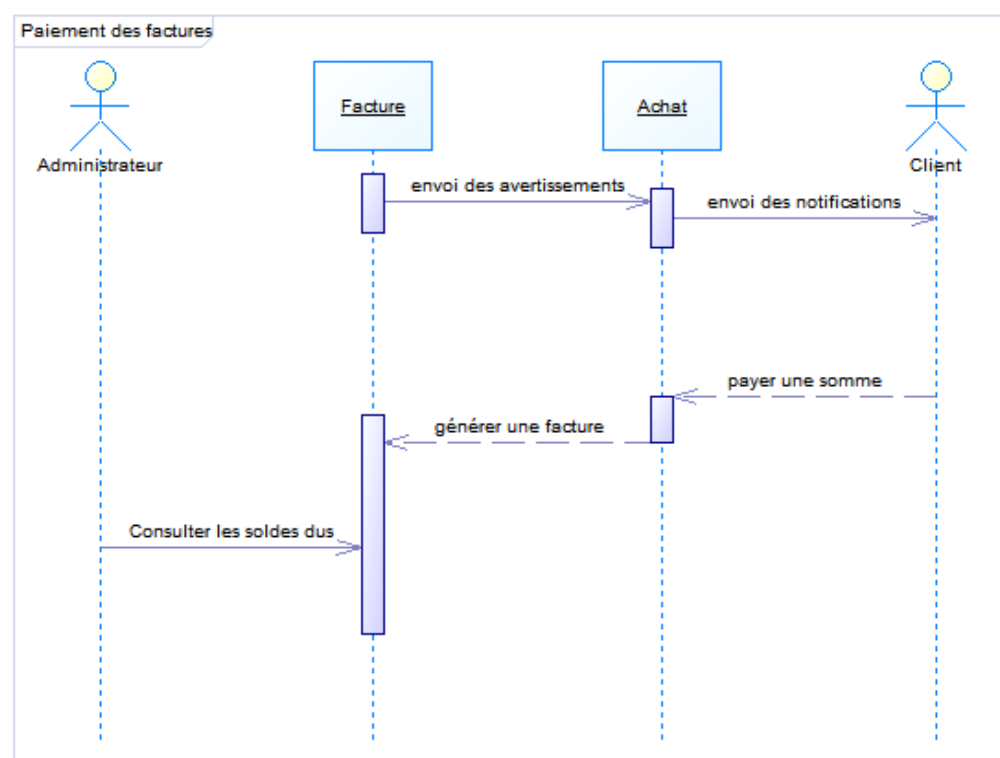


Figure 21:Diagramme de séquence pour les paiements des factures

Diagramme de séquence pour les prestations des fournisseurs

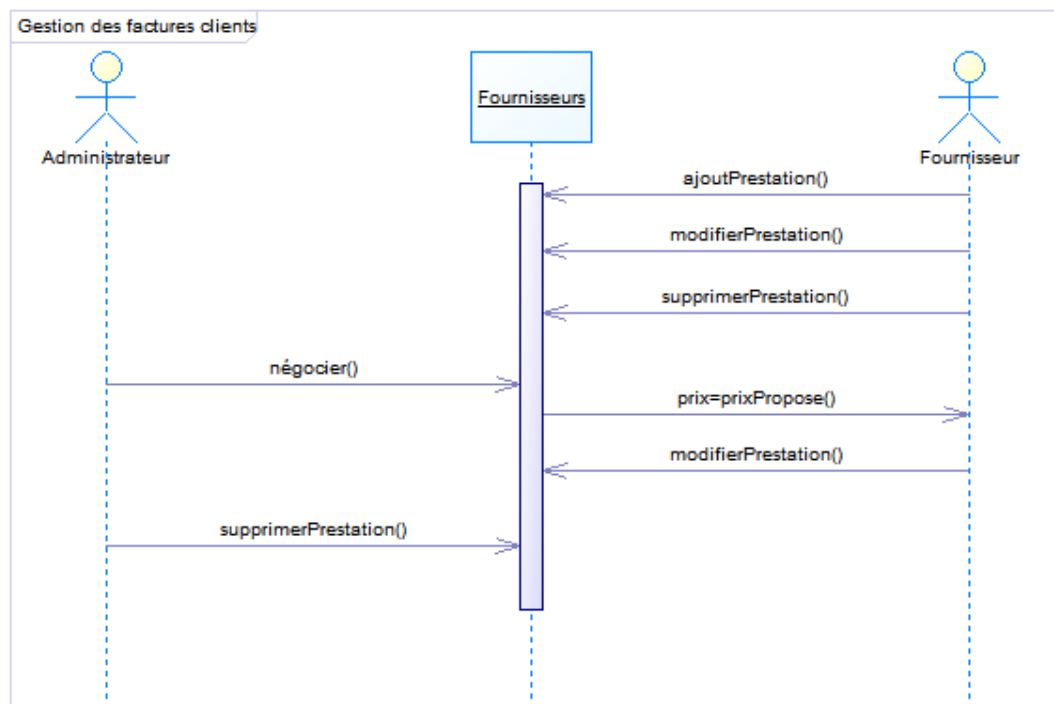


Figure 22:Diagramme de séquence pour la gestion des prestations des fournisseurs

Diagramme de séquence pour gérer la caisse :

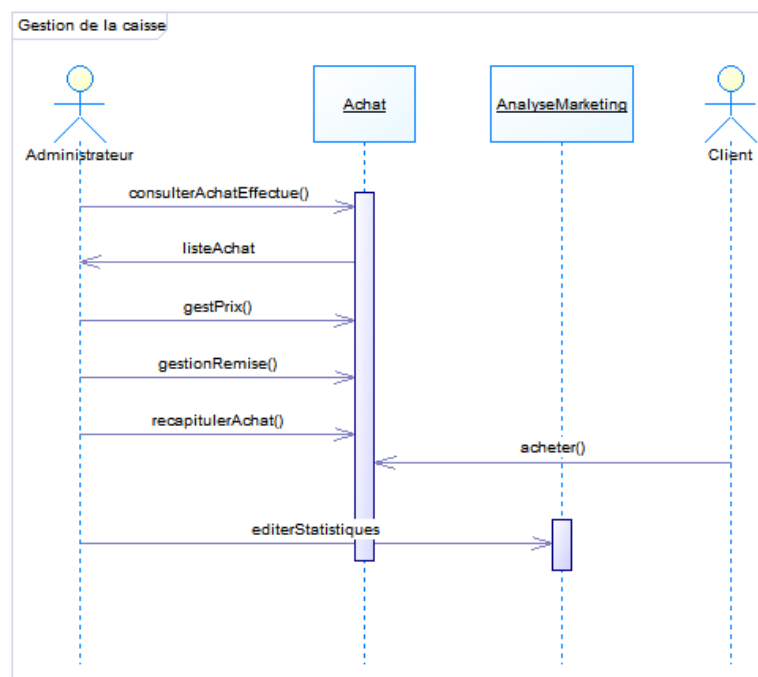


Figure 22:Diagramme de séquence pour la gestion de la caisse

2.8. Diagramme des classes

Le diagramme des classes est un schéma utilisé pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels.

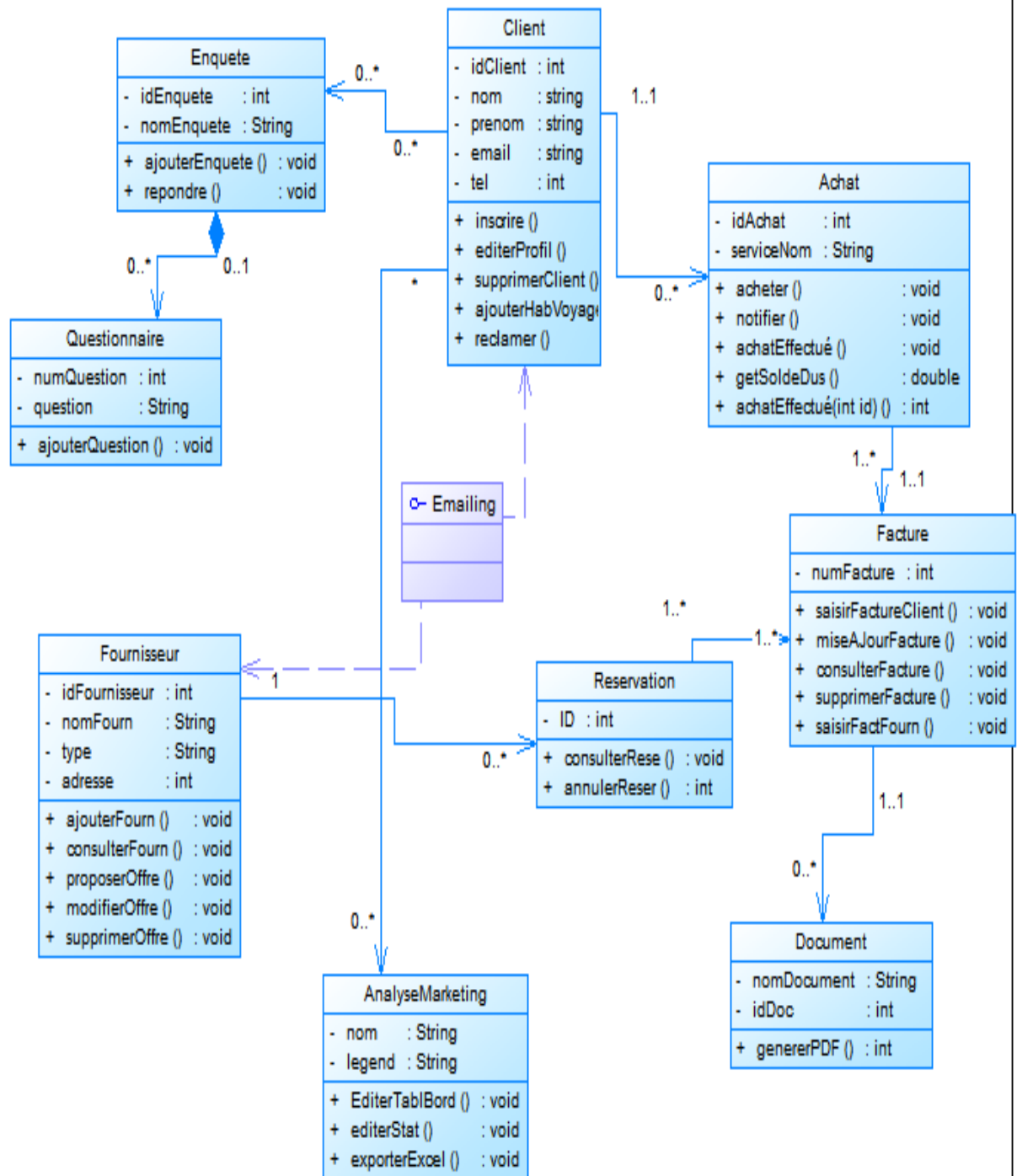


Figure 23:Diagramme des classes

2.9. Diagramme du paquetage

Le diagramme de packages, appelé aussi diagrammes de paquetages ; permet de découper l'ensemble de classes (d'un modèle de classes) en sous-ensemble de classes en fonction de leurs dépendances, d'une tâche commune. En clair, on essaye de regrouper toutes les classes qui sont fortement dépendantes entre-elles avec le reste des classes. Par exemple toutes les classes qui servent aux techniques de marketing, toutes les classes qui servent au fonctionnement des servlets.

En d'autres termes c'est un arrangement logique, exactement comme on range des choses dans des cartons, on essaye de mettre ensemble tous les objets qui ont quelque chose en commun et on étiquète le carton pour le retrouver.

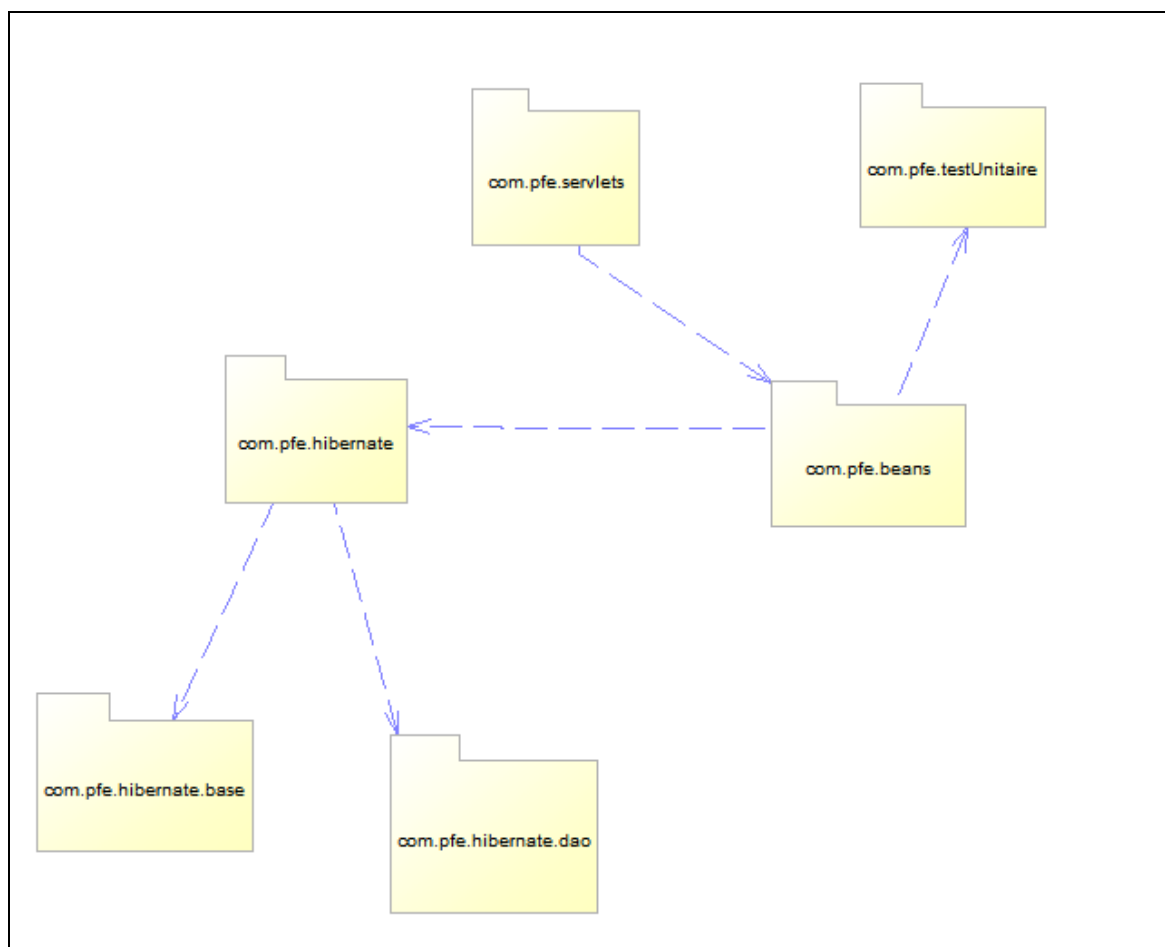


Figure 27 : diagramme de paquetage

2.10. Patrons de conception

Les patrons de conception (en anglais **Design Pattern**), on appelle aussi modèles de conception ou motifs de conception, sont un recueil de bonnes pratiques de conception pour un certain nombre de problème récurrents en programmation orientée objet.

Dans cette application, on va utiliser quelques patrons de conception comme :

- Modèle Vue Controlleur (MVC) : tous comme les patrons modèle-vue-contrôleur, est un modèle destiné à répondre aux applications interactives en séparant les problématiques liées au différents composants au sein de leur architecture respectives.
- Patron observateur/observable est utilisé en programmation pour envoyer un signal à des modules qui jouent le rôle d'observateur. On peut utiliser ce patron dans le cas des notifications.

2.11. Conclusion

Dans la phase de conception, on a décrit des diagrammes statiques comme le diagramme des classes et le diagramme de paquetage et on a décrit aussi des diagrammes dynamiques comme les diagrammes des séquences. En outre, on a détaillé chaque fonctionnalité par un raffinement.

Chapitre 3 :

Réalisation

Chapitre 3

Réalisation

3.1 Introduction

Après avoir achevé la partie de la définition des besoins et la partie de conception, on va entamer l'étape de la réalisation qui représente une partie très importante du temps consacré à ce projet.

L'objet de ce chapitre, ici, n'est pas de décrire les lignes du code source les une après les autres. Cela serait fastidieux et profondément ennuyeux pour le lecteur. Il s'agit plutôt de présenter l'environnement de travail, les interfaces utilisateurs de l'application ainsi que les tests d'évaluation du produit final.

3.2 Environnement de travail :

Dans la partie suivante, on présente les différents outils, des langages, des technologies, des plateformes et des frameworks utilisés pour développer cette application.

3.2.1 Eclipse

L'environnement qu'on a adopté pour développer notre application est Eclipse qui 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.

Au fur et à mesure que vous programmez, eclipse compile automatiquement le code que vous écrivez, en soulignant en rouge ou jaune les problèmes qu'il détecte. Il souligne en rouge les parties du programme qui ne compilent pas, et en jaune les parties qui compilent mais peuvent poser éventuellement problème (on dit qu'eclipse lève un avertissement, ou *warning* en anglais). Pendant l'écriture du code, cela peut sembler un peu déroutant au début, puisque tant que la ligne du code n'est pas terminée (en gros jusqu'au point virgule), eclipse indique une erreur dans le code. Il est déconseillé de continuer d'écrire le programme quand il contient des erreurs, car eclipse est dans ce cas moins performant pour vous aider à écrire le programme.

Eclipse a plusieurs versions. Chaque version a un nom de code. Voici la liste des versions d'eclipse :

Nom de code	Date	Version
Mars	Juin 2015	4.5
Luna	25 juin 2014	4.4
Kepler	27 juin 2013	4.3
Indigo	27 juin 2012	3.7
Helios	22 juin 2011	3.6
Galileo	23 juin 2010	3.5
Ganymède	Février 2009	3.4.2
Europa	29 juin 2007	3.3
Callisto	Janvier 2006	3.2.1
Eclipse 3.1	Mars 2006	3.1.2
Eclipse 3.0	Mars 2004	3.0.2
Eclipse v2	2003	2.1.3
Eclipse v1	2001	1.0

Tableau 2 : Liste des versions d'eclipse.

Eclipse peut intervenir dans tous les différentes phases de développement d'une application.

Phases	outils
Analyse de besoin	Latex (pour éditer des documents).
Conception	UMLet 13.2
Codage	-JSF, Hibernate, plugins C++, pages HTML, feuille CSS,...
Test	JUnit

Tableau 3 : Les outils proposés par eclipse.

On a pris le choix d'utiliser Eclipse comme un éditeur car il d'intègre **Java Entreprise Edition**, ou Java EE (anciennement J2EE). Il est plus facile à mettre en place, toutes les APIs qui sont intégrées.

L'application est destinée à être hébergée en ligne en utilisant les technologies du web. Il existe plusieurs langages de programmation pour le développement de l'application web. Parmi ces langages on trouve :

- ASP** : technologie développée par Microsoft à 1996 dans le but de créer des pages web dynamique. ASP permet d'ajouter de code dans la page HTML qui sera interprété par le serveur.

- PHP** : (HyperText Preprocessor) est un langage de script interprété coté serveur. Il a plusieurs framework comme Zend, Symfony, Surikat,...

- Django** : est un framework open-source de développement web en python.

Vu la popularité de java dans les dernières années par son explosion, il est raisonnable de penser pour utiliser ce langage pour développer cette application.

Java propose plusieurs avantages, qui sont :

- Langage orienté objet pur (tout orienté objet sauf les types primitifs).
- Grande robustesse : langage fortement typé, gérer de façon automatique la mémoire, gérer les erreurs,...
- Portabilité.
- Gestion de la sécurité.
- Forte capacité d'intégration aux environnements web.
- Facilité d'écriture d'interfaces graphiques professionnelles.

Java est composée de nombreuses bibliothèques ou API (interface de programmation), citons par exemple java.lang, java.io, java.math, etc. Bref, toutes ce bibliothèques contiennent un nombre conséquent de classes et des méthodes prêtes à l'emploi pour effectuer toutes sortes de tâches.

La technologie adoptée : J2EE

Le terme « Java EE » signifie Java Entreprise Edition, et était anciennement raccourci en « J2EE ». 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.

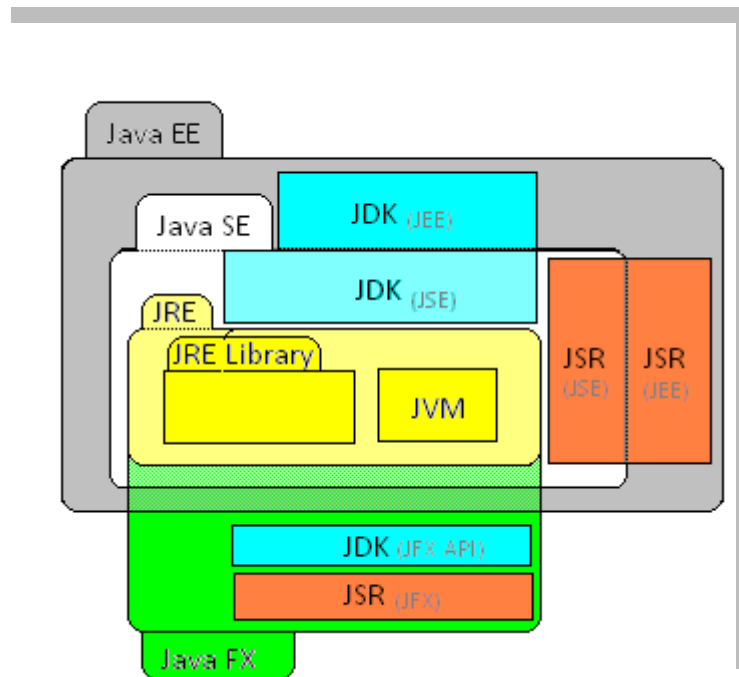


Figure 24 : Position du Java Entreprise Edition

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.

3.2.2 MySQL

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

Cette application a besoin de plusieurs données pour bien tourner comme les données et prospects des clients, différents achats effectué, etc.

Alors, on a décidé d'utiliser MySQL comme système de gestion de bases de données. Le choix est pris car :

- .Solution très courante en hébergement public.
- .Open-source, bien que les critères de licence soient de plus en plus difficile.
- .Plusieurs de stockage adaptés aux différentes problématiques, configurable.
- .C'est un SGBD relationnel.

Notre principale mission est bien entendu d'effectuer la liaison entre MySQL et notre projet Java EE. Car ce que nous souhaitons, c'est pouvoir interagir avec les tables de la base « **touroperateur** » directement depuis le code de notre application.

Pour faire la liaison entre une application Java EE et MySQL, on a utilisé une solution standard se nomme JDBC qui est une API qui fait partie intégrante de la plate-forme Java, et qui est constituée de classes permettant l'accès depuis notre applications Java à des données stockées dans un SGBD. Les actions rendues possibles par cette API sont : la connexion avec le SGBD, l'envoi de la requête SQL au SGBD depuis une application Java, le traitement des données et éventuelles erreurs retournées par le SGBD lors des différentes étapes de dialogues.

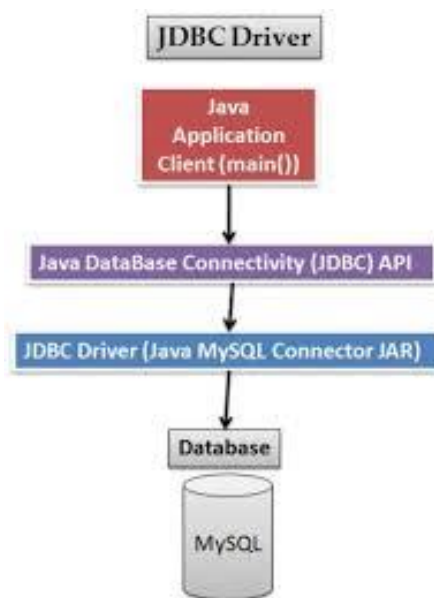


Figure 25 : Liaison entre application Java et MySQL

3.2.3 WampServer

Pour gérer les données, on a pris le choix d'utiliser WampServer qui est une plateforme web de type WAMP.

WampServer n'est pas un logiciel en soi un logiciel mais un environnement comprenant deux serveurs qui sont Apache et MySQL, un interpréteur de script PHP ainsi que **phpMyAdmin** pour l'administration web des bases MySQL.

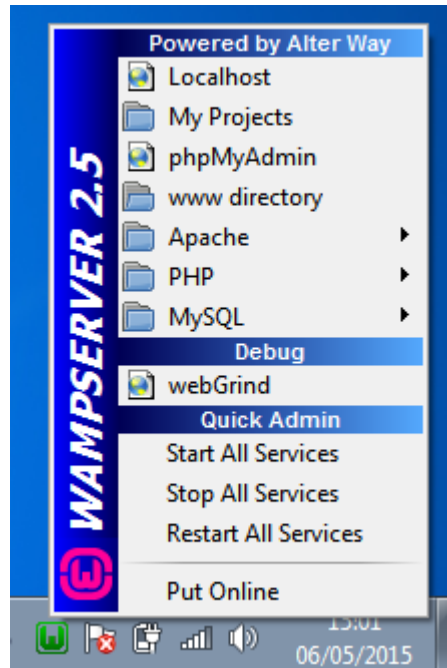


Figure 26 : Menu de WampServer

Dans l'environnement du WampServer, on va utiliser que la partie **phpMyAdmin** pour administrer les bases des données MySQL.

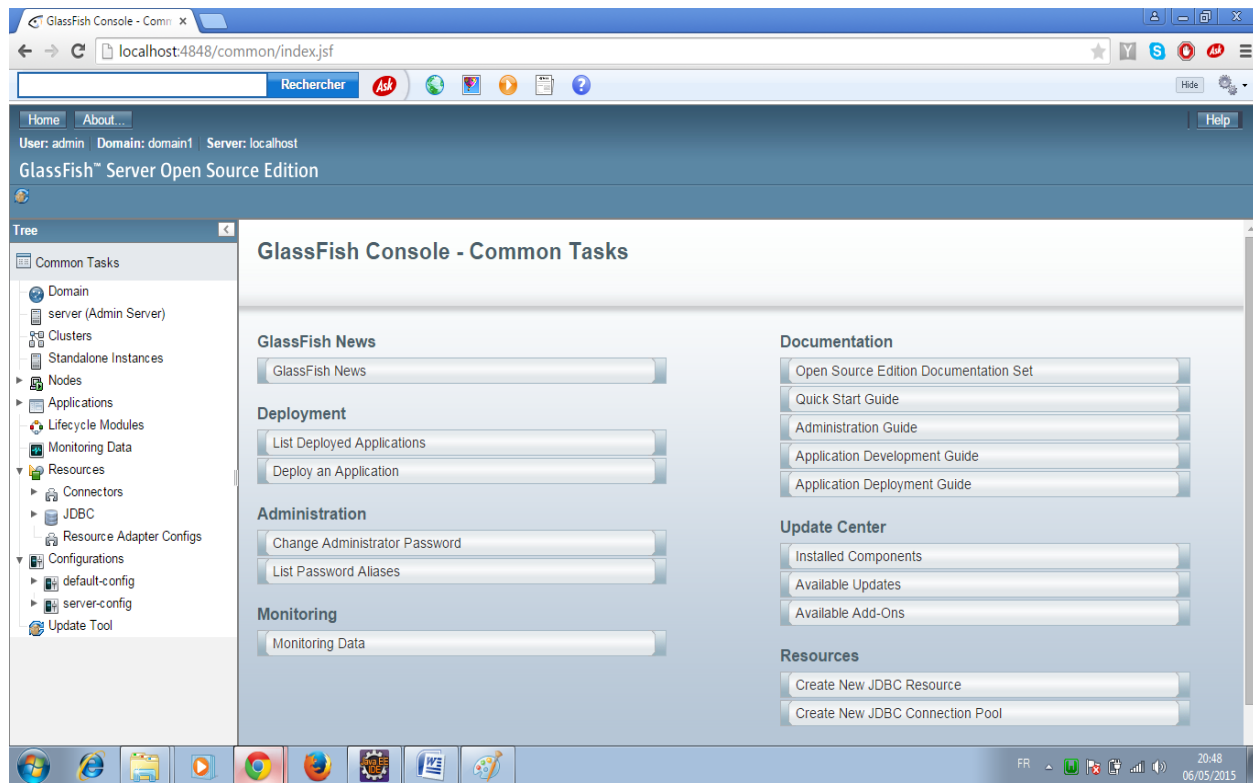
Il existe plusieurs façons d'accéder à sa base de données et d'y faire des modifications. On peut utiliser une ligne de commande (console), exécuter des requêtes SQL ou faire appel à un programme qui nous permet d'avoir rapidement une vue ensemble. Ici on va découvrir phpMyAdmin, un des outils les plus connus permettant de manipuler une base des données MySQL.

PhpMyAdmin est l'une de plus célèbre interfaces pour gérer une base des données MySQL. Cette interface pratique permet d'exécuter, très facilement et sans grandes connaissances dans le domaine des bases des données, des nombreuses requêtes comme les créations de table des données, les insertions, les mises à jour, les suppressions les modifications de structure de la base des données. Ce système est très pratique pour sauvegarder une base des données sous forme de fichier **.sql** et ainsi transférer facilement ses données. De plus celui-ci accepte la formulation de requête SQL directement en langage SQL, cela permet de tester ses requêtes par exemple lors de la création d'un site et ainsi de gagner un temps précieux.

3.2.4 Glassfish

Pour faire fonctionner une application Java EE, nous avons besoin de mettre en place un serveur d'applications. Il existe plusieurs sur le marché. J'ai choisi d'utiliser **GlassFish**.

Glassfish est un serveur d'applications open-source Java EE5 et désormais Java EE6 avec la version 3.



On a débuté le codage de l'application avec le serveur Tomcat. En revanche, ce dernier n'est pas performant en utilisant le framework JSF et le standard JPA. Alors, on a décidé d'utiliser le serveur Glassfish. On a trouvé que **Glassfish 4** est léger et complet. L'utilisation de Tomcat nécessite l'ajout des jars des différentes solutions dont on a besoin pour continuer à travailler avec ce serveur.

On a trouvé quelques difficultés dans l'installation du serveur Glassfish. Puisque Eclipse a plusieurs versions (Helios, Luna,...), ils existent des versions qui ne peuvent pas intégrer le serveur Glassfish. Pour cela, on a passé plus de temps pour trouver une solution et installer le serveur dans notre machine. Le tableau ci-dessous permet de lister les IDE, les versions d'Eclipse ainsi que la version de java implémenté :

IDE	GF 3.1.x	GF 4.0	Java EE 6	Java EE 7	Status
NetBeans 7.3.1	✓	✓ Co-bundled	✓	✓	Released
Eclipse Kepler (4.3 - Mid 2013)	✓	✓	✓ WTP	✓ WTP	Released
Eclipse Juno (4.2 - Mid 2012)	✓	✓	✓ WTP	.	Released
Eclipse Indigo (3.7 - Mid 2011)	✓	.	✓ WTP	.	Released
Eclipse Helios (3.6 - Mid 2010)	✓	.	✓ WTP	.	Released
IntelliJ Ultimate 12.1	✓	.	✓	.	Released
IntelliJ Ultimate 13	✓	✓	✓	✓	Pre-release
JDeveloper	✓	.	✓	.	Released

Tableau 4 : Listes des IDE correspond au serveur Glassfish

On a trouvé une difficulté dans l'installation du la machine java virtuelle (JVM) qui est un appareil informatique fictif qui exécute des programmes compilés sous forme **bytecode**(suivre l'annexe). Pour cela, on a fait commencer par le JRE (suivre l'annexe)(Java Runtime Environnement), mais celui-ci n'est pas compatible avec la version Eclipse (Eclipse Kepler) ainsi qu'avec le serveur Glassfish.

On a installé JDK(suivre l'annexe) (Java Developpement Kit) pour faire la correspondance. De plus, on a décidé de choisir la version 8 JDK qui est *jdk1.8.0_40*.

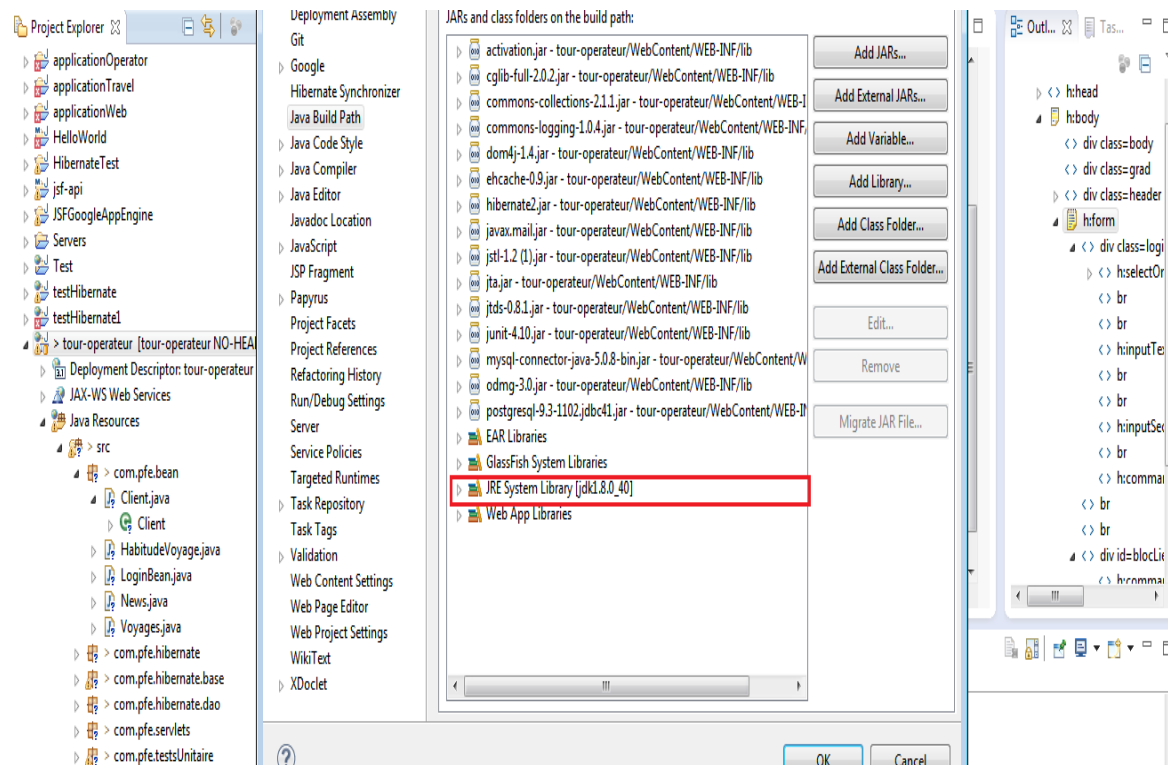


Figure 28 : JDK installé

Pour que le serveur Glassfish exécute l'application et on fait associer l'application à une URL. Ainsi lorsque le client la saisira, la requête http sera automatiquement aiguillée par notre conteneur de servlet vers la bonne servlet, celle qui est en charge de répondre à cette requête. Ce quelque part se présente sous la forme d'un simple fichier texte : le fichier **glassfish-web.xml**

Le fichier glassfish-web.xml est le cœur de serveur : ici vont se trouver tous les paramètres qui permettent de contrôler le serveur. Ce fichier est doit être impérativement se nommer **glassfish-web.xml** et se situer juste sous le répertoire **/WEB-INF** de l'application. Voici ci-dessous le contenu du fichier glassfish-web.xml dans notre application :

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE glassfish-web-app PUBLIC "-
//GlassFish.org/DTD GlassFish Application Server 3.1 Servlet 3.0//EN"
"http://glassfish.org/dtds/glassfish-web-app_3_0-1.dtd">

<glassfish-web-app>
  <context-root>/tour-operateur/index.xhtml</context-root>
</glassfish-web-app>
```

Figure 29 : Contenu de glassfish-web.xml

3.2.5 Framework Hibernate

Les applications d'entreprises s'orientent de plus en plus vers des architectures n-tiers. La technologie J2EE et les frameworks qui se sont offerts beaucoup d'outils pour répondre aux besoins modernes. Pour la couche présentation, on trouve par exemple, le plus populaire **Struts** respectant le modèle MVC de séparation de code, de l'interface et des données.

Nous allons, cependant, nous intéresser à une couche plus basse d'une architecture applicative, la couche d'accès aux données. Celle-ci permet d'interfacer le code métier avec une source des données. L'intérêt est de pouvoir changer de base de données en n'ayant besoin de ne modifier que la couche d'accès. Pour réaliser cette tâche, il existe plusieurs solutions fournies par J2EE qui sont appelées **JPA** (suivre l'annexe) (Java Persistence API).

Ce mécanisme qui gère la correspondance entre des objets d'une application et les tables de base des données se nomme ORM (Object-Relationnal Mapping). Pour cela, on a pris la décision d'utiliser **Hibernate** comme un **framework ORM**.

On a pris le choix d'utiliser Hibernate car :

- .Génère le code SQL nécessaire, ce qui rend l'application plus portable.
- .La persistance est transparente.
- .La récupération des données est optimisée.
- .Portabilité du code en cas de changement de la base des données.

Pour la technologie de Sun, la spécification récente de JDO qui s'avère assez peu utilisée dans les faits. Le framework le plus populaire pour J2EE est sans contexte Hibernate. Voici comment se présente globalement l'architecture d'Hibernate.

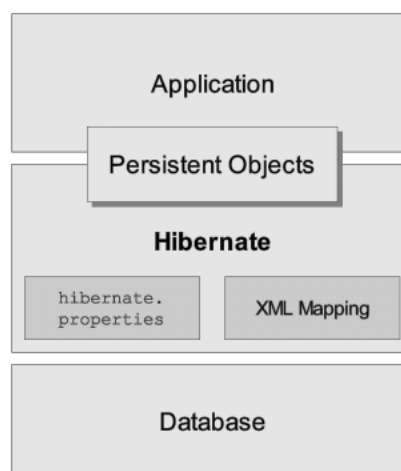


Figure 30 : Présentation de Hibernate

En effet, la couche applicative voit les données comme des classes dont le contenu reste en mémoire même après la fin d'exécution. D'où persistance objet des données. De plus, le lien entre les classes exposées et la source physique des données (souvent une base des données relationnelle) est définie par un fichier .xml . D'où mapping objet-relationnel.

On a installé Hibernate dans l'éditeur Eclipse, puis on fait préparer le fichier **hibernate.cfg.xml** que va permettre d'indiquer au moteur d'Hibernate les paramètres de connexion à la base des données.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE hibernate-configuration
PUBLIC "-//Hibernate/Hibernate Configuration DTD//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-2.0.dtd">

<hibernate-configuration>
  <session-factory>
    <!-- local connection properties -->
    <property name="hibernate.connection.url">
      jdbc:mysql://localhost:3306/touroperateur
    </property>
    <property name="hibernate.connection.driver_class">
      com.mysql.jdbc.Driver
    </property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.connection.password">
    </property>
    <!-- property name="hibernate.connection.pool_size"></property -->
    <!-- dialect for MySQL -->
    <property name="dialect">
      net.sf.hibernate.dialect.MySQLDialect
    </property>
    <property name="hibernate.show_sql">false</property>
    <property name="hibernate.use_outer_join">true</property>
    <property name="hibernate.transaction.factory_class">
      net.sf.hibernate.transaction.JTATransactionFactory
    </property>
    <property name="jta.UserTransaction">
      java:comp/UserTransaction
    </property>
  </session-factory>
</hibernate-configuration>
```

Figure 31 : Le contenu de fichier hibernate.cfg.xml

Hibernate un fichier du mapping xml qui est un élément majeur puisqu'il va permettre à Hibernate de faire le pont entre les classes de persistance et la source des données. Le nom de fichier généralement contient le nom de table dans la base des données MySQL et sous l'extension .xml .

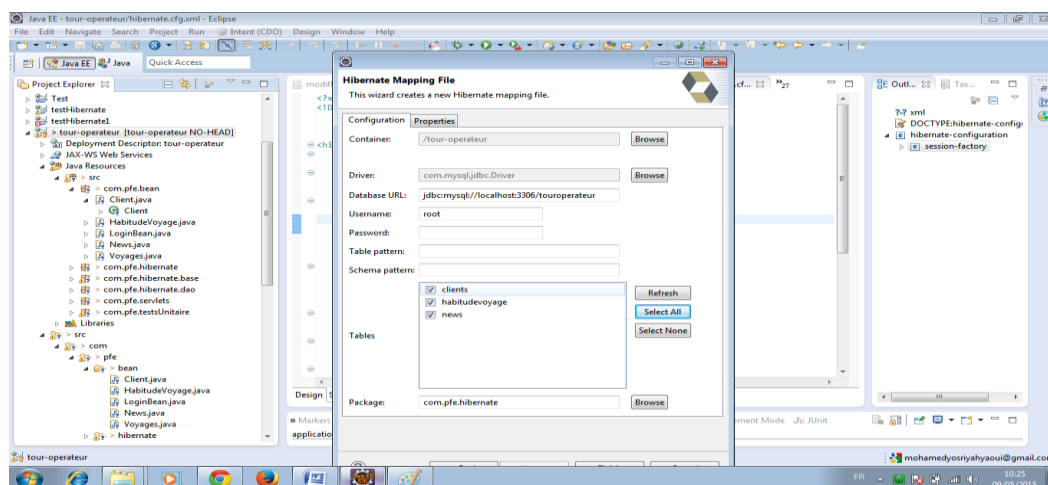


Figure 32 : Configuration de mapping d'Hibernate

Dans la réalisation de la partie d'Hibernate, on a trouvé aussi plusieurs difficultés et plusieurs exceptions qui sont levés. Parmi les exceptions qu'on a vu :

- `java.lang.ExceptionInInitializerError`
- `org.hibernate.HibernateException`

Hibernate génère des classes DAO qui sert à faire la liaison avec la base des données.

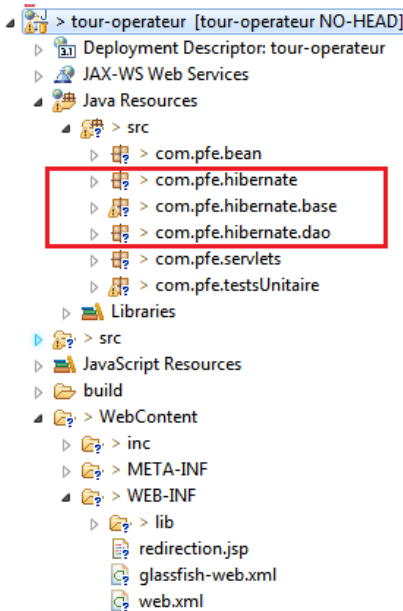


Figure 33 : Les fichiers DAO générés par Hibernate.

3.2.6 JSF, Servlets, Facelets, JSP

3.2.6.1 JSF

Pour respecter le modèle de conception MVC (Modèle-Vue-Contrôleur) et récupérer l'avantage de ce fameux patron de conception. On a le choix d'utiliser un standard adopté par Java EE 6 qui est un **framework MVC**.

On a plusieurs choix pour les frameworks MVC proposés. On a deux types de framework soit des frameworks basés sur les composants soit des frameworks basés sur des requêtes.

Framework basés sur les composants	Framework basés sur les requêtes
.JSF	.Spring
.Wicket	.Struts
.Tapestry	.Stripes

Tableau 4 : Les différents types des frameworks MVC

Notre choix est d'utiliser JSF (Acronyme :JavaServer Faces). Dans ce qui suit, on va présenter les avantages du JSF :

- 1) L'assemblage des composants serveurs qui génèrent le code de leur rendu avec la possibilité d'associer certains composants à une source des données encapsulée dans un bean.
- 2) L'utilisation d'un modèle de développement standardisé reposant sur l'utilisation d'événements et de listeners.
- 3) La conversion et la validation des données avant leur utilisation dans les traitements.
- 4) La gestion de l'état des composants de l'interface graphique.

3.2.6.2 Servlets

On a utilisé des servlets dans l'application comme un rôle de contrôleur ou plus précisément comme un aiguilleur. Dans ce qui suit, on fait présenter les différents avantages d'un servlet et les types des servlets.

Les avantages des servlets dans une application J2EE :

- Indépendance issue de la plate-forme java.
- Modèle de sécurité issue de serveur Web.
- Support dans la plupart des serveurs Web.
- Exploite tous les API java.

Dans cette application, il existe deux types de servlets :

1. On a utilisé des simples servlets qui ont le rôle de contrôleur dans le patron de conception MVC. Ils ont la tâche d'aiguilleur. Ces servlets sont déclarés dans le fichier **web.xml**.

```
<servlet>
  <servlet-name>redirect</servlet-name>
  <servlet-class>com.pfe.servlets.Redirection</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>redirect</servlet-name>
  <url-pattern>/redirect</url-pattern>
</servlet-mapping>
</web-app>
```

Figure 34 : Contenu de web.xml (1)

2. Le framework JSF utilise une seule servlet appelé *front-controller* va elle même regrouper, valider, convertir les paramètres de requêtes, et mettre à jour les valeurs de modèle. Cette servlet est déclarée aussi dans le fichier **web.xml**.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns
  <context-param>
<param-name>javax.faces.PROJECT_STAGE</param-name>
<param-value>Development</param-value>
</context-param>
<!-- Déclaration du contrôleur central de JSF : la FacesServlet -->
<servlet>
<servlet-name>Faces Servlet</servlet-name>
<servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
<load-on-startup>1</load-on-startup>
</servlet>
<!-- Mapping : association des requêtes dont le fichier porte
l'extension .xhtml à la FacesServlet -->
<servlet-mapping>
<servlet-name>Faces Servlet</servlet-name>
<url-pattern>*.xhtml</url-pattern>
</servlet-mapping>
```

Figure 35 : contenu du fichier web.xml (2)

3.2.6.3 Facelets

Dans cette application, plus précisément dans le composant **Vue** dans le patron conception, on a utilisé des simples pages sous l'extension XHTML, pages que l'on nomme des **facelets**. On appelle **facelets** ou **templating**.

Une facelet est un fichier XML pur, il ne peut par conséquent contenir que des balises, et il est impossible d'y inclure des scripts comme la technologie JSP.

La facelet offre plusieurs avantages :

- Uniformiser la structure des pages.
- Simplifier la mise à jour : une modification dans le template se propage automatiquement dans toutes les pages qui l'utilisent.
- Gain de productivité: moins de code à écrire : une page ne contient ce qui lui propre.

De plus, les pages facelets intègrent plusieurs bibliothèques par l'utilisation des directives.

3.2.6.4 JSP

Dans l'utilisation des pages de type facelets, on a tombé sur plusieurs imperfection.

Par exemple, il n'existe pas une fonction dans une page XHTML qui permet de faire la redirection d'une à une autre.

Alors, on a trouvé la solution qui est l'utilisation des pages JSP.

Notre principal intérêt est de rendre possible la création de pages dynamiques : puis qu'il y a une étape de génération sur le serveur, il devient possible de faire varier l'affichage et d'interagir avec l'utilisateur, en fonction notamment de la requête et des données reçus.

Dans une page JSP, on peut utiliser des portions code java appelé des scriptlets.

On va citer les avantages des pages JSP :

- un langage dédié : les pages JSP sont des documents au format texte, à l'opposé des classes Java que sont les servlets, qui décrivent indirectement comment traiter une requête et construire une réponse. Elles contiennent des balises qui combinent à la fois simplicité et puissance, via une syntaxe simple, semblable au HTML et donc aisément compréhensible par un humain.

- la simplicité d'accès aux objets Java : des balises du langage rendent l'utilisation directe d'objets au sein d'une page très aisée.

- des mécanismes permettant l'extension du langage utilisé au sein des pages JSP : il est possible de mettre en place des balises qui n'existent pas dans le langage JSP, afin d'augmenter les fonctionnalités accessibles. Pas de panique, ça paraît complexe a priori mais nous y reviendrons calmement dans la partie concernant la JSTL, et tout cela n'aura bientôt plus aucun secret pour vous.

Les pages JSP nous donne la possibilité d'utiliser la technologie JSTL qui est une bibliothèque, une collection regroupant des balises implémentant des fonctionnalités à des fins générale, communes aux applications web. Citons par exemple la mise en place de boucles, de tests conditionnels, le formatage des données ou encore la manipulation des données XML. L'objectif est d'éviter le développeur à utiliser de code java.

3.2.7 EGit

Lors de la réalisation de cette application, on a utilisé un logiciel de gestion des versions qui est un logiciel qui permet de stocker un ensemble de fichiers en conservant la chronologie de toutes les modifications qui ont été effectué dessus. Il permet notamment de retrouver les différentes versions d'un lot de fichier connexes.

Pour cela, on fait le choix sur EGit qui est simplement un *plug-in* pour l'IDE Eclipse, qui fournit une interface graphique pour utiliser GIT directement.

Voici ci-dessous les avantages d'EGit :

- Pas d'aller-retour entre Eclipse et une application tierce. Votre IDE préféré contient tous les outils nécessaires. Un sous-menu ainsi que des raccourcis (à ajouter manuellement) et plusieurs vues vous permettent de l'utiliser facilement.
- EGit est totalement gratuit et fonctionne sur toutes les plateformes (il est basé sur JGit, une implémentation Java de Git).
- En tant que plug-in et comme Eclipse est portable, il ne nécessite pas d'installation comme pour un programme Windows par exemple, donc vous pouvez l'utiliser même si vous n'êtes pas administrateur.
- Contrairement à certaines applications graphiques, EGit ne vous impose aucune contrainte, que ce soit quant à l'emplacement des dépôts sur votre machine, au nombre de dépôts distants, etc. Rien ne vous empêche d'avoir plusieurs espaces de travail (workspaces) avec des dépôts distincts, vous êtes libres !

Avec l'utilisation d'EGit, on a fait deux étapes :

1. On a créé un dépôt local pour le projet, dans lequel Git stockera toutes les informations qui concerneront le suivi de l'état de vos fichiers.
2. Envoi de projet au GitHub pour mettre le projet à distant.

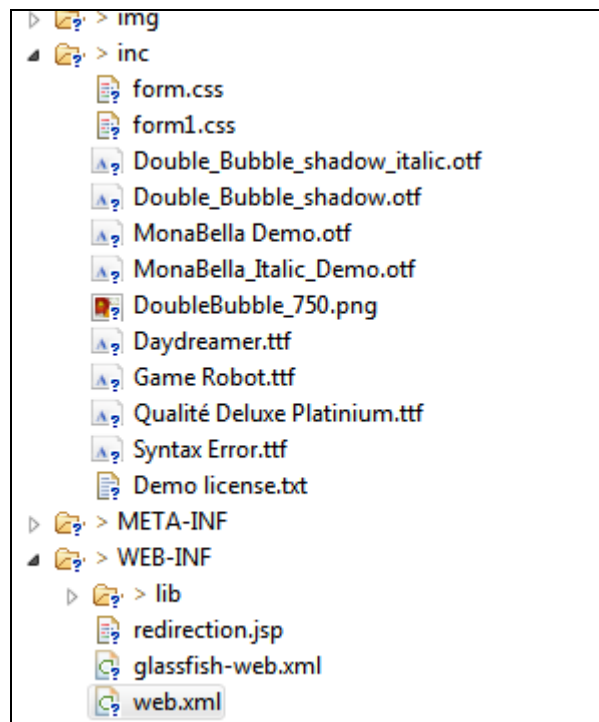


Figure 36 : Capture sur le fichier Git

3.2.8 Test unitaire : JUnit

Au cours de la réalisation de cette application, on a utilisé des tests unitaires, plus précisément sur les *classes bean*(4). Pour cela, on a fait le choix sur l'utilisation de JUnit qui est framework open source pour le développement et l'exécution des tests unitaires automatisables. Le principal intérêt est de s'assurer que le code répond toujours aux besoins même après d'éventuelles modifications. Plus généralement, ce type de tests est appelé tests unitaires de non régression.

JUnit propose :

- Un framework pour le développement des tests unitaires reposant sur des assertions qui testent les résultats attendus.
- Des applications pour permettre l'exécution des tests et afficher les résultats.

Le but est d'automatiser les tests. Ceux-ci sont exprimés dans des classes sous la forme de cas de tests avec leurs résultats attendus. JUnit exécute ces tests et les comparent avec ces résultats.

Cela permet de séparer le code de la classe, du code qui permet de la tester. Souvent pour tester une classe, il est facile de créer une méthode `main()` qui va contenir les traitements de tests. L'inconvénient est que ce code "superflu" est inclus dans la classe. De plus, son execution doit se faire manuellement.

De plus, l'éditeur Eclipse peut intégrer JUnit par l'ajout d'un simple plug-in.

```
package com.pfe.testsUnitaire;

import com.pfe.bean.Client;

import org.junit.Test;

public class ClassTest {

    @Test
    public void test() {
        //Tester la classe Client.
        Client client=new Client();
        //Test de méthode afficheNews() qui permet d'afficher tous les actualités.
        client.afficheNews();
    }

}
```

Figure 37 : Exemple d'une classe du test unitaire.

3.2.9 Essai avec Cloud du Google

Dans cette application, on a essayé à utiliser un Cloud pour héberger en ligne. Pour cela, on a fait le choix par l'utilisation de Cloud du Google appelé *Google App Engine* (suivre l'annexe). (On appelle aussi GAE).



Figure 38 : Serveurs de Google.

Les avantages de Google App Engine sont montrés par ses services offerts:

- **AppEngine** : le service PaaS de Google, la star de la maison.
- **Compute Engine** : le service IaaS de Google. Si vous avez besoin de puissance de calcul brute (pour calculer des modèles mathématiques par exemple), c'est ce qu'il faut utiliser.
- **Cloud Storage** : un service potentiellement « illimité » de stockage de fichiers dans le cloud.
- **Big Query** : des fonctionnalités permettant d'analyser de grosses quantités de données en peu de temps (on parle de Big Data). Très utile pour les entreprises qui veulent faire de la Business Intelligence (BI) pour avoir des indicateurs de tendance basés parfois sur plusieurs Tera Octets de données.
- **Cloud SQL** : une base de données MySQL distribuée dans le cloud. Vous n'avez pas besoin d'installer ni de mettre à jour MySQL. Pas même besoin de le configurer. Vous pouvez créer des serveurs SQL répliqués en quelques clics si vous avez besoin de plus de puissance.

On a fait installer le plugin de Google App Engine. On a débuté le développement sur le serveur de Google. Le mini-serveur du Google qui est en local a été bien fonctionné. Mais lors de déploiement, on a trouvé plusieurs problèmes. Alors, on arrêté le travail avec le serveur de Google.

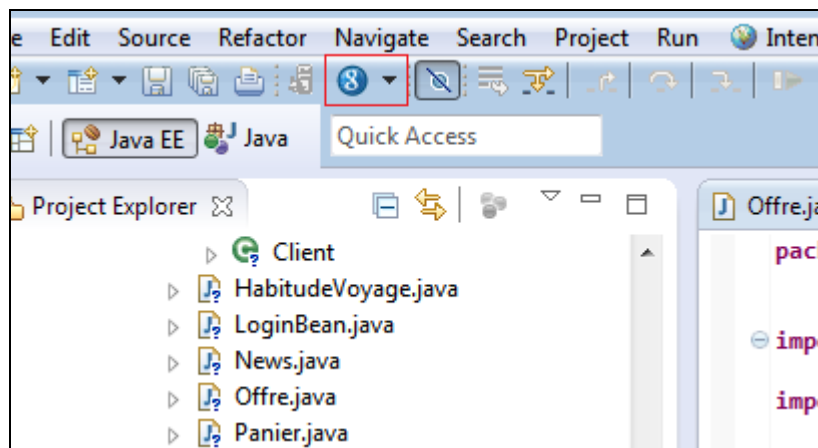


Figure 39 : Existence de plugin GAE dans Eclipse.

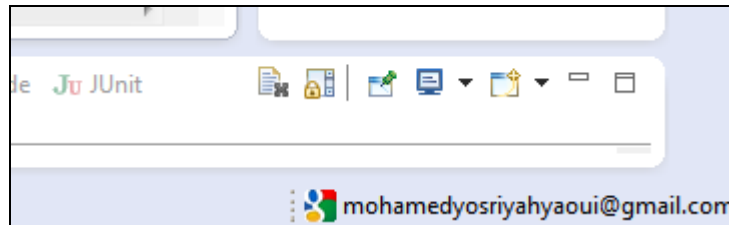


Figure 40 : Identifiant d'accès à l'application

Google app engine mohamedyosriahyaoui@gmail.com | [My Account](#) | [Help](#) | [Sign out](#)

My Applications

« Prev 20 1-7 of 7 Next 20 »

Application	Title	Storage Scheme	Status
applicationsiteoperator	applicationsiteoperator	High Replication	None Deployed
applicationsiteoperatortr	applicationsiteoperatortr	High Replication	Running
apptravelopera1	apptravelopera1	High Replication	Running
linear-ellipse-87306	Agency	High Replication	None Deployed
tour-tour-travel	tour-tour-travel	High Replication	Running
travel-operator	travel-operator	High Replication	Running
travel-tour-operator	Tour operateur	High Replication	Running

Create Application You have 18 applications remaining. « Prev 20 1-7 of 7 Next 20 »

Figure 41 : Déploiement de cette application.

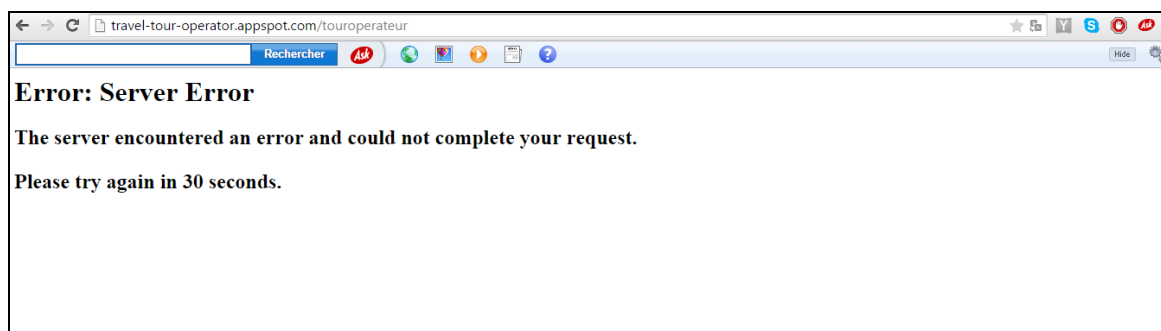


Figure 42 : Erreur survenu lors de déploiement

3.3 Présentation des interfaces :

On présente quelques interfaces de l'application web.

La figure ci-dessous est la page de bienvenue qui permet aux acteurs de l'application de s'identifier par leurs pseudonymes et leurs mots de passe en précisant leurs statuts soit un client visiteur, fournisseur ou l'administrateur qui a le rôle de gérer le contenu de l'application.

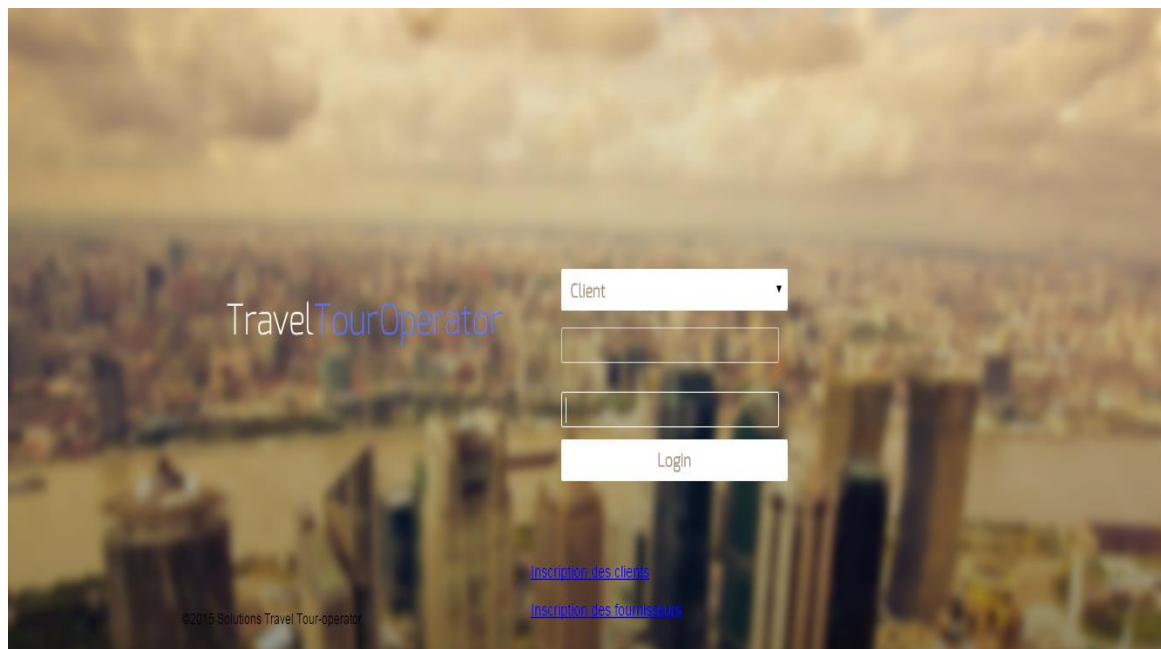


Figure 48 : page d'authentification

On oblige les visiteurs de cette application (fournisseurs et clients) à inscrire avant d'accéder, pour récupérer plus d'informations. Ceci, va assurer l'intégration de système *business intelligence*.

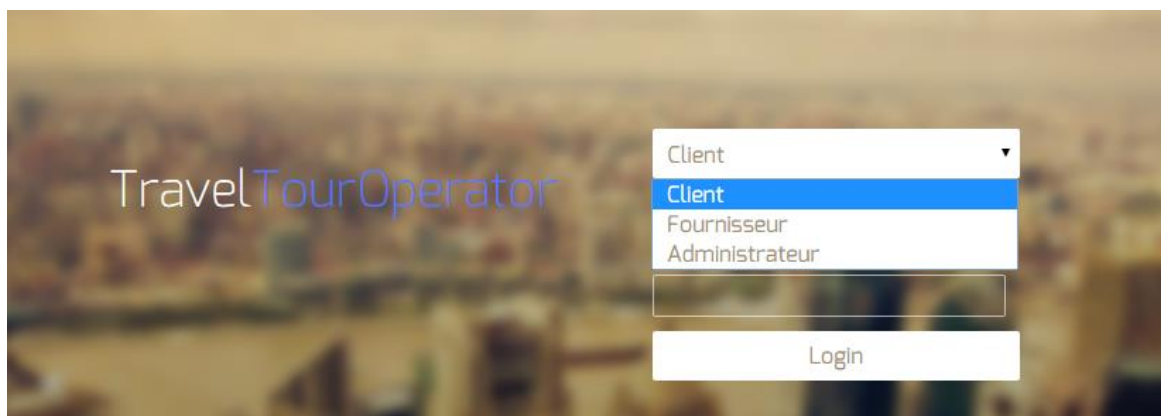


Figure 49 : Type des acteurs dans l'application

On a donné à chaque internaute, le droit de s'inscrire dans cette application. En revanche, on a contrôlé leurs données saisies.

Inscription des clients:

- Il faut saisir un nom client
- Il faut saisir un prénom client
- Merci de sélectionner votre pays
- Il faut saisir votre adresse
- Il faut saisir un numéro du téléphone
- Il faut saisir un numéro du fax
- Il faut saisir votre adresse électronique
- Merci de saisir une date naissance
- Merci de saisir un mot de passe

civilité *:

Nom *: Prénom *:

Pays:

Adresse *:

numéro de téléphone *:

numéro du fax *:

adresse électronique *:

Date du naissance *: sous le forme jj/mm/aaaa

mot de passe *: confirmation *:

Figure 50 : Formulaire d'inscription pour les clients

Une fois l'accès à l'application est réussite, le client peut soit suivre son navigation au sein de l'application soit compléter ses habitudes des voyages.

Habitudes de voyages

Informations générale

Age :

Niveau du scolarité :

Tranche du reven :

Pays du naissance :

Voyage

Date du voyage:

Destination:

Type d'hébergement:

Transport:

[ajouter Voyage](#)

Figure 51 : Formulaire pour les habitudes des voyageurs

Ci-dessous, la page d'accueil pour les clients.

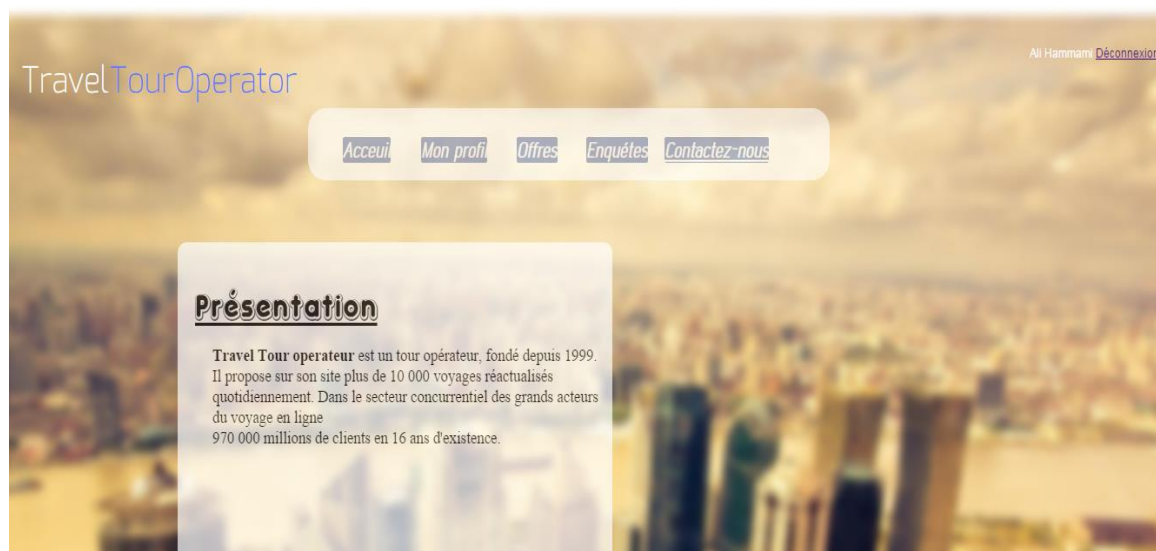


Figure 52 : Page d'accueil pour les clients.

Dans cette application, on intégré un système du *e-commerce* pour donner aux clients le droit d'effectuer des achats en ligne.

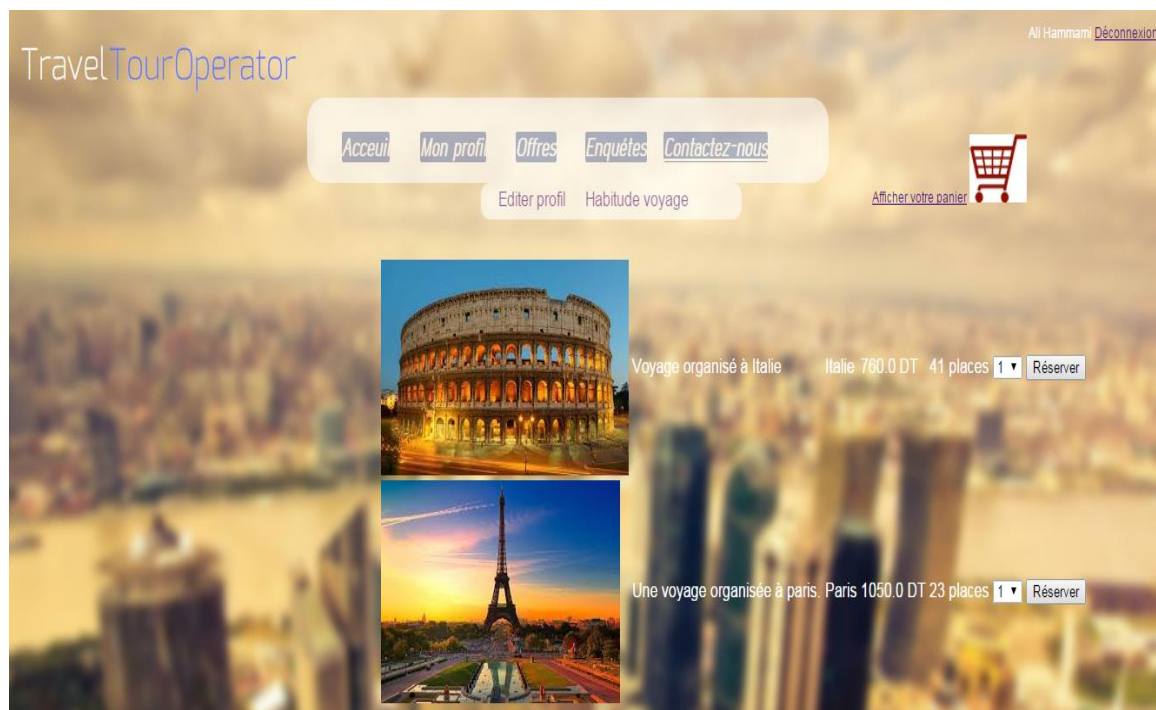


Figure 53 : Offres proposées

3.4 Diagramme de Gantt réel

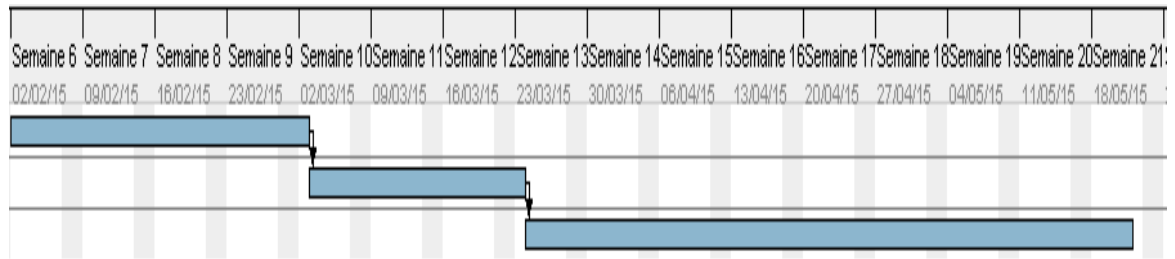


Figure 55 : Diagramme de Gantt réel

Lors en mise en pratique, on a fait un retard dans la spécification des besoins car on a trouvé plusieurs contraintes lors de la compréhension du sujet et la définition des besoins. On a fait un retard au codage parce qu'on a trouvé des problèmes techniques et plusieurs complexité pour réaliser cette application.

3.5 Conclusion

Dans ce chapitre, on a présenté les choix techniques que nous avons faits pour élaborer cette application, ainsi que des interfaces développées.

On a terminé une grande partie de cette application. On va essayer de compléter jusqu'au bout pour la déployer et pour qu'être visité.

Après son déploiement dans un réseau Internet, Extranet ou dans une plateforme Cloud, on va travailler sur sa visibilité et l'interrogation avec les internautes et les communautés par le terme populaire : c'est le **webmarketing**.

Chapitre 4 :

Déploiement ET Webmarketing

Chapitre 4

Déploiement et Webmarketing

4.1 Introduction

Forcément, la finalité d'une application web déployée sur Internet c'est du bien la visité. On ne crée pas un site web juste pour faire joli. Une question va alors très vite se poser : c'est comment générer ce trafic d'utilisation et comment amener les utilisateurs à visiter cette application web à partir du réseau Internet. La réponse qui reviendra le plus souvent, c'est celle du passage par le fameux terme dans les dernières années c'est le **Webmarketing**.

4.2 Le webmarketing

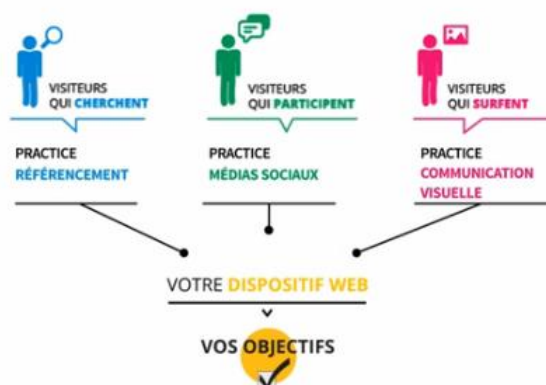
Le webmarketing, netmarketing ou cybermarketing c'est consiste à améliorer la visibilité et le trafic d'un site web en utilisant l'internet comme canal de prospection et de développer une relation durable de fidélisation avec les internautes utilisateurs ou clients d'un site web. Ce terme (webmarketing) est composé de deux mots clés **web** et **marketing**.

On va parler tout d'abord sur le **web** qui est une façon de communiquer et qui porte l'objectif de développement. Le web a aussi deux objectifs qui sont :

1. Attirer les prospects.
2. Convertir les prospects.

On va parler ensuite sur le **marketing** qui est la segmentation des utilisateurs de web. Pour cela, on trouve trois types des internautes :

1. Visiteurs qui cherchent. On peut utiliser le **référencement**.
2. Visiteurs qui participent et qui sont réunissent en communauté pour parler entre eux et peut faire des conversations pour les interagissent. C'est la le travail du **community management**.
3. Visiteurs qui surfent, on peut implémenter la solution **d'e-mailing**.



Tout de suite, on va parler des composants de dispositif du webmarketing ainsi que de préciser quelques solutions pour atteindre des objectifs.

4.3 Référencement

On utilise le terme « référencement » s'est vite imposé par abus de langage comme un terme générique pour désigner « **référencement** » et « **positionnement** ». Pour parler aussi bien de la présence de son site dans l'index que de sa position, on a vite commencé à n'utiliser que le mot « référencement ». N'hésitez donc pas à parler de référencement de votre site sur des forums ou dans une conversation. C'est le mot qui sera le plus compris.

Dans cette partie, on va travailler avec le moteur de recherche **Google**.

Les deux termes désignés ci-dessous se sont deux étapes. On va débiter par le référencement puis avec le positionnement.

Le référencement d'un site est l'action de signaler aux moteurs de recherches, aux annuaires, aux sites de réseaux sociaux, l'existence du site web.

On peut tester notre existence de notre site dans le moteur de recherche Google par un simple saisir dans la zone du recherche en tapant : **site : touroperateur-travel.com** (touroperateur-travel.com c'est le nom de domaine de cette application web).



Figure 44: Test d'indexation d'un site internet.

Puisque cette application n'est pas déployé, alors n'est pas indexés par Google plus précisément n'est pas connu par le robot de Google GoogleBot(suivre l'annexe).

Pour atteindre l'objectif de référencement, on va débiter par le référencement naturel puis par le référencement payant.

Référencement naturel :

La réalisation de référencement se fait en deux étapes qui sont :

- 1 Préparation et optimisation des pages du site.
- 2 Le suivi de référencement.

1. Préparation et optimisation des pages du site :

Dans cette partie, on peut suivre une trame chronologique des actions à mener :

- Définir les objectifs de positionnement désirés.
- Choisir des mots clés pertinents pour mettre dans la balise *<meta>*.
- Choisir de préférence un moteur de recherche et/ou annuaire à prendre en compte.
- Optimisation ou mise à jour des pages du site en fonction de ces mots-clés et des critères de pertinence des moteurs.
- Soumission du site auprès des index de moteur de recherche et/ou des annuaires.
- Contrôle du positionnement et/ou du trafic.
- Suivre régulièrement les phases de positionnement et prévoir des correctifs à faire éventuellement pour obtenir de meilleurs résultats.

2. Le suivi de référencement :

On peut suivre le référencement en utilisant des outils de Google. Voici ci-dessous quelques outils :

- Google Analytics est une solution professionnelle d'analyse d'audience Internet qui fournit des indications précieuses sur la fréquentation du site Web et son efficacité en termes de marketing. Il comporte des fonctionnalités performantes, flexibles et faciles à utiliser qui permettent aujourd'hui de consulter et d'analyser les données de trafic. Google Analytics permet d'aider à la rédaction des annonces afin de mieux cibler. Cet outil est gratuit, il suffit de s'y inscrire et d'installer le code JavaScript (Le code est donné par Google, mais n'est pas valid W3C) valid W3C, sur le site Web.
- L'outil « Google pour webmaster » : il indique le nombre de liens pointant vers votre site, le nombre de pages indexées par les moteurs.

Référencement payant :

Le référencement payant répond au besoin de se positionner immédiatement en fonction d'un budget fixé sur un ou plusieurs mot-clé/expression. Il s'agit donc d'une méthode afin de se démarquer de la concurrence comme nous avons pu le voir dans les premières parties de ce dossier. En outre, cela permet d'avoir plusieurs avantages comme :

- une localisation géographique précisée.
- une proposition automatique par Google des mots clés.

- une apparition immédiate sur Google.
- une proposition d’affichage sur le « réseau de contenu ».

Dans cette partie, on peut utiliser le service du Google, appelé **Google AdWords** qui est le nom commercial donné par Google aux liens sponsorisés, donc à la publicité présente sur les pages de résultats.

4.4 Community management :

Cette partie est concentrée sur l’utilisation des réseaux sociaux pour faire animer des communautés. On va mettre une méthodologie c’est EcPaRe c'est-à-dire Ecouter, Parler et Répondre en utilisant les réseaux sociaux.

Pour débiter la gestion de communauté sur les réseaux sociaux, il faut que préparer une stratégie qui permet d’identifier les cibles, pour quelles objectifs, par quelles contenus et enfin quelles diapositifs qu’on peut utiliser pour passer le message.

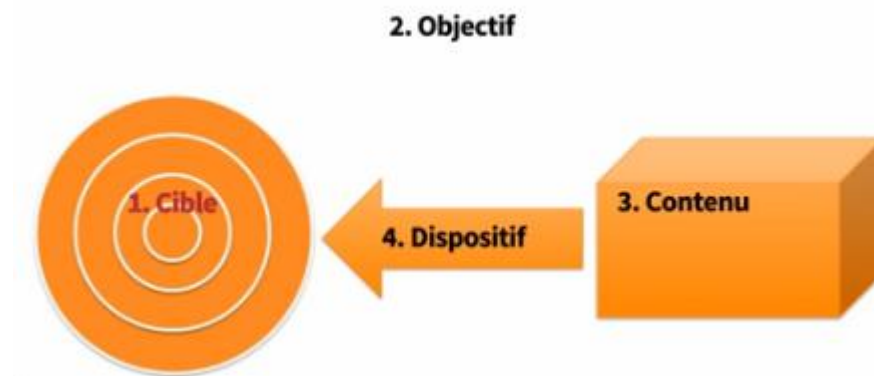


Figure 45 : Schéma de la stratégie pour un community manager.

Pour être en veille, il faut être présent dans les réseaux sociaux. On peut réaliser des tableaux de pilotage dans chaque manifestation décidée par la communauté du site.

Cible	Objectif	Contenu	Dispositif
fans de BD locaux	Les prévenir d'un évènement local	programme du festival, lieux des évènements, horaires, auteurs présents en argumentant le côté « local »	pages FB : du festival, des librairies locales, des magazines/sites de BD, des maisons d'édition. forums autour de la BD
familles locales en quête de sorties	Les prévenir d'une sortie ludique possible	programme et lieux en argumentant le côté « familial » de l'évènement.	forums locaux dédiés aux familles et aux sorties
festivalliers fans d'auteurs présents	Les prévenir de la présence de leur auteur préféré	liste des auteurs présents	pages FB des auteurs et de leurs maisons d'édition
fans de BD	Les prévenir d'un évènement BD	programme du festival	pages FB des magazines/sites de BD, des maisons d'édition.
libraires spécialisés BD	Qu'ils communiquent	auteurs présents	pages FB des librairies BD
autres festivals de BD	Qu'ils communiquent	programme, auteurs	pages FB des festivals, comptes Twitter des festivals, pages FB des billetteries (Fnac spectacles etc.)

4.5 E-mailing

L'utilisation de l'e-mailing est une méthode de marketing direct qui utilise le courrier électronique comme moyen de communication commerciale de masse pour envoyer des messages à un auditoire.

Dans cette application, on a fait intégrer un système de la newsletter qui permet de faire des campagnes. L'e-mailing permet de diffuser des offres à des clients inscrits dans la base des données. L'e-mailing peut être efficace pour les internautes qui surfent pour attirer leurs prospects.

Parmi les fonctionnalités dans cette application c'est d'exporter des données à un tableur Excel. Alors, on peut utiliser une solution d'e-mailing.

On cite la solution **MailChimp** qui est une solution populaire et gratuite.



Figure 47 : Logo de la solution d'e-mailing.

4.6 Conclusion

Le travail pour assurer l'e-réputation dans les réseaux sociaux, visibilité dans les réseaux sociaux ainsi que d'attirer les internautes exige beaucoup de précision. L'utilisation des techniques proposées par le dispositif du webmarketing assure une bonne démarche de marketing.

Conclusion générale

Ce stage a été sous plusieurs aspects riches d'enseignements.

Le stage consistait à réaliser une application pour gérer un tour opérateur. Ce stage est une opportunité pour améliorer les connaissances en création des applications web, et notamment en ce qui concerne l'utilisation poussée de technologies comme le JSF, Hibernate et JSP. Ainsi, la mise en pratique par l'utilisation d'UML et le paradigme POO.

On n'a pas complété l'application web jusqu'au bout car on a trouvé des problèmes techniques durant la phase du codage. En revanche, on a développé les principales fonctionnalités.

En conclusion, mon stage m'a permis de mettre en œuvre des compétences scolaires, professionnelles et humaines pour un sujet intéressant. J'ai de plus acquis de nouvelles compétences dans le domaine du développement web.

Annexe

- **Tour opérateur** : (de l'anglais tour operator), ou « voyageur » est un organisme chargé d'assembler plusieurs prestations de ses fournisseurs (compagnies aériennes, hôteliers, autocaristes, restaurateurs, guides, etc.) et de les vendre à un prix tout compris, c'est-à-dire un « forfait » ou « package ». Il réunit des prestations qu'il achète à prix négociés, en négociant lui-même ou par l'intermédiaire d'agences réceptives (agences de voyages sur place) et anticipe la demande de la clientèle en proposant ses offres de forfaits en brochure.

- **Cloud Computing** représente la meilleure solution pour gérer votre entreprise. Au lieu d'exécuter vos applications vous-même, c'est une plate-forme multitenant partagée qui s'en charge. Lorsque vous utilisez une application exécutée dans le cloud, il vous suffit de vous y connecter, de la personnaliser et de l'utiliser. C'est aussi simple que cela, le cloud computing.

- **Business Intelligence** : La Business Intelligence (BI), également "intelligence d'affaires" ou "informatiquedécisionnelle", englobe les solutions informatiques apportant une aide à la décision avec, en bout de chaîne, rapports et tableaux de bord de suivi à la fois analytiques et prospectifs. Le but est de consolider les informations disponibles au sein des bases de données de l'entreprise.

- **Bytecode** est destiné à regrouper des instructions exécutables par une machine virtuelle java. Par extension, il désigne un flux d'octets binaire au format d'une classe java. Ce flux est habituellement le résultat de la compilation d'un code source, ce code source n'étant pas obligatoirement écrit en langage java. Ce *bytecode* peut être exécuté sous de nombreux systèmes d'exploitation par une machine virtuelle Java.

- **JRE** est une famille de logiciels qui permet l'exécution des programmes écrits en langage de programmation Java, sur différentes plateformes informatiques. Il est distribué gratuitement par Oracle Corporation, sous forme de différentes versions destinées aux systèmes d'exploitation Windows, Mac OS X et Linux, toutes conformes aux Java Specification Requests (JSR).

- **JDK** (*Java Développement Kit, Kit de développement en Java*) Environnement de développement pour Java que l'on peut trouver chez Sun.

- **JPA** : il s'agit d'un standard faisant partie intégrante de la plate-forme Java EE, une spécification qui définit un ensemble de règles permettant la gestion de la correspondance entre des objets Java et une base de données, ou autrement formulé la gestion de la

persistance. Ce mécanisme qui gère la correspondance entre des objets d'une application et les tables d'une base de données se nomme ORM, pour « Object-Relational Mapping ».

-**GoogleBot** est un robot utilisé par le moteur de recherche Google pour parcourir le Web, dans le but d'indexer les pages et leur contenu, afin de procéder à leur référencement naturel. De plus en plus « intelligents », ces robots sont capables d'analyser le contenu de la page pour vérifier s'il correspond bien à ce qui est déclaré dans ses balises.

Application desktop VERSUS Application Web

Application Web	Application desktop
<ul style="list-style-type: none">• Hébergée sur un réseau (Internet, Intranet,...).• Manipulable par un navigateur web (Firefox, Google Chrome, Internet Explorer,...).• Utilisation des protocoles de la communication. Ex : http, Ftp, SMTP.• On parle du client léger.	<ul style="list-style-type: none">• Installée localement.• On parle du client lourd.

Tableau 1 : Différence entre une application web et une application desktop.

Bibliographies ET nethographie

Les ouvrages :

[1] **Le projet du Louvois** pour définir la complexité.

Les sites internet :

- www.developpez.com visité le 15/02/2015.
- www.compta-online.com visité le 16/02/2015.
- www.bittle-solutions.com visité le 17/02/2015.
- www.relationclients.com visité le 19/02/2015.
- www.petite-entreprise.net visité le 22/02/2015.
- www.cnil.fr visité le 22/02/2015.
- www.openclassrooms.com visité le 20/04/2015.
- www.eclipse.org/downloads visité le 23/04/2015.
- www.mkyong.com visité le 25/04/2015.
- www.wikipedia.org visité le 29/04/2015.
- www.jmdoudoux.fr visité le 05/05/2015.
- www.isi.rnu.tn visité le 20/05/2015.
- www.mes.tn visité le 22/05/2015.

Résumé

Notre projet de fin d'études s'est déroulé au sein de GWE, Global Web Engineering. Pour réaliser cette application, nous faisons subdiviser en 3 phases : spécification des besoins, conception, réalisation avec des tests unitaire en parallèle.

La solution adoptée consiste à réaliser une application web pour gérer un tour opérateur. Cette application intègre une solution entreprise qui est le Business Intelligence. La méthodologie la mieux adaptée pour concevoir l'application, est UML (Unified Modeling Language).

Le travail a été réalisé dans un environnement libre (Eclipse et GlassFish) à l'aide de la plateforme JEE.

Mots clés : JEE, JSF, Hibernate, JSP, Servlets, Business Intelligence.

Abstract

Our project of end of studies proceeded in GWE- Global Web Engineering. To realize this application, we divide into 3 phases: requirements specification, design and coding with unit tests in parallel.

The solution adopted is to make a web application for managing a tour operator. This application integrated an enterprise solution which is the business intelligence. The methodology is the best adapted to design this application, is UML (Unified Modeling Language).

The work was carried out in a free environment (Eclipse and GlassFish) using the JEE platform.

Keywords: JEE, JSF, Hibernate, JSP, Servlets, Business Intelligence.

التلخيص

مشروعنا لنهاية الدراسة، قد تم إنجازه في مؤسسة هندسة الويب العالمية (جي دوبليف إي). قسم هذا المشروع إلى ثلاث مراحل : تحديد الحاجة، التصميم و الانجاز مع اختبارات.

يتمثل هذا المشروع في إعداد تطبيق في الويب للتصرف في وكالة للمرشدين السياحيين. في هذا التطبيق، UML قمنا بادمج حل من حلول مؤسساتية و هي ذكاء الأعمال. استعملنا في هذا التطبيق طريقة لتصميمها.

هذا المشروع اعتمد على برامج مجانية و هي اكليبس و جلفش مع استعمال ال جي إي إي.

الكلمات المفاتيح : جي إي إي، جي أس إف، هيرنت، جي أس بي، سرفلت، ذكاء الأعمال