

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABDERRAHMANE MIRA DE BÉJAÏA



FACULTÉ DES SCIENCES EXACTES
DÉPARTEMENT D'INFORMATIQUE
MÉMOIRE DE MASTER PROFESSIONNEL
OPTION : GÉNIE LOGICIEL

Thème

CONCEPTION ET RÉALISATION
D'UNE APPLICATION MOBILE POUR
LE SERVICE DE TOURISME, CAS
D'ÉTUDE "WILAYA DE BEJAIA"

Présenté par :

MOUSSOUNI ZOBIR ET RAMDANI MASSINISSA

Soutenu devant le jury composé de :

<i>Président</i>	Mr SLIMANI H.	U. A/MIRA BÉJAÏA
<i>Examinatrice</i>	Mme METIDJI R.	U. A/MIRA BÉJAÏA
<i>Examineur</i>	Mr ALLEM K.	U. A/MIRA BÉJAÏA
<i>Encadrant</i>	Mme EL BOUHISSI H.	U. A/MIRA BÉJAÏA

Promotion 2016 – 2017

DÉDICACE

Nous dédions ce modeste travail et notre profonde gratitude :

À Nos très chers parents qui nous ont fourni au quotidien un soutien et une confiance sans faille et de ce fait, nous ne saurions exprimer notre gratitude seulement par des mots. Que dieu vous protège et vous garde pour nous.

À nos chers frères et nos précieuses sœurs, les mots ne peuvent résumer notre reconnaissance et notre amour à votre égard.

À tous les membres de nos famille

À nos adorables amies, pour votre fidélité et votre soutien

À tous nos amis avec lesquels nous avons partagé nos moments de joie et de bonheur.

À tout nos enseignants pour votre soutien, votre enseignement et vos conseils tout au long de notre parcours éducatif et professionnel.

Que toute personne nous ayant aidé de près ou de loin, trouve ici l'expression de notre reconnaissance.

REMERCIEMENTS

Nous remercions Dieu tout puissant qui a été toujours avec nous et qui nous a aidé tout au long de notre parcours.

Nous tenons à exprimer notre sincère reconnaissance et notre profonde gratitude aux personnes suivantes qui ont aidé à la contribution de ce mémoire :

À Madame **Houda EL BOUHISSI**, notre encadrant, pour son aide, ses conseils et son encouragement durant la réalisation de ce travail.

Nos vifs remerciements s'adressent aux membres du jury, pour l'honneur qu'ils nous ont fait en examinant ce mémoire de fin d'études, soyez assurés de notre respectueuse considération.

À tous les enseignants du département informatique et surtout nos enseignants qui par leur engagement scientifique et éducatif, durant ces cinq années d'études, ont été pour nous une source d'inspiration.

Nous présentons nos sincères remerciements à Monsieur **Mourad BECHIRI**, le directeur de la direction de tourisme de la wilaya de Bejaia pour son accueil au sein de son établissement.

À tout le personnel de la direction de tourisme de la wilaya de Béjaia, pour leur compréhension, leur aide et leur disponibilité durant la période de stage.

TABLE DES MATIÈRES

Table des Matières	i
Table des Figures	v
Liste des Tableaux	vii
Glossaire	viii
Introduction Générale	1
1 Présentation de l'organisme d'accueil et methodologie de conception	3
1.1 Présentation de la wilaya de Bejaia	4
1.1.1 Historique	4
1.1.2 Atouts touristiques	4
1.1.3 Organigramme de l'organisme d'accueil	5
1.2 Problématique	6
1.3 Solutions proposées	6
1.4 Méthodologie de conception	6
1.4.1 Processus de développement	6
1.4.2 Processus UP (Unified Process)	7
1.4.3 Principes fondamentaux du Processus Unifié(UP)	7
1.4.4 Phases et disciplines d'UP	7
1.5 Un processus de modélisation avec UML	9
1.5.1 UML (Unified Modeling Language)	9
1.5.2 Bases d'UML	10
1.5.3 Différents types de diagrammes d'UML	10
2 L'univers Android	12
2.1 L'informatique mobile	13
2.1.1 Présentation	13

2.1.2	Les terminaux mobiles	13
2.2	Les applications mobiles	15
2.2.1	Définition	15
2.2.2	Les particularités d'une application mobile	15
2.2.3	Domaine des applications mobiles	15
2.2.4	Les différents types d'application mobile	16
2.3	Le système d'exploitation Android	16
2.3.1	Présentation	16
2.3.2	Historique	17
2.3.3	Architecture	18
2.4	Les applications Android	20
2.4.1	Présentation	20
2.4.2	État d'une activité (application)	20
2.4.3	Cycle de vie d'une activité Android	20
3	Spécification et analyse des besoins	23
3.1	Spécification des besoins	24
3.1.1	Besoin fonctionnels	24
3.1.2	Besoins non fonctionnels	24
3.2	Analyse des besoins	25
3.2.1	Identification des acteurs	25
3.2.2	Modélisation du contexte	26
3.2.3	Identification des cas d'utilisation	27
3.2.4	Diagrammes des cas d'utilisation et leurs descriptions	28
3.3	Conclusion	36
4	Conception	37
4.1	Diagrammes de séquences des cas d'utilisation	39
4.1.1	Diagramme de séquence du cas d'utilisation "s'authentifier"	39
4.1.2	Diagramme de séquence du cas d'utilisation "Gérer ServiceTouristique"	40
4.1.3	Diagramme de séquence du cas d'utilisation "Gérer un Utilisateur"	41
4.1.4	Diagramme de séquence du cas d'utilisation "Gérer une Publication"	42
4.1.5	Diagramme de séquence du cas d'utilisation "Gérer une Statistique"	43
4.1.6	Diagramme de séquence du cas d'utilisation "Suivre une Statistique"	44
4.1.7	Diagramme de séquence du cas d'utilisation "Consulter un ServiceTouristique"	45
4.1.8	Diagramme de séquence du cas d'utilisation "s'inscrire"	46
4.2	Patrons de conception (Design Pattern)	47

4.2.1	Patron client/serveur à 3-tiers	47
4.2.2	Patron Modèle-vue-contrôleur	48
4.3	Diagramme de classe de conception	49
4.4	Passage au modèle relationnel	50
4.4.1	Règles de passage au modèle relationnel	50
4.4.2	Modèle logique des données	51
5	Réalisation et développement	53
5.1	Environnement de développement	54
5.1.1	Android Studio	54
5.1.2	Eclipse	54
5.1.3	WampServer	55
5.1.4	PhpMyAdmin	55
5.1.5	Apache	55
5.2	Langages de développement	55
5.2.1	JAVA	55
5.2.2	PHP	55
5.2.3	SQL	56
5.2.4	XML	56
5.2.5	JSON	56
5.3	Les services web	57
5.3.1	Définition d'un service web	57
5.3.2	Service web utilisé dans notre application	57
5.4	Schéma de navigation de notre système	60
5.5	Quelques interfaces de notre système	60
5.5.1	Interfaces de l'application Android	60
5.5.2	Interfaces de l'application d'administrateur	68
	Conclusion Générale et Perspectives	70
	Bibliographie	73
	Annexes	73
A	Annexe	
	Diagrammes de séquence système	i
A.1	Diagramme de séquence système " Gérer un Utilisateur "	ii
A.2	Diagramme de séquence système " Gérer un ServiceTouristique "	iii
A.3	Diagramme de séquence système " Suivre une Statistique "	iv

TABLE DES MATIÈRES

A.4	Diagramme de séquence système " Gérer une Statistique "	v
A.5	Diagramme de séquence système " Consulter un ServiceTouristique "	vi
A.6	Diagramme de séquence système "s'inscrire"	vii
A.7	Diagramme de séquence système " s'authentifier "	viii
A.8	Diagramme de séquence système " Gérer une Publication "	ix
B Annexe		
Schémas de navigation du système		x
B.1	Côté touriste	xi
B.2	Côté Responsable agence de voyage et Responsable association touristique	xii
B.3	Côté Responsable d'hôtel	xiii
B.4	Côté administrateur	xiv
B.5	Côté responsable de service tourisme	xv

Table des figures

1.1	Organigramme de la direction de tourisme "Bejaia"	5
1.2	Phases et Itérations d'UP	8
1.3	Historique d'UML	10
1.4	Types de diagrammes d'UML	11
2.1	Un assistant personnel	13
2.2	Les smartphones	14
2.3	Les tablettes	14
2.4	L'architecture d'un système Android	18
2.5	Cycle de vie d'une activité Android	22
3.1	Diagramme de contexte dynamique	26
3.2	Diagramme de cas d'utilisation global	28
3.3	Diagramme de cas d'utilisation associé à " Administrateur "	29
3.4	Diagramme de cas d'utilisation associé au " Responsable ServiceTourisme " .	30
3.5	Diagramme de cas d'utilisation associé au " Responsable AgenceVoyage " . .	31
3.6	Diagramme de cas d'utilisation associé au " Responsable AssociationTouristique"	32
3.7	Diagramme de cas d'utilisation associé au " Responsable Hôtel "	32
3.8	Diagramme de cas d'utilisation associé au " Touriste "	33
4.1	Diagramme de séquence du cas d'utilisation " s'authentifier "	39
4.2	Diagramme de séquence du cas d'utilisation " Gérer un ServiceTouristique " .	40
4.3	Diagramme de séquence du cas d'utilisation "Gérer un Utilisateur"	41
4.4	Diagramme de séquence du cas d'utilisation "Gérer une Publication"	42
4.5	Diagramme de séquence du cas d'utilisation "Gérer une Statistique"	43
4.6	Diagramme de séquence du cas d'utilisation "Suivre une Statistique"	44
4.7	Diagramme de séquence du cas d'utilisation "Consulter une ServiceTouristique"	45
4.8	Diagramme de séquence du cas d'utilisation "s'inscrire"	46
4.9	Architecture 3- tiers	47

TABLE DES FIGURES

4.10	Patron MVC (Modèle, Vue, Contrôleur)	48
4.11	Diagramme de classe de conception	49
5.1	Exemple d'un format JSON	58
5.2	Carte de Google map.	59
5.3	Interface d'authentification	60
5.4	Interface d'inscription (Responsable de l'association touristique)	61
5.5	Interface d'accueil	62
5.6	Interface service touristique (Site touristique)	63
5.7	Interface détail de service touristique (Site touristique)	64
5.8	Interface localisation de service touristique (Site touristique)	65
5.9	Interface gérer statistique	66
5.10	Interfaces gérer publication	67
5.11	Interface d'authentification	68
5.12	Interface gérer un service touristique (Restaurant)	68
5.13	Interface gérer les utilisateurs	69
A.1	Diagramme de séquence système " Gérer un Utilisateur "	ii
A.2	Diagramme de séquence système " Gérer un ServiceTouristique "	iii
A.3	Diagramme de séquence système " Suivre une Statistique "	iv
A.4	Diagramme de séquence système " Gérer une Statistique "	v
A.5	Diagramme de séquence système " Consulter un ServiceTouristique "	vi
A.6	Diagramme de séquence système " s'inscrire "	vii
A.7	Diagramme de séquence système " s'authentifier "	viii
A.8	Diagramme de séquence système " Gérer une Publication "	ix
B.1	Schémas de navigation de notre système "côté touriste"	xi
B.2	Schémas de navigation de notre système "côté responsable agence de voyage et responsable association touristique"	xii
B.3	Schémas de navigation de notre système "côté responsable d'hôtel"	xiii
B.4	Schémas de navigation de notre système "côté administrateur"	xiv
B.5	Schémas de navigation de notre système "côté responsable de service tourisme"	xv

Liste des tableaux

2.1	Historique des versions du système Android	17
3.1	Message échangés entre les acteurs et le système	26
3.2	Cas d'utilisation associés au système	27
3.3	Description de cas d'utilisation " Gérer un Utilisateur "	29
3.4	Description de cas d'utilisation " Gérer un ServiceTouristique"	30
3.5	Description de cas d'utilisation " suivre une statistique "	31
3.6	Description de cas d'utilisation " Gérer une Statistique "	33
3.7	Description de cas d'utilisation " Consulter un ServiceTouristique "	34
3.8	Description de cas d'utilisation " s'inscrire "	34
3.9	Description de cas d'utilisation " s'authentifier "	35
3.10	Description de cas d'utilisation " Gérer une publication "	36

GLOSSAIRE

-A-

Android Market : a été remplacé en mars 2012 (pour la France) par Google Play. Il s'agit donc de l'ancien nom du magasin d'application proposé par Google

ADT : Android Development Tools

APK : Android Package

API : Application Programming Interface

-B-

Back stack : c'est une pile où on stock les activités lancé

-D-

Dalvik : est la machine virtuelle pour le langage Java des applications du système d'exploitation Android de Google. Le nom provient d'une ville d'Islande, d'où le créateur est originaire. Dalvik est produit par l'Open Handset Alliance

-H-

HTTP : (Hypertext Transfer Protocol) C'est le protocole de transfert sur internet le plus courant.

-I-

IDE : (Integrated Development Environment) est une interface qui permet de développer, compiler et exécuter un programme dans un langage donné.

IOS : (Internetworking Operating System) est un système d'exploitation et plate-forme logicielle pour smartphones et tablettes créé et utilisé exclusivement par Apple dont la version 1.0 est sortie le 29 Juin 2007 avec le lancement de l'iPhone.

-J-

JSP : (Java Server Pages) est une technique basée sur Java qui permet aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de page web.

JSON : (JavaScript Object Notation) est un format léger d'échange de données.

JDK : (Java Development Kit) désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en bytecode destiné à la machine virtuelle Java.

-M-

MySQL : est un serveur de bases de données relationnelles Open Source.

MVC : (Modele Vue Controler) est un patron de conception, son but est de séparer l'interface, l'action et l'accès aux données.

-O-

OS : (Operating system) est un ensemble de programmes qui dirige l'utilisation des ressources d'un ordinateur

-P-

PDA :(Personal Digital Assistant) ou assistant personnel ou ordinateur de poche en français.

Plugin : est un programme informatique conçu pour ajouter des fonctionnalités à un autre logiciel

-S-

SDK : (Software Development Kit) désigne un ensemble d'outils utilisés par les développeurs pour le développement d'un logiciel destiné à une plateforme déterminée (Linux, Windows, Android, etc.)

SGL : (Skia Graphics Library), Bibliothèque graphique 2D.

SGBD : (Système de Gestion de Base de Données) est un logiciel système conçu pour créer et gérer des bases de données.

SWT : (Standard Widget Toolkit) est une bibliothèque graphique libre pour Java,

SQL : (Structured Query Language) est un langage permettant de communiquer avec une base de données.

SOAP : (Simple Object Access Protocol) est un protocole qui permet la transmission de messages entre objets distants.

-U-

UP : (Unified Process) Le processus Unifié est un processus de réalisation ou d'évolution de logiciel entièrement basé sur UML

UML : (Unified Modeling Language) se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes.

UDDI : (Universal Description Discovery and Integration) est un annuaire de services fondé sur XML et plus particulièrement destiné aux services Web.

-W-

WSDL : (Web Services Description Language) est une grammaire XML permettant de décrire un service web.

-X-

XML : (eXtensible Markup Language) est un langage informatique qui sert à enregistrer des données textuelles.

INTRODUCTION GÉNÉRALE

Il y a environ vingt-sept ans, l'invention de l'internet a révolutionné la façon de penser et de vivre dans le monde, elle a permis aux consommateurs de faire des transactions, et accomplir leurs tâches sans devoir se déplacer physiquement. Une dizaine d'années après, cette innovation est suivie par l'apparition de la technologie mobile qui a pris une place importante dans notre société, les assistants personnels (PDA), téléphones cellulaires, smartphones, tablettes, etc., et aussi les moyens de connexion comme les réseaux sans fil (Wifi, GPRS et d'autres) ont permis de suivre et accéder aux informations dont on a besoin par tout où il y a une couverture réseau, et cela se fait à l'aide des applications mobiles, ce qui nous a poussé et motivé à développer une application mobile.

Bejaia que certains qualifient de la perle de l'Afrique du Nord, est très connue en tant que zone touristique, elle est visé par des touristes venant partout dans le monde, ces touristes ont besoin de connaître les hôtels, les sites touristiques et toutes les autres services touristiques et leur localisation.

Pour remédier à ça, nous avons pensé à réaliser une application mobile pour le tourisme à Bejaia. Mais peut-on réellement créer un tel système ? Si c'est oui comment et qu'elle est la procédure à suivre pour le faire. La réponse à ses questions sera détaillée dans les chapitres suivants.

Dans le premier chapitre, nous allons présenter l'organisme d'accueil en faisant un aperçu rapide de la wilaya de Bejaïa et son historique. Puis nous définissons la méthodologie à suivre afin de réaliser notre système.

Ensuite, dans le second, nous allons introduire l'univers Android, en décrivant c'est quoi l'informatique mobile, le système Android et les applications Android.

Le troisième chapitre, focalise sur la première étape du processus qu'on a choisi pour le développement de notre application qui est la phase d'analyse des besoins. Dans cette partie nous allons identifier les acteurs de notre système et les besoins du client qui deviendront ensuite les fonctionnalités de notre application.

Puis dans le chapitre qui suit, nous illustrons la deuxième étape qui est la phase de conception, et nous détaillerons les besoins identifiés dans le troisième chapitre.

En fin, le cinquième chapitre, dédié à la réalisation de l'application ou nous allons présenter l'environnement de développement, les outils et langages de programmation utilisés dans notre travail, puis montrer quelques interfaces graphiques de notre application réalisée. Et nous concluons notre travail par une conclusion générale et quelques perspectives.

1

PRÉSENTATION DE L'ORGANISME D'ACCUEIL ET METHODOLOGIE DE CONCEPTION

Introduction

Dans ce chapitre nous allons présenter l'organisme d'accueil, ensuite nous citons les problèmes à résoudre, et les solutions proposées. puis nous décrivons la méthodologie de travail adoptée pour développer et mettre en œuvre notre solution.

1.1 Présentation de la wilaya de Bejaia

1.1.1 Historique

Bejaia que certains la qualifie de la perle de l'Afrique du Nord a porté plusieurs noms et ces derniers correspondent à chaque époque par laquelle elle est passée en effet **Saldea** à l'époque Romaine, **An Naciria** à l'époque Arabo-musulman et **Bougie** à l'époque coloniale.

Elle est limitée Au Nord par la mer Méditerranéenne, à l'Ouest par les Wilayas de Tizi-Ouzou et Bouira, à l'Est par la Wilaya de Jijel, et enfin Sétif et Bordj Bou-Argeridj au Sud. Sa superficie est 3.261,26 Km².

Spécificités

Bejaïa et ses environs sont l'un des plus beaux paysages de la côte magrébine et de la méditerranée, Elle jouit d'une situation géographique privilégiée, une façade maritime de plus de 100Km surplombée par des massifs montagneux, pourvus de forêts denses et créatifs, des ressources naturelles et historiques, des sites touristiques incomparables, des monuments et grottes féériques.

Géographie

Bejaïa est limitée à l'est et au sud-est par les Bâbord auxquels viennent se souder les Bibans au sud, la mer Méditerranée au Nord et les Crêtes du Djurdjura à l'ouest complète les limites géographiques de la Wilaya. Bejaïa est caractérisée par la prédominance des Zones montagneuses. Elle se présente comme une masse montagneuse compacte et bosselée, traversée par le couloir formé par la vallée de la Soummam.

1.1.2 Atouts touristiques

La wilaya de Bejaia considérée comme une région de fascination, d'inspiration, de savoir et de convoitises, elle compte des richesses considérables notamment :

- **Sites naturels** : tel que le Cap Carbon, lance des Aiguades, Mont Gouraya, le Pic de Singe, les Grottes féériques, les Cascades, les Gorges, les Lacs...etc.
- **Sites culturels** : comme le fort de Gouraya, le château de la Comtesse, la Citadelle "la Casbah"...etc.

1.1.3 Organigramme de l'organisme d'accueil

La figure suivante représente l'organigramme de la direction de tourisme de la wilaya de Bejaïa.

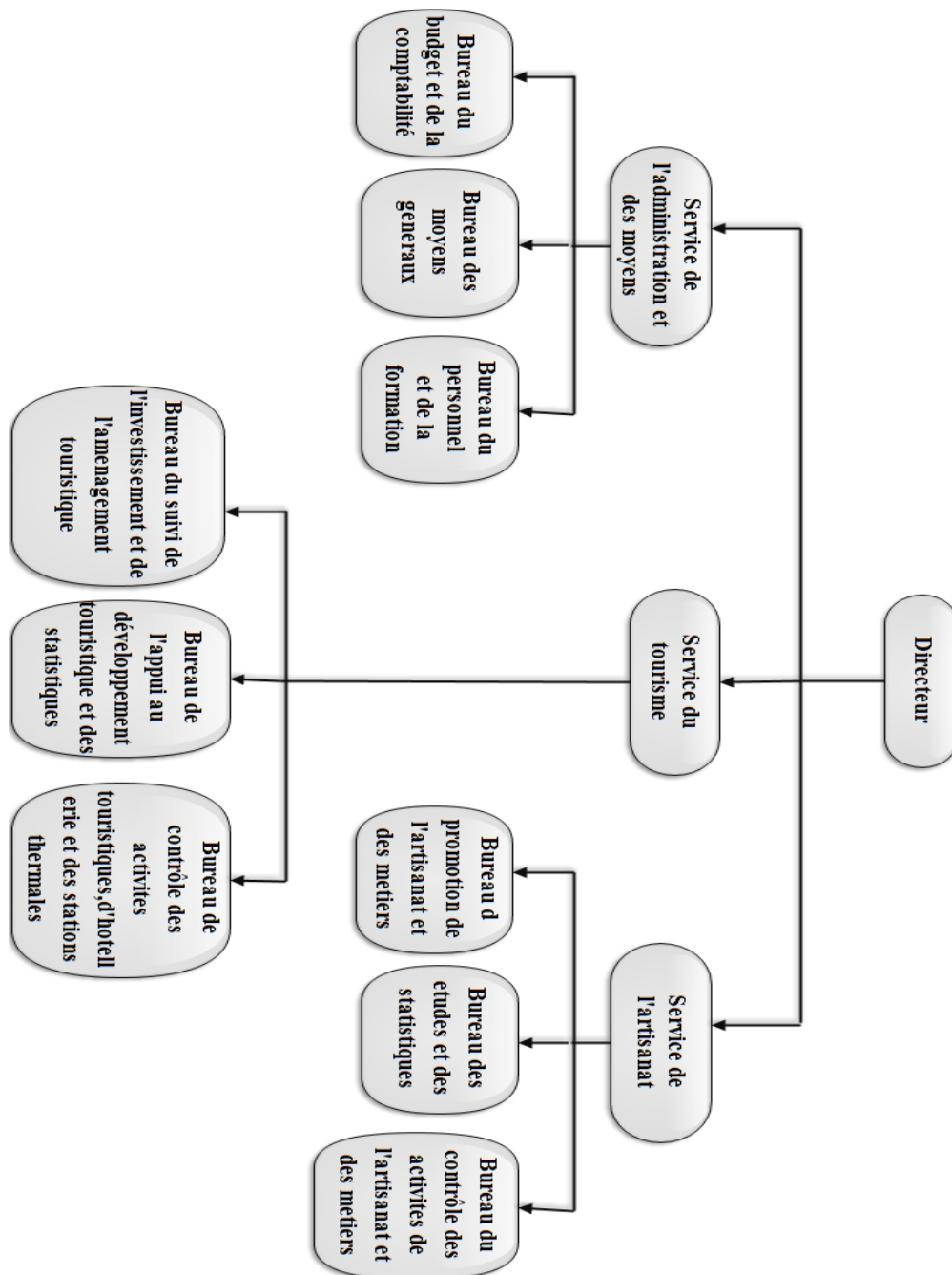


FIGURE 1.1 – Organigramme de la direction de tourisme "Bejaïa"

1.2 Problématique

- Le service de tourisme a manifesté des difficultés dans la gestion des services touristiques (agences de voyage, associations touristiques, restaurants, hôtels, campings, plages sources thermales, lieux touristiques), et aussi la recherche et les calculs des statistiques qui prend beaucoup de temps, avec le risque d'erreurs.
Les touristes rencontrent des problèmes lors de la recherche et la localisation des services touristiques dont elle dispose la wilaya de Bejaïa.
- L'informatique est presque devenue indispensable au niveau de tous les domaines dans notre vie courante, pour cela il est utile de réaliser une application pour le service de tourisme. Mais peut-on vraiment intégrer la technologie et l'informatique à ce domaine? Existe-t-il un moyen pour réaliser une application qui répond aux besoins de ce service et de satisfaire leurs attentes en terme de fonctionnalités et d'ergonomie? Si c'est le cas, comment et Quelle est la méthode ou processus de développement à suivre pour réaliser un tel système?

1.3 Solutions proposées

Pendant la période de stage pratique passé au niveau du service de tourisme, nous avons pu comprendre l'organisme et le fonctionnement de ce service, ce qui nous a permis après une étude et analyse préliminaire d'opter pour les propositions suivantes qui pourraient bien répondre aux questions posées précédemment.

- Conception et réalisation d'un système de gestion pour le service de tourisme qui se compose de deux applications, une sous android pour les touristes, et une autre pour l'administration en basant sur une architecture 3-tiers.

1.4 Méthodologie de conception

Le choix du processus de développement à utiliser se fait selon plusieurs critères tels que la complexité et le type de projet, le délai de livraison, le coût de développement et les compétences de l'équipe.

1.4.1 Processus de développement

Un processus définit une séquence d'étapes, en partie ordonnées, qui concourent à l'obtention d'un système ou à l'évaluation d'un système existant.

L'objet d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles. [1]

Généralement Il existe plusieurs processus de développement d'applications, après avoir parcouru ces processus et leurs caractéristiques, et en ayant connaissance de notre projet et ses besoins, nous avons opté pour le processus UP (Unified Process) qui semble adéquat pour ce genre d'application.

1.4.2 Processus UP (Unified Process)

Le processus Unifié est un processus de réalisation ou d'évolution de logiciel entièrement basé sur UML, d'où l'intérêt de le mettre en œuvre dans notre projet. Le processus Unifié est constitué d'un ensemble de directives afin de produire un logiciel à partir des exigences. Chaque directive définit qui fait quoi et à quel moment. Il permet donc de structurer les différentes étapes d'un projet informatique. [2]

1.4.3 Principes fondamentaux du Processus Unifié(UP)

Le processus Unifié (UP) est un processus de développement logiciel "itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques". [1]

- **Itératif et incrémental** : Le projet est découpé en itérations de courte durée (environ 1 mois) qui aident à mieux suivre l'avancement global. A la fin de chaque itération, une partie exécutable du système final est produite, de façon incrémentale. [3]
- **Centré sur l'architecture** : Tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML et pas seulement documentée en texte. [3]
- **Piloté par les risques** : Les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations. [3]
- **Conduit par les cas d'utilisation** : Le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et priorisés. [3]

1.4.4 Phases et disciplines d'UP

La gestion d'un tel processus est organisée suivant les quatre phases suivantes : initialisation, élaboration, construction et transition.

PRÉSENTATION DE L'ORGANISME D'ACCUEIL

- **La phase d'initialisation** : conduit à définir la "vision" du projet, sa portée, sa faisabilité, son business case, afin de pouvoir décider au mieux de sa poursuite ou de son arrêt. [3]
- **La phase d'élaboration** : Poursuit trois objectifs principaux en parallèle : Identifier et décrire la majeure partie des besoins des utilisateurs, construire (et pas seulement décrire dans un document !) l'architecture de base du système, et enlever les risques majeurs du projet. [3]
- **La phase de construction** : Consiste surtout à concevoir et implémenter l'ensemble des éléments opérationnels (autres que ceux de l'architecture de base). C'est la phase la plus consommatrice en ressources et en effort. [3]
- **La phase de transition** : Permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux. Les mots clés sont : conversion des données, formation des utilisateurs, déploiement, bêta-tests. [3]

La figure suivante montre les différentes phases et les itérations du processus UP.

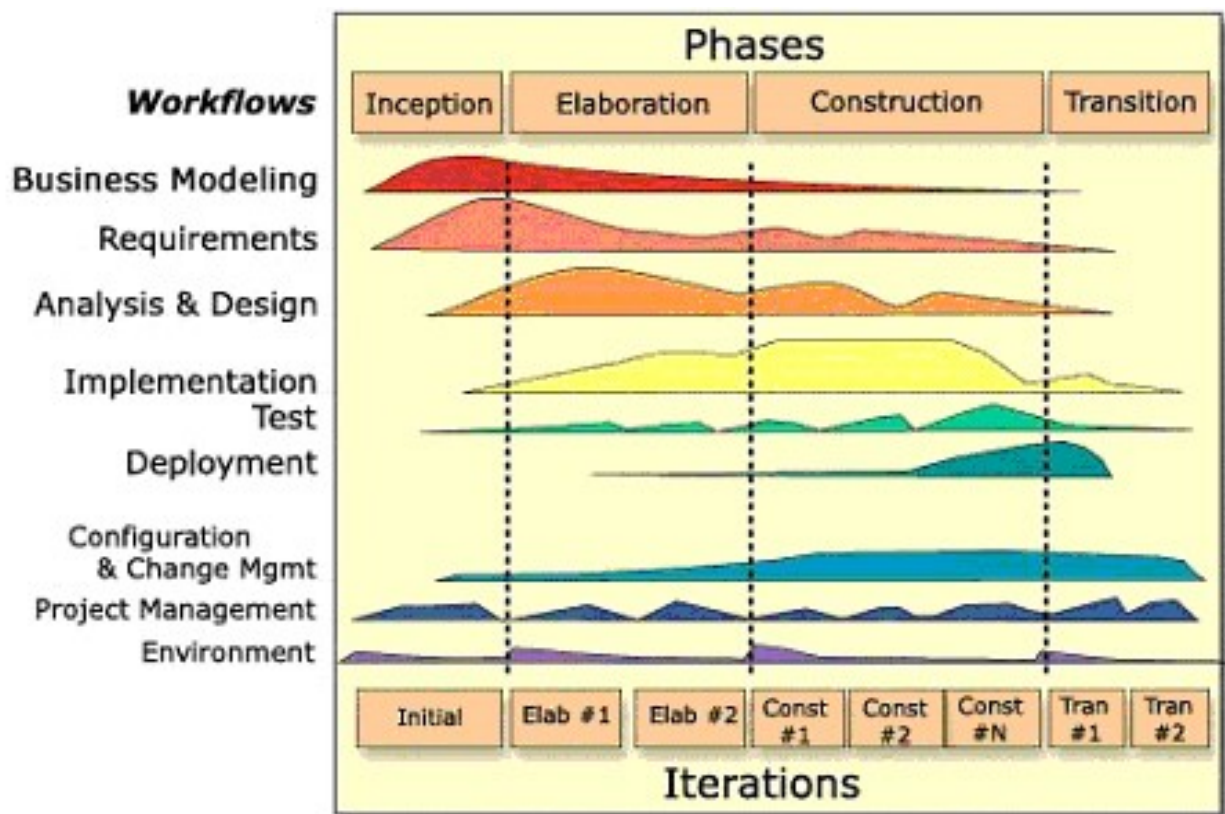


FIGURE 1.2 – Phases et Itérations d'UP

Les phases sont elle-même divisées en itérations et produisent des versions du produit. Chaque itération produit une version du système (un jalon mineur) tandis que les phases

définissent les grandes étapes du développement (les jalons majeurs), qui contrôlent ainsi le nombre d'itérations. [4]

Chaque itération couvre les activités traditionnelles (workflow) d'analyse des besoins, d'analyse, de conception, de réalisation et de test. En fait, les parts respectives de ces activités évoluent selon la phase considérée, les activités en amont sont plus poussées dans les itérations des premières phases, les activités en aval sont plus approfondies dans les itérations des phases ultérieures. Chaque itération donne lieu à un produit résultat. Plus le projet est grand, plus il y a d'itérations dans une phase. [4]

À ces activités correspondent des modèles (cas d'utilisation, analyse, conception, déploiement, implantation et test) décrits avec la notation unifiée, une vue architecturale de ces modèles et des descriptions complémentaires. Nous retrouvons alors la notation UML, qui sert à décrire ces modèles. Nous traçons maintenant les grandes lignes de la corrélation entre le processus et la notation unifiée. [4]

En plus des activités de développement, on trouve des activités support tel que la gestion de configuration et versions, la gestion de projet (organisation, risques, planification) et l'environnement du projet (support et méthode).

De ce fait UP est une méthode complète pour aborder un projet. Tests et qualité sont pris en compte par le processus. [4]

1.5 Un processus de modélisation avec UML

Le processus UP est basé sur UML comme langage de modélisation.

1.5.1 UML (Unified Modeling Language)

UML se définit comme 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. [1]

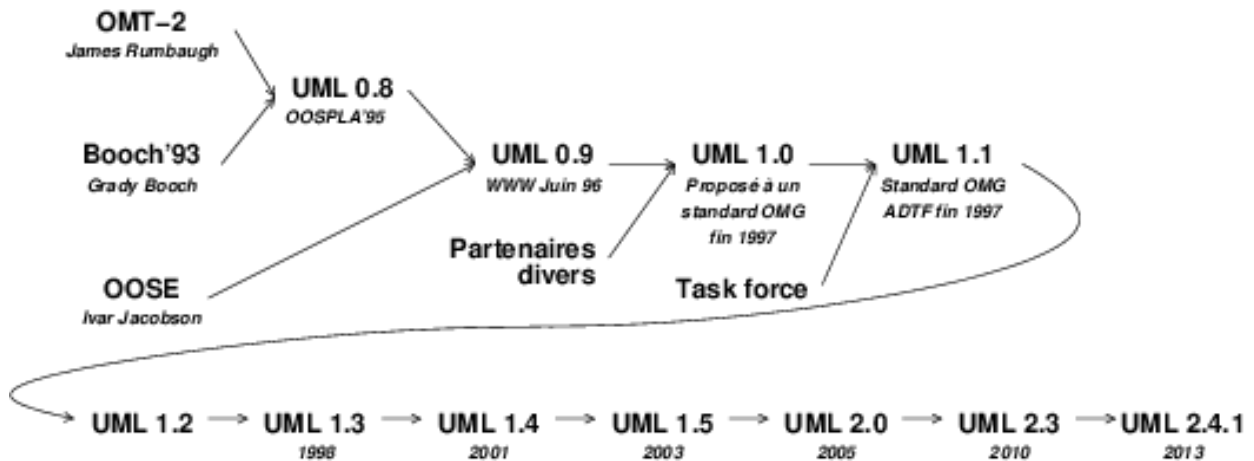


FIGURE 1.3 – Historique d’UML

1.5.2 Bases d’UML

UML unifie à la fois les notations et les concepts orientés objet. Il ne s’agit pas d’une simple notation, mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d’un langage. Ce langage est certes issu du développement logiciel mais pourrait être appliqué à toute science fondée sur la description d’un système. Dans l’immédiat, UML intéresse fortement les spécialistes de l’ingénierie système. [3]

UML unifie également les notations nécessaires aux différents activités d’un processus de développement et offre, par ce biais, le moyen d’établir le suivi des décisions prises, depuis la spécification jusqu’au codage. Dans ce cadre, un concept appartenant aux besoins des utilisateurs projette sa réalité dans le modèle de conception et dans le codage. [3]

1.5.3 Différents types de diagrammes d’UML

UML s’articule maintenant autour de 13 diagrammes différents, dont quatre nouveaux diagrammes introduits par UML 2.0. Chacun d’eux est dédié à la représentation d’un système logiciel suivant un point de vue particulier. Par ailleurs, UML modélise le système suivant deux modes de représentation : l’un concerne la structure du système pris “ au repos”, l’autre concerne sa dynamique de fonctionnement. Les deux représentations sont nécessaires et complémentaires pour schématiser la façon dont est composé le système et comment ses composants fonctionnent entre elles. [1]

La figure suivant présente les différents types de diagramme de l’UML.

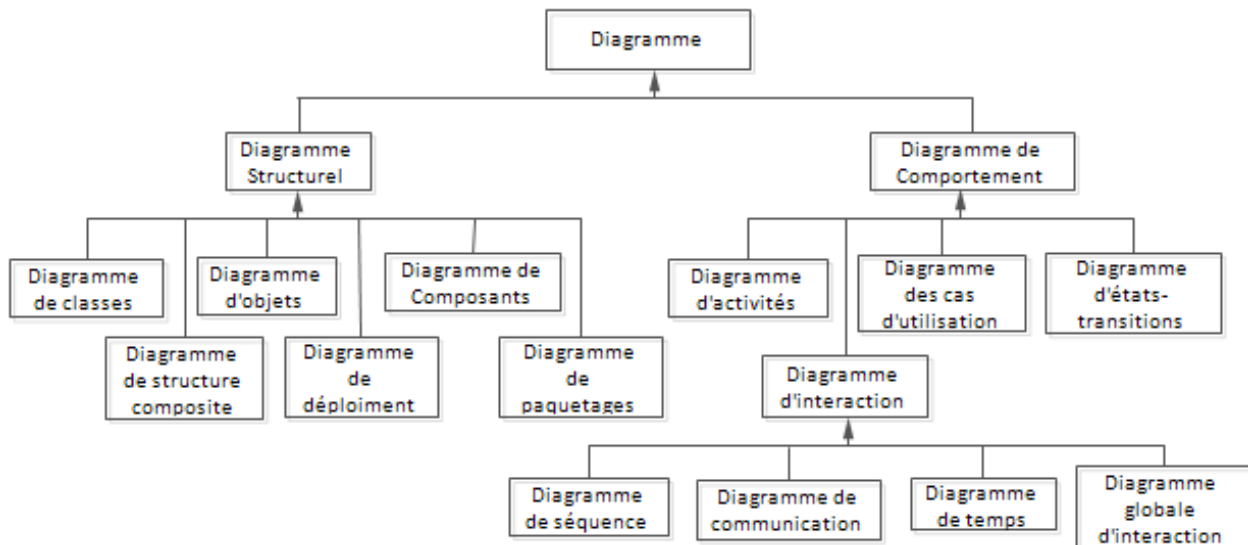


FIGURE 1.4 – Types de diagrammes d’UML

Dans notre projet nous allons utiliser seulement les diagramme suivant :

- **Diagramme de cas d’utilisation** : représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il est utilisé dans les deux étapes de capture des besoins fonctionnels et techniques. Voir le chapitre III. [1]
- **Diagramme de séquence** : est un diagramme d’interaction, il représente les échanges de messages entre objets, dans le cadre d’un fonctionnement particulier du système. Ils servent ensuite à développer en analyse les scénarios d’utilisation du système. l’utilisation de ce diagramme sera présentée au chapitre III. [1]
- **Diagramme de classe** : a toujours été le plus important dans toutes les méthodes orientés objet. c’est également celui qui contient la plus grande gamme de notations et de variantes centralise l’organisation des classes de conception, c’est lui qui se transforme le plus aisément en code. Vous trouverez l’utilisation du diagramme de classe au chapitre IV. [1]

Conclusion

Dans ce chapitre nous avons présenté notre cas d’études, en commençant par un petit historique, par la suite nous avons cité la problématique, à laquelle notre projet doit répondre, ce qui nous a permis d’avoir une idée sur les besoins qui sont un facteur important pour choisir la méthode de développement à suivre afin de réaliser notre projet “BejaiaTourisme”. Dans le chapitre suivant nous allons présenter l’univers d’Android.

2

L'UNIVERS ANDROID

Introduction

Rares sont les systèmes ayant connu une progression telle que celle que connaît actuellement le système Android.

Dans ce chapitre nous allons définir c'est quoi l'informatique mobile, et quelques notions de base, puis nous allons présenter l'univers Android tout en définissant c'est quoi le système Android, son architecture et le cycle de vie d'une application Android.

2.1 L'informatique mobile

2.1.1 Présentation

Difficile, aujourd'hui, de passer à côté de l'explosion de l'internet mobile. En l'espace de quelques mois, les nouveaux terminaux mobiles que sont les tablettes et les smartphones ont envahi notre quotidien. Cet afflux technologique change radicalement notre rapport à l'informatique en affranchissant de l'utilisation d'un micro-ordinateur classique. [5]

2.1.2 Les terminaux mobiles

Un terminal mobile est un petit appareil informatique ou de communication qu'on peut transporter avec soi dans ses déplacements et utiliser comme terminal donnant accès sans fil à un ou plusieurs réseaux. [6] Parmi les terminaux mobiles, on trouve des assistants numériques personnels (PDA), téléphones intelligents (Smartphone), des tablettes, etc.

- Les assistants personnels (PDA)



FIGURE 2.1 – Un assistant personnel

Un assistant personnel est un périphérique portable qui fonctionne comme un gestionnaire d'informations personnelles. Les PDA sont utilisés pour la navigation sur

le Web, les applications bureautique, les vidéos, les photos ou les téléphones mobiles. Les caractéristiques du modèle PDA varient, mais les fonctionnalités communes courantes incluent les écrans tactiles, la connectivité Bluetooth et Wifi, etc. Les PDA contiennent des logiciels pour synchroniser les informations. [7]

- Les smartphones



FIGURE 2.2 – Les smartphones

Un smartphone (téléphone intelligent) est un téléphone portable doté de fonctionnalités très avancées. Un smartphone typique dispose d'un écran tactile haute résolution, de la connectivité WIFI, des capacités de navigation Web et de la capacité d'accepter des applications sophistiquées. La plupart de ces appareils fonctionnent sur l'un de ces systèmes d'exploitation mobiles populaire : Android, Symbian, IOS, BlackBerry OS et Windows. [8]

- Les tablettes



FIGURE 2.3 – Les tablettes

Une tablette PC est un ordinateur portable hybride entre un assistant numérique personnel(PDA) et un ordinateur portable. Équipé d'une interface à écran tactile, possède généralement une application logicielle utilisée pour exécuter un clavier virtuel. Cependant, de nombreuses tablettes PC prennent en charge les claviers externes. Elles

ont des fonctions de navigation Web intégrées, des options de connectivité multiples, des écrans tactiles capacitifs et multimédia, y compris un support haut définition (HD). [9]

2.2 Les applications mobiles

2.2.1 Définition

Une application mobile est un type de logiciel ou programme conçu pour s'exécuter sur un appareil mobile, tel qu'un smartphone ou une tablette. Les applications mobiles servent souvent à fournir aux utilisateurs des services similaires à ceux du PC. [10]

2.2.2 Les particularités d'une application mobile

Pour répondre aux besoins et aux attentes du mobinaute, une application doit être pensée pour la mobilité. Même si le service existe déjà, par exemple en application web, l'application mobile ne doit pas être présentée comme une application Desktop (ou être copie). Tout d'abord car la taille du terminal est différent et aussi car le mobile offre d'autres possibilité que le web. Parmi les particularités d'une application mobile on trouve la performance et l'environnement. [11]

- **La performance**

Une application mobile est développée pour atteindre un niveau de qualité, en termes d'ergonomie mais aussi de performance, inégale par rapport aux autres moyens d'accéder à un service sur le mobile. Une application mobile peut être indépendante ou liée à un service web et utiliser ou non le web. [11]

- **Environnement ou absence d'internet**

Une application mobile n'a donc pas obligatoirement besoin d'internet pour fonctionner sur le téléphone. Si l'utilisateur se retrouve dans une zone privée d'internet, alors il pourra continuer à utiliser son application (même si les informations affichées ne sont pas les dernières en ligne dans le cas où l'application récupère des informations). [11]

2.2.3 Domaine des applications mobiles

Le domaine d'utilisation des applications mobiles est très vaste, on peut citer par exemple la géolocalisation, itinéraire, Paiement mobile, traitement du texte, les jeux, réseaux sociaux et éducation, etc.

2.2.4 Les différents types d'application mobile

Il existe plusieurs types d'applications mobiles qui sont les applications natives, web et hybrides.

- Les applications natives

Une application native est une application développée spécifiquement pour une seule plateforme, grâce aux outils conçus pour celle-ci. Elle est développée avec un langage spécifique à son système d'exploitation et est distribuée uniquement par l'intermédiaire de son store (AppStore IOS, PlayStore pour Android, ...). Pour développer une application pour deux plateformes différentes, il vaut mieux de développer deux applications distincts. [12]

- Les applications Web (WebApps)

En opposition à une application native, une WebApps est une application mobile développée avec les outils de développement web actuel : HTML5, CSS3 et JavaScript. C'est une application qui une fois développée est accessible et exécutable sur tous les smartphones via leur navigateur web. L'avantage de ces applications, c'est le gain de temps et d'argent réalisé grâce à leur développement unique et leur déploiement multiplateformes. Dans un cas, vous développez une seule application alors que dans l'autre, vous développez trois applications (pour Android, IOS et Windows Phone). [12]

- Les applications hybrides

Les applications hybrides sont des applications qui combinent les éléments d'une WebApps et les éléments d'une application native. Elles reposent essentiellement sur la solution Cordova/PhoneGap, cette solution sert de passerelle entre le langage web et le natif. Cette solution nous permet d'utiliser un seul et même outil pour le développement et les langages issus du développement Web pour tous les mobiles (IOS, Android et Windows Phone). [12]

2.3 Le système d'exploitation Android

2.3.1 Présentation

Android est une plate-forme intégrée pour la première fois dans un smartphone (téléphone intelligent) sorti en France en mars 2009, Android s'est depuis émancipé très rapidement pour conquérir de nombreux appareils, mobile ou non, tels que les netbooks (mini-ordinateurs), les tablettes tactiles et les télévisions connectées, au point de devenir aujourd'hui l'un des systèmes d'exploitation majeurs dans le monde. [13]

2.3.2 Historique

L'histoire d'Android commence en octobre 2003, lorsqu'Andy Rubin, Rich Miner, Nick Sears et Chris White créent la société Android à Palto Alto (Californie).

Google a racheté la société en août 2005. Deux ans plus tard, l'Open Handset Alliance est annoncée et Android devient officiellement open source.

La première version du SDK Android 1.0 sort en 2008 avec le premier téléphone sous Android (HTC Dream). En avril 2009, la version 1.5 (API 3) d'Android sort. Cette version baptisée Cupcake (petit gâteau) inaugure les nouveaux noms des versions d'Android ce qui donnera pour les futures versions comme le montre la figure dessous. [14]

Nom	N version	Date	Niveau API
N/A	1.0	Septembre 2008	1
	1.1	Février 2009	2
Cupcake	1.5	Avril 2009	3
Donut	1.6	Septembre 2009	4
Eclair	2.0-2.1	Octobre 2009	5-7
Froyo	2.2-2.2.3	Mai 2010	8
Gingerbread	2.3-2.3.7	Décembre 2010	9-10
Honeycomb[a]	3.0-3.2.6	Février 2011	11-13
Ice Cream Sandwich	4.0-4.0.4	Octobre 2011	14-15
Jelly Bean	4.1-4.3.1	Juin 2012	16-18
Kitkat	4.4.x	Octobre 2013	19-20
Lollipop	5.0-5.1.1	Octobre 2014	21-22
Marshmallow	6.0-6.0.1	Mai 2015	23
Nougat	7.0-7.1.1	Septembre 2016	24-25

TABLE 2.1 – Historique des versions du système Android

2.3.3 Architecture

Cette figure illustre les différentes couches au niveau du système android.

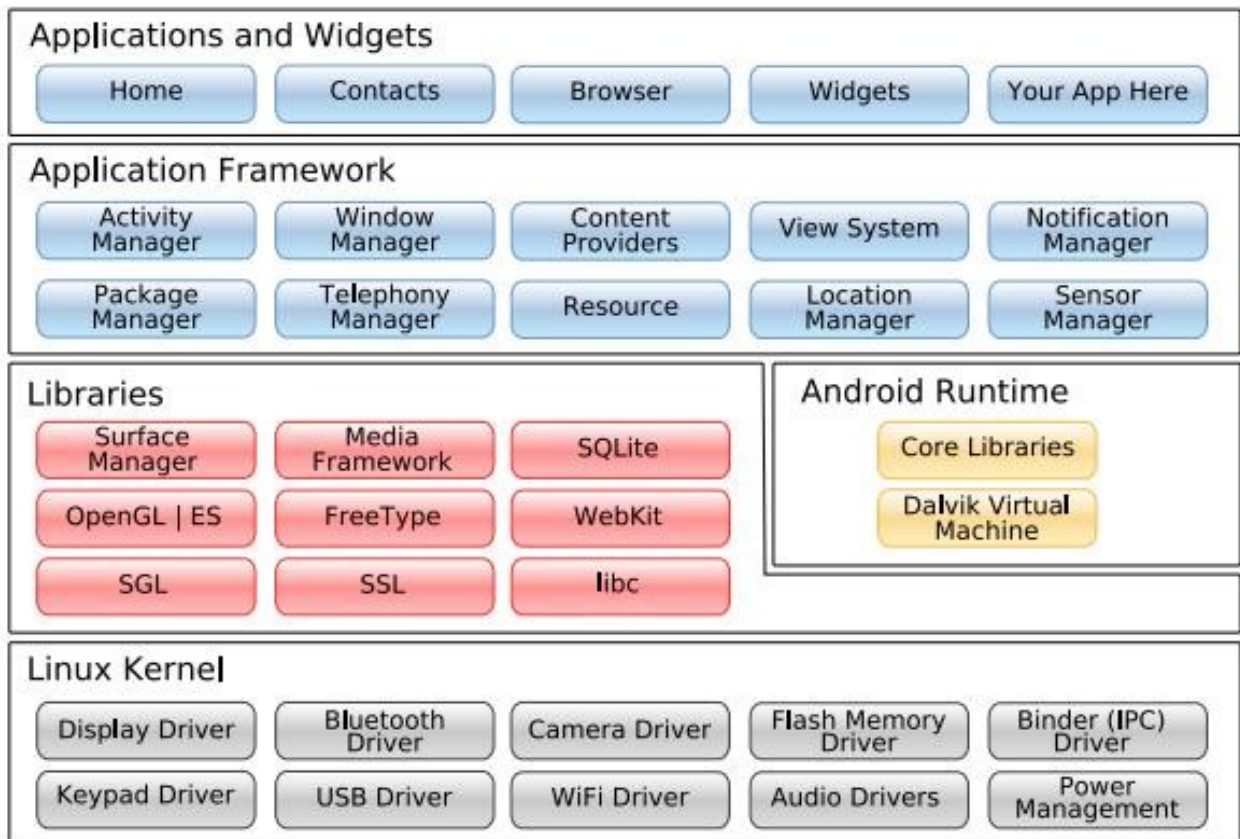


FIGURE 2.4 – L'architecture d'un système Android

Comme le montre la figure ci-dessus, Le système d'exploitation Android est basé sur une architecture à quatre niveaux.

- **Linux Kernel**

Le système d'exploitation Android s'appuie sur un noyau Linux, régulièrement mis à jour selon les versions du système : si les premières versions utilisaient la version 2.6.x du noyau Linux les actuelles dernières versions (Android 5.x) est basées sur la version 3.4.x de Linux.

Cette première couche prend en charge la gestion des couches basses (gestion des processus, de la mémoire, de la couche matérielle) ainsi que les droits utilisateurs. [15]

Au-dessus de noyau se trouve une couche d'espace utilisateur native, constituée par le binaire init (le premier processus démarré, qui fait tourner tous les autres processus), plusieurs démon et quelque centaines bibliothèque natives sont utilisées dans tout le système. [16]

- **Les librairies**

A ce niveau, figure la couche des bibliothèques principales du système fournies par des tiers. Celles-ci, de bas niveau, sont écrites en C et/ou C++. Elles fournissent des services essentiels tels que la bibliothèque d'affichage en 2D (SGL), la bibliothèque d'affichage graphique 3D (OpenGL), la bibliothèque de base de données (SQLite), la lecture et l'enregistrement audio et vidéo (Media Framework), un moteur de navigation Web (WebKit) ... [13]

- **Android Runtime (Le moteur d'exécution linux)**

Au même niveau que cet ensemble de librairies se trouve l'environnement fonctionnel d'Android basé sur une technologie Java avec une spécificité unique : sa machine virtuelle optimisée pour les applications mobiles. Cette machine virtuelle, dénommée Dalvik, est initialisée par chaque application lors de son lancement et ouvre, pour chacune, une Dalvik spécifique, ce qui rend l'OS complètement multitâche.

Dalvik est fournie avec son kit de développement spécifique ou Application Programming Interface (API) écrit en Java ; les applications Android sont donc écrites en Java, mais un java spécifique à Android. Le Java Development Kit (JDK) Android est, en fait, un sous-ensemble du JDK de SUN et intègre une partie des classes JDK SUN et JDK SWING, plus des classes spécifiques à Android. [17]

- **Framework Android**

Les fonctionnalités offertes par ces bibliothèques sont ensuite reprises et utilisées par cette couche sous forme de bibliothèques Java. Celles-ci fournissent des bibliothèques et composants réutilisables spécifiques à des domaines particuliers. On y retrouve par exemple les bibliothèques de base de données, de téléphone, localisation géographique, de communication en champ proche . [13]

- **Les applications**

Enfin, une dernière couche parachève cette structure : les applications. Ces applications peuvent être, bien sûr, les applications tierces téléchargées sur le magasin d'application officiel, mais également des applications installées par défaut, telles que l'application d'accueil (aussi appelée Launcher), le navigateur web, les applications de SMS et téléphonie, etc. Toutes ces applications sont communément développées en Java. [15]

2.4 Les applications Android

2.4.1 Présentation

Une application Android est une application mobile spécifiquement développée pour les smartphones et les tablettes utilisant le système Android. Elles sont de nature très variables tel que les applications jeux, mobile commerce, utilitaire, service d'information . [18]

Sous Android, une application est composée d'une ou plusieurs activités. Une activité est la base d'un composant pour la création d'interfaces utilisateur. Afin de faciliter la cinématique de l'application, il est préconisé de n'avoir qu'une interface visuelle par activité. [19]

2.4.2 État d'une activité (application)

Une application Android peut se trouver dans des états différents qui sont : en cours d'exécution, en pause, stopper et tuer.

- **En cours d'exécution** : l'activité se trouve au premier plan et reçoit les interactions utilisateurs. Si l'appareil a besoin de ressources, l'activité se trouvant en bas de la back stack.
- **Pause** : l'activité est visible mais l'utilisateur ne peut pas interagir avec (cachée par une boîte dialogue par exemple). La seule différence avec l'état précédent est la non-réception des évènements utilisateurs.
- **Stopper** : l'activité n'est plus visible mais toujours en cours d'exécution. Toutes les informations relatives à son exécution sont conservées en mémoire. Quand une activité passe en état stopper, vous devez sauvegarder les données importantes et arrêter tous les traitements en cours d'exécution.
- **Tuer** : l'activité est tuée, elle n'est plus en cours d'exécution et disparaît de la back stack. Toutes les données présentes en cache non sauvegardées sont perdues.

2.4.3 Cycle de vie d'une activité Android

Par défaut, chaque application Android s'exécute dans un processus séparé.

Android gère les ressources disponibles dans un appareil et peut, si besoin, fermer des applications pour libérer des ressources (hors application en cours d'exécution).

Le choix de l'application à fermer dépend fortement de l'état du processus dans lequel elle se trouve. Si Android doit choisir entre deux applications se trouvant dans le même état, il choisira celle qui se trouve dans cet état depuis plus longtemps.

Le cycle de vie d'une activité est assez complexe et sa compréhension est indispensable dans le développement Android. Le schéma ci-dessous résume ce cycle de vie.

Lors du lancement d'une activité, la méthode `onCreate` est appelée. Dans cette méthode, vous devez initialiser votre vue et lier les données à une liste. Cette méthode prend en paramètre un `Bundle` (pile) contenant l'état précédent de l'activité.

Cet appel est suivi par la méthode **`onStart`** afin de signifier le lancement effectif de l'application (elle est maintenant visible). Cette méthode peut aussi être appelée par la méthode **`onRestart`**.

Puis la méthode **`onResume`** est appelée afin d'exécuter tous les traitements nécessaires au fonctionnement de l'activité (thread, processus, traitement), initialiser des variables et les listeners. Ces traitements devront être arrêtés lors de l'appel à la méthode **`onPause`** et relancés si besoin lors d'un futur appel à la méthode **`onRésumé`**.

Après ces trois appels, l'activité est utilisable et peut recevoir les interactions utilisateurs.

Si une autre activité passe au premier plan, l'activité en cours d'exécution passera en pause. Juste avant l'appel à la méthode **`onPause`**, la méthode **`onSaveInstanceState`** est appelée afin de vous permettre de sauvegarder les informations importantes portées par l'activité. Ces informations pourront être appliquées aux futurs lancements de l'activité lors de l'appel à la méthode **`onRestoreInstanceState`** ou de l'appel à **`onCreate`**.

La méthode **`onPause`** permet de stopper tous les traitements effectués (traitement non nécessaire si l'activité n'est pas visible) par l'activité (traitement, thread, processus). Si votre activité est de nouveau visible, cela correspondra à un appel à la méthode **`onResume`**. Le passage de l'activité à l'état "stopper" correspond à un appel à la méthode **`onStop`**. Dans cette méthode, il faut arrêter tous les traitements restants. Une fois stoppée, votre activité peut :

- **Soit être relancée** : cela s'effectue par un appel à la méthode **`onRestart`** suivi du cycle de vie normal de l'activité.
- **Soit être tuée** : cela s'effectue par un appel à la méthode **`onDestroy`**, dans laquelle vous devez arrêter tous les traitements restants, fermer toutes les connexions à la base de données, tous les threads, tous les fichiers ouverts, etc. Vous pouvez provoquer l'appel à la méthode **`onDestroy`** en utilisant la méthode **`finish`**. [20]

Conclusion

Dans ce chapitre nous avons présenté brièvement l'informatique mobile et le monde Android tout en décrivant le système Android, son architecture et l'évolution de ses versions à travers le temps depuis son apparition jusqu'à la version actuel. Cela nous a aidés à bien comprendre le fonctionnement de ce système.

Le chapitre suivant sera consacré à la phase de spécification et analyse des besoins.

3

SPÉCIFICATION ET ANALYSE DES BESOINS

Introduction

Dans Ce chapitre, nous allons identifier toutes les fonctionnalités de l'application selon les exigences fonctionnelles et non fonctionnelles. En Commencant par les Besoins fonctionnels, les besoins non fonctionnels, puis les entités externes qui vont interagir avec le système, ensuite les différents cas d'utilisations et leur modélisation par des diagrammes et enfin une petite conclusion.

3.1 Spécification des besoins

La spécification des besoins est la première phase du cycle de vie d'une application. Le but de cette étape est de décrire l'application à développer.

3.1.1 Besoin fonctionnels

Dans cette partie nous détaillons les fonctionnalités, que le système doit fournir aux différents acteurs, qui se présentent comme suit :

- **Gérer les utilisateurs** : mettre à jour les utilisateurs ainsi que l'attribution des droits d'accès aux données.
- **Gérer les Services Touristiques** : mise à jour des Services Touristiques (agences de voyage, associations touristiques, hôtels, restaurants, campings, plages, lieux touristiques, sources thermales).
- **Gérer les publications** : possibilité d'ajout, modification et suppression des publications.
- **Gérer les statistiques** : mise à jour et consultation des statistiques des hôtels.
- **Consulter les services touristiques** : consulter les services touristiques disponibles et voir leurs détails et leur localisation
- **Suivre les statistiques** : suivre les statistiques concernant les touristes.
- **Inscription** : cette fonctionnalité est obligatoire pour avoir l'accès au contenu de l'application.

3.1.2 Besoins non fonctionnels

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Ces besoins peuvent concerner les contraintes d'implémentation (langage de programmation, type SGBD, de système d'Exploitation...).

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.

- **Authentification** : l'application devra être hautement sécurisée car les informations ne devront pas être accessibles à tout le monde.
- **L'ergonomie** : l'application doit offrir une interface conviviale, explicite et simple à utiliser.
- **Le contrôle des champs** : l'application doit avoir un contrôleur des champs de saisis, pour éviter l'introduction des informations qui ne correspondent pas aux types des champs.

3.2 Analyse des besoins

L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences du client. Il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution. [21]

3.2.1 Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externe (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. [1]

Notre application contient cinq acteurs principaux qui agissent directement avec le système :

- **Administrateur** : Le rôle de l'administrateur est la gestion des utilisateurs qui peuvent accéder au système.
- **Responsable du service tourisme** : Le responsable de service de tourisme est chargée de gérer les services touristiques cité ci-dessus et les publications, ainsi de suivre les statistique fourni par les responsable des hôtels.
- **Touriste** : Le touriste consulte les services touristiques dont il a besoin, leurs détails, ainsi que leur localisation.
- **Responsable de l'agence de voyage** :Le responsable de l'agence gère les publications concernant l'agence de voyage.
- **Responsable de l'association touristique** : Le responsable de l'association gère les publications concernant l'association touristique.
- **Responsable de l'hôtel** : Le responsable de l'hôtel est chargé de fournir au service de tourisme les statistiques des touristes ayant passés des nuitées au sein de son hôtel.

3.2.2 Modélisation du contexte

Le tableau suivant illustre les différents messages (informations, objets, services) échangés entre les acteurs de notre système et le système.

ID Messages Acteur – Système	ID Messages Système – Acteur
M1 Demande de l'authentification	M1' Interface Authentification
M2 Demande de l'inscription	M2' Interface Inscription
M3 Mise à jour des services touristiques	M3' Interface gestion ServiceTouristique
M4 Mise à jour des publications	M4' Interface gestion Publication
M5 Mise à jour des utilisateurs	M5' Interface gestion Utilisateur
M6 Mise à jour des statistiques	M6' Interface gestion Statistique
M7 Demande des statistiques	M7' Interface Statistique
M8 Demande des services touristiques	M8' Interface ServiceTouristique

TABLE 3.1 – Message échangés entre les acteurs et le système

Tous les messages (Système–Acteurs) identifiés précédemment peuvent être représentés de façon synthétique sur un diagramme appelé diagramme de contexte.

Le diagramme suivant représente les différents acteurs du système, ainsi que les messages échangés entre ces derniers.

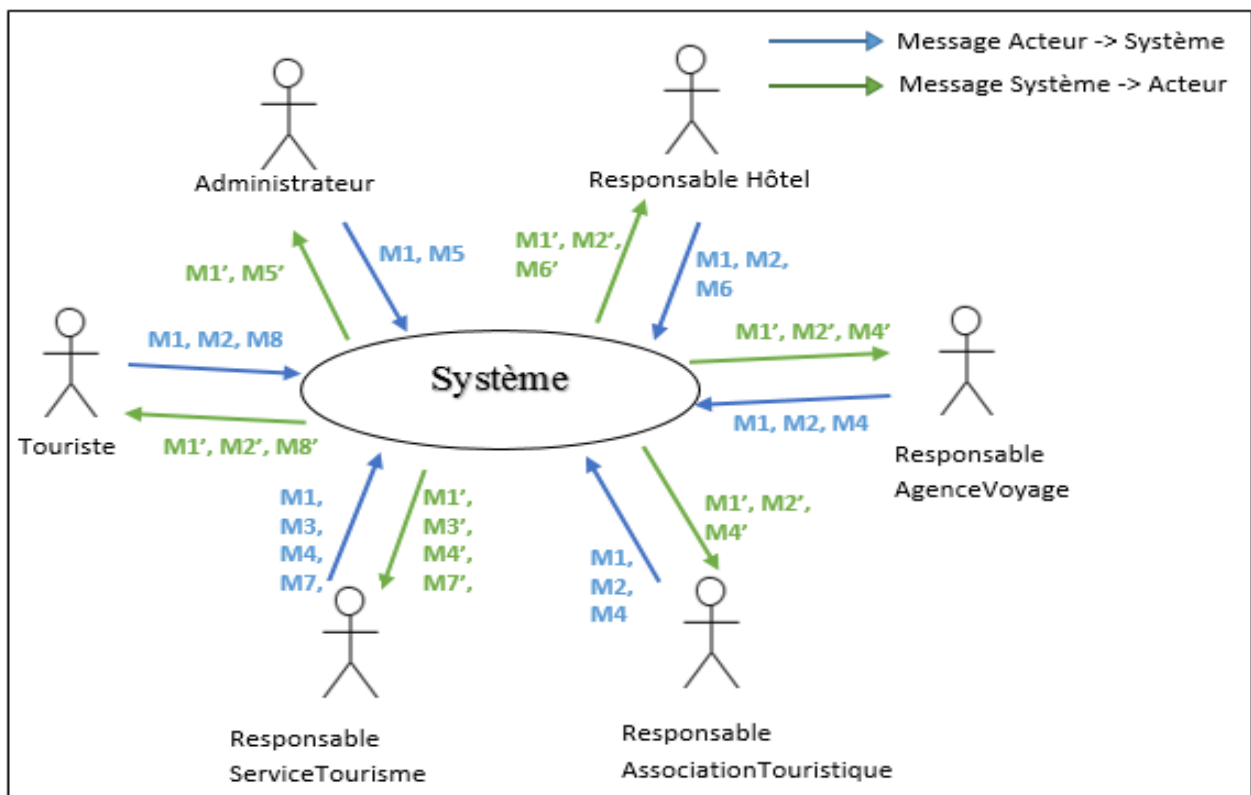


FIGURE 3.1 – Diagramme de contexte dynamique

3.2.3 Identification des cas d'utilisation

Un cas d'utilisation (use case) représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier.

Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée "notable" à l'acteur concerné. [1]

Pour chaque acteur identifié précédemment on associe les cas d'utilisation qui lui correspondent.

Le tableau suivant liste les différents cas d'utilisation associés à notre système.

N°	Cas d'utilisation	Acteur
1	S'inscrire	Responsable AssociationTouristique
		Responsable AgenceVoyage
		Responsable Hôtel
		Touriste
2	Gérer un Utilisateur	Administrateur
3	Gérer un ServiceTouristique	Responsable ServiceTourisme
4	Suivre une Statistique	
5	Gérer une Publication	Responsable ServiceTourisme
		Responsable AssociationTouristique
		Responsable AgenceVoyage
6	Gérer une Statistique	Responsable Hôtel
7	Consulter un ServiceTouristique	Touriste

TABLE 3.2 – Cas d'utilisation associés au système

3.2.4 Diagrammes des cas d'utilisation et leurs descriptions

— Diagramme de cas d'utilisation global

Le diagramme suivant résume tous les cas d'utilisation associés à tous les acteurs de notre système.

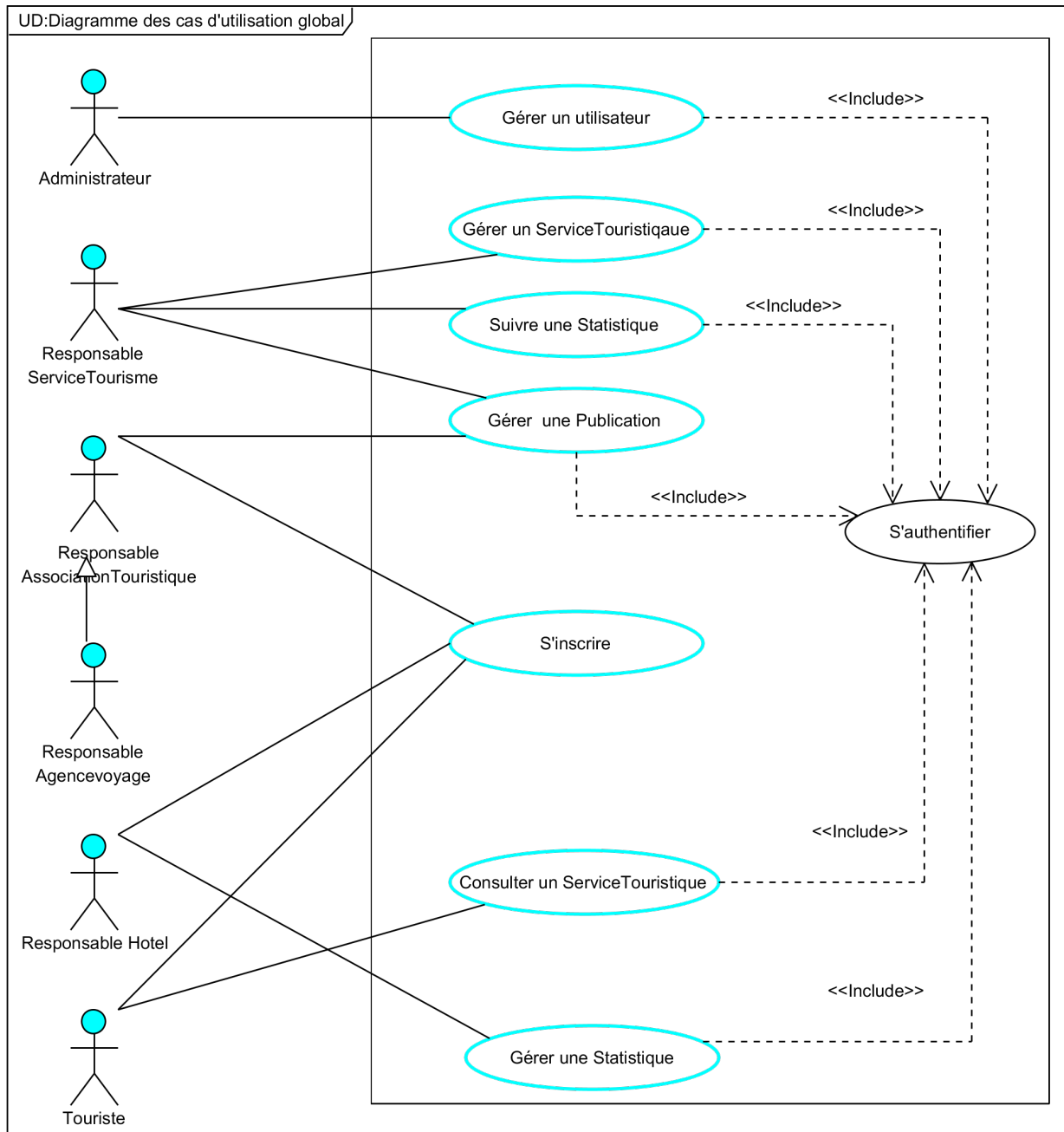


FIGURE 3.2 – Diagramme de cas d'utilisation global

— Diagrammes de cas d'utilisation Détaillés

Diagrammes de cas d'utilisation Détaillés

- Côté " Administrateur "

Le diagramme ci-dessous représente le diagramme des cas d'utilisation associés à "l'Administrateur".

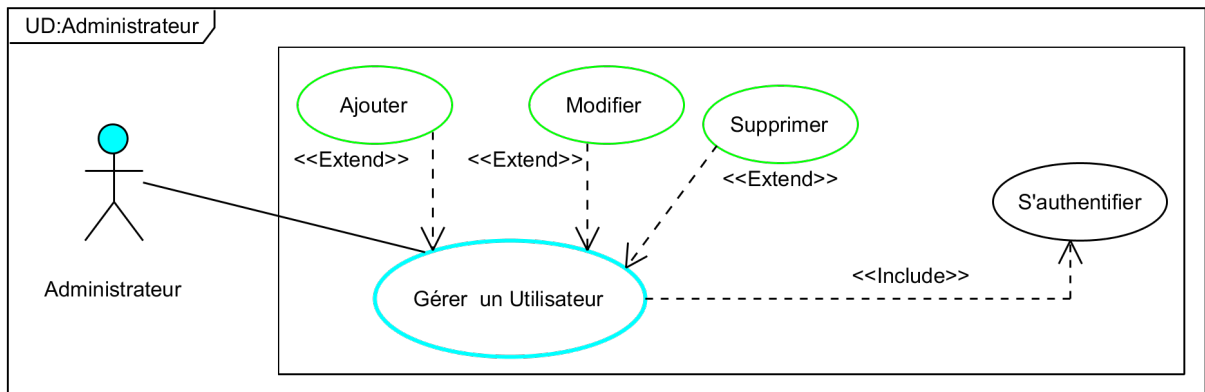


FIGURE 3.3 – Diagramme de cas d'utilisation associé à " Administrateur "

- Description de cas d'utilisation " Gérer un Utilisateur "

Cas d'utilisation	Gérer un Utilisateur
Acteur	Administrateur
Objectif	Mise à jour des utilisateurs de l'application
Pré-condition	Authentification
Scénario nominal	<ul style="list-style-type: none"> - Le système affiche l'interface gestion Utilisateur - L'administrateur effectue la mise à jour et valide [A1] - Le système enregistre les données
Alternative [A1]	Si un champ d'information n'est pas valide ou l'utilisateur existe déjà, le système affiche un message d'erreur et réaffiche l'interface gestion Utilisateur

TABLE 3.3 – Description de cas d'utilisation " Gérer un Utilisateur "

Cette description a une représentation avec un diagramme de séquence système. voir l'annexe à la page (II).

- Côté " Responsable ServiceTourisme "

Le diagramme ci-dessous représente le diagramme des cas d'utilisation associés au responsable service tourisme.

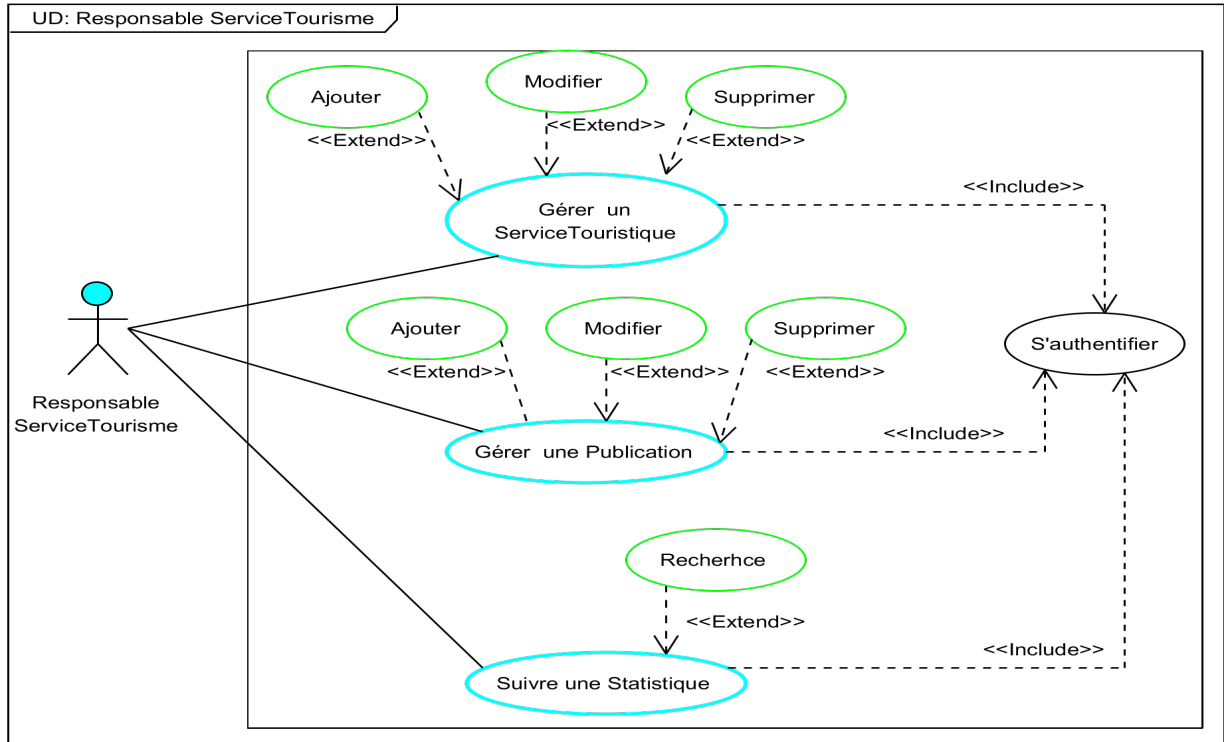


FIGURE 3.4 – Diagramme de cas d'utilisation associé au " Responsable ServiceTourisme "

- Description de cas d'utilisation "Gérer un ServiceTouristique"

Cas d'utilisation	Gérer un ServiceTouristique
Acteur	Responsable ServiceTourisme
Objectif	Mise à jour des services touristiques
Pré-condition	Authentification
Scénario nominal	<ul style="list-style-type: none"> - Le système affiche l'interface gestion ServiceTouristique - L'utilisateur effectue la mise à jour et valide [A1] - Le système enregistre les données
Alternative [A1]	Si un champ d'information n'est pas valide ou le service touristique existe déjà, le système affiche un message d'erreur et réaffiche l'interface gestion ServiceTouristique

TABLE 3.4 – Description de cas d'utilisation " Gérer un ServiceTouristique"

la description ci-dessus a une représentation avec un diagramme de séquence système. voir l'annexe à la page (III).

- Description de cas d'utilisation " Suivre une Statistique "

Cas d'utilisation	Suivre une Statistique
Acteur	Responsable ServiceTourisme
Objectif	suivre et calculer les statistiques
Pré-condition	Authentification
Scénario nominal	<ul style="list-style-type: none"> - Le système affiche l'interface Statistique - L'utilisateur demande une statistique - Le système renvoie le résultat

TABLE 3.5 – Description de cas d'utilisation " suivre une statistique "

la description précédente est présentée sous forme d'un diagramme de séquence système. voir l'annexe à la page (IV)

- Côté " Responsable AgenceVoyage "

Le diagramme ci-dessous représente le diagramme des cas d'utilisation associés au "responsable AgenceVoyage".

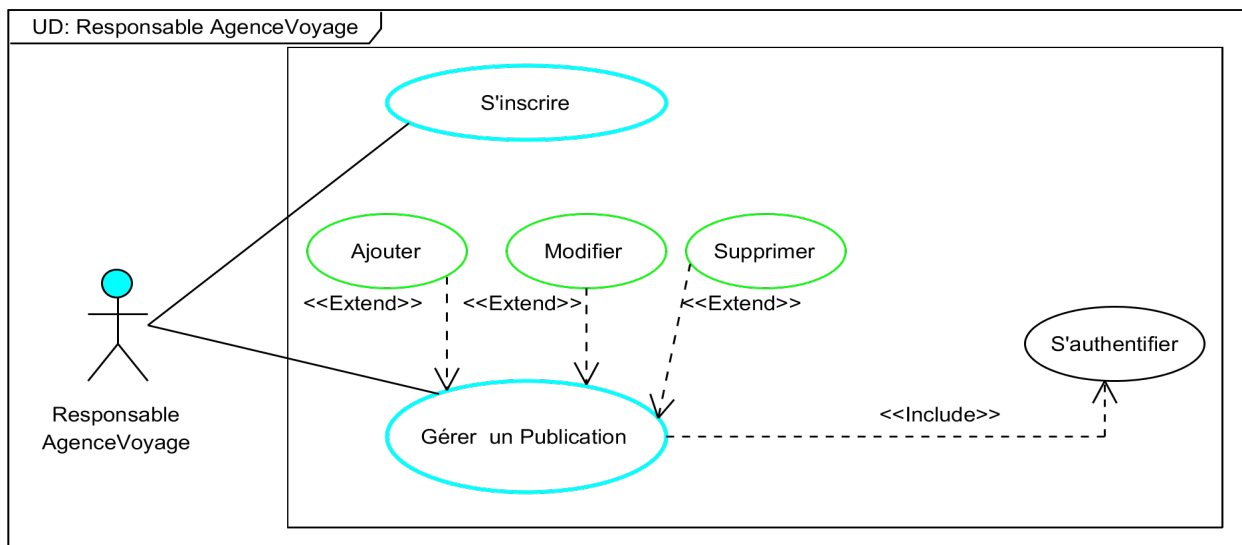


FIGURE 3.5 – Diagramme de cas d'utilisation associé au " Responsable AgenceVoyage "

- Côté " Responsable AssociationTouristique "

Le diagramme ci-dessous représente le diagramme des cas d'utilisation associés au "responsableAAssociationTouristique".

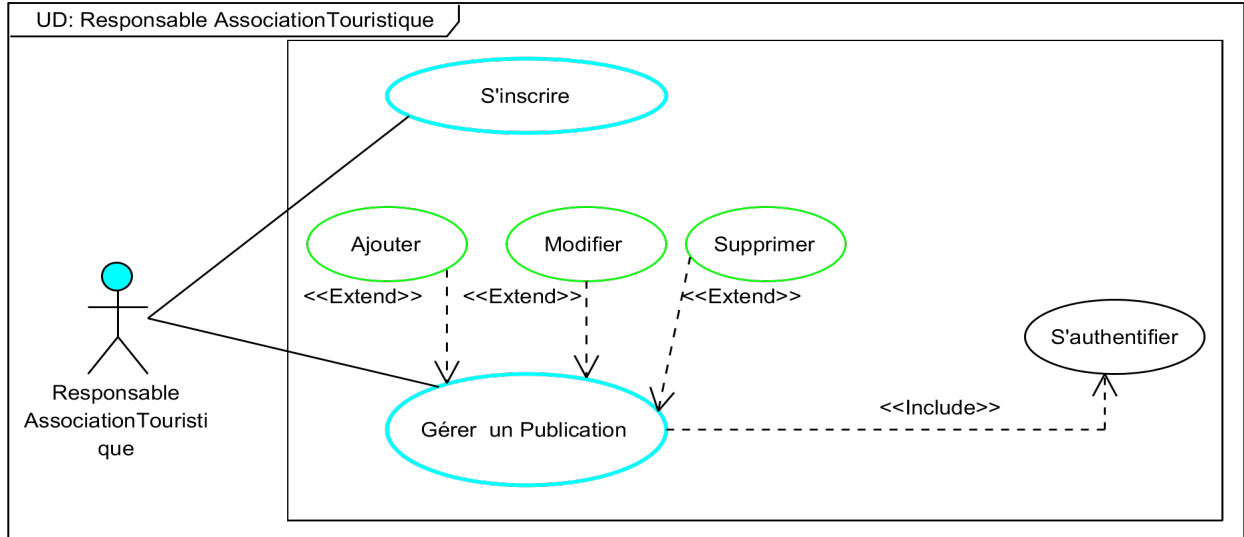


FIGURE 3.6 – Diagramme de cas d'utilisation associé au " Responsable AssociationTouristique "

- Côté " Responsable Hôtel "

Le diagramme ci-dessous représente le diagramme des cas d'utilisation associés au responsable hôtel.

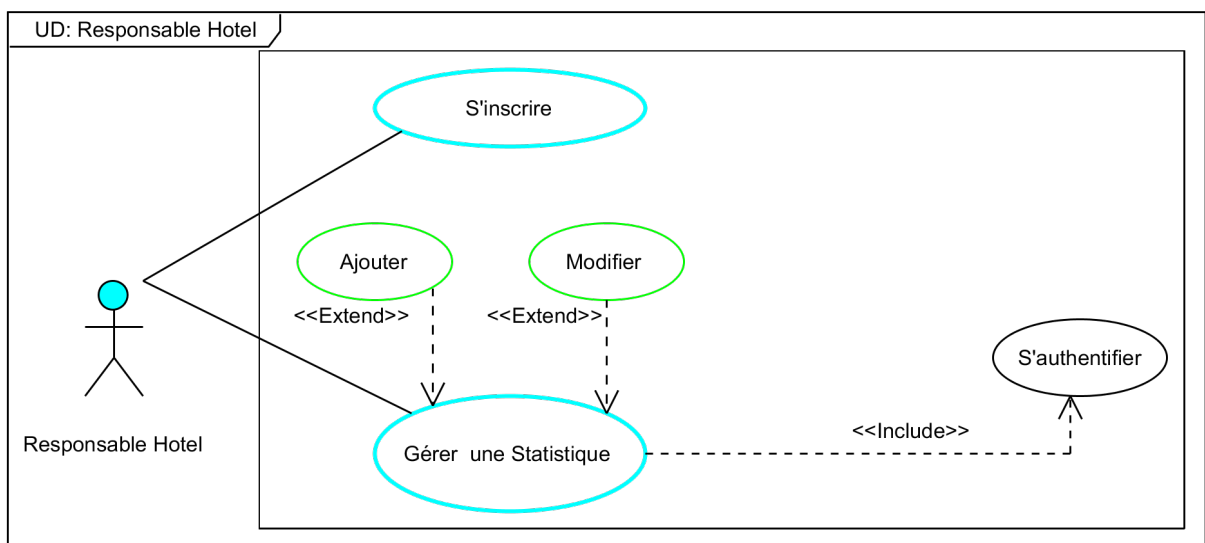


FIGURE 3.7 – Diagramme de cas d'utilisation associé au " Responsable Hôtel "

- Description de cas d'utilisation " Gérer une Statistique "

Cas d'utilisation	Gérer une statistique
Acteur	Responsable Hôtel
Objectif	Mise à jour des statistiques
Pré-condition	Authentification
Scénario nominal	- Le système affiche l'interface gestion Statistique - L'utilisateur effectue la mise à jour et valide [A1] - Le système enregistre les données
Alternative [A1]	Si un champ d'information n'est pas valide ou la statistique existe déjà, le système affiche un message d'erreur et réaffiche l'interface statistique

TABLE 3.6 – Description de cas d'utilisation " Gérer une Statistique "

Cette description contient un diagramme de séquence système associé à elle. voir l'annexe à la page (V) .

- Côté " Touriste "

Le diagramme ci-dessous représente le diagramme des cas d'utilisation associés au "Touriste".

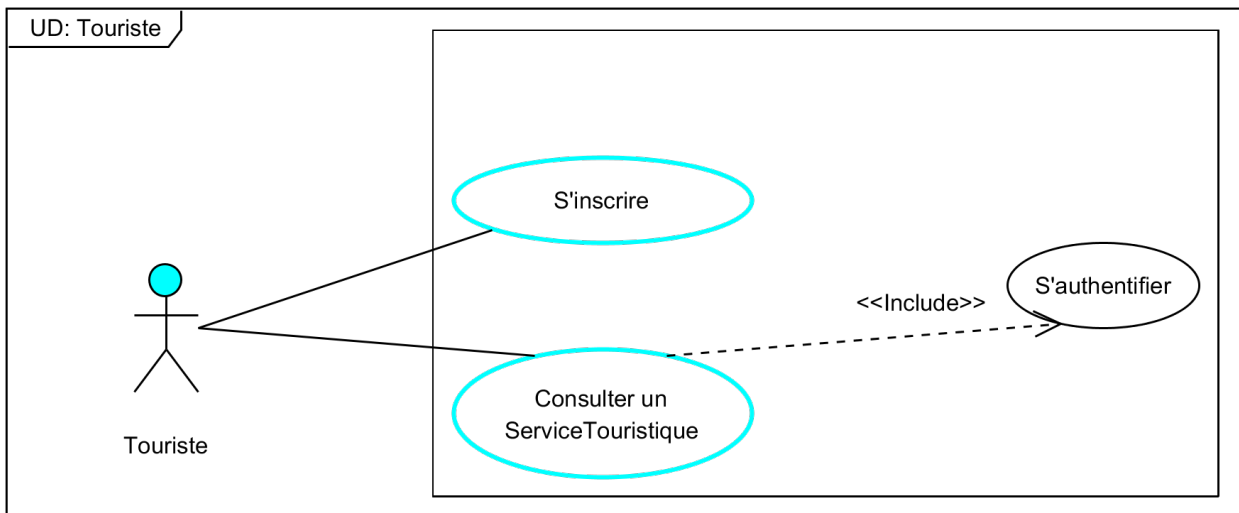


FIGURE 3.8 – Diagramme de cas d'utilisation associé au " Touriste "

SPÉCIFICATION ET ANALYSE DES BESOINS

- Description de cas d'utilisation " Consulter un ServiceTouristique "

Cas d'utilisation	Consulter un ServiceTouristique
Acteur	Tourist
Objectif	Consultation des services touristiques
Pré-condition	Authentification
Scénario nominal	<ul style="list-style-type: none"> - Le système affiche l'interface SeviceTouristique - L'utilisateur demande les détails d'un service touristique - Le système affiche les détails

TABLE 3.7 – Description de cas d'utilisation " Consulter un ServiceTouristique "

la description ci-dessus a une représentation avec un diagramme de séquence système. voir l'annexe à la page (VI).

— Descriptions communes

- Description de cas d'utilisation " s'inscrire "

Cette description est la même pour (Responsable AssociationTouristique, Responsable AgenceVoyage, Responsable Hôtel et le Touriste)

Cas d'utilisation	s'inscrire
Acteur	<ul style="list-style-type: none"> -Responsable AssociationTouristique -Responsable AgenceVoyage -Responsable Hôtel -Touriste
Objectif	Créer un compte avec lequel l'utilisateur accède aux fonctionnalités dont il a le droit
Scénario nominal	<ul style="list-style-type: none"> - L'utilisateur demande l'inscription - Le système affiche le formulaire d'inscription - L'utilisateur remplir le formulaire et valider - Le système vérifie la validité des informations [A1] - Le système crée un compte pour l'utilisateur
Alternative [A1]	Si un champ d'information n'est pas valide ou l'utilisateur existe déjà, le système affiche un message d'erreur et réaffiche l'interface statistique

TABLE 3.8 – Description de cas d'utilisation " s'inscrire "

Cette description est représentée sous forme d'un diagramme de séquence système. voir l'annexe à la page (VII).

- Description de cas d'utilisation " s'authentifier "

La description ci-dessous concerne les acteurs (Administrateur, Responsable ServiceTourisme, Responsable AssociationTouristique, Responsable AgenceVoyage, Responsable Hôtel et le Touriste).

Cas d'utilisation	Authentification
Acteur	<ul style="list-style-type: none"> -Administrateur -Responsable ServiceTourisme -Responsable AssociationTouristique -Responsable AgenceVoyage -Responsable Hôtel -Touriste
Objectif	Vérifier que l'utilisateur a bien le droit d'accès à l'application
Scénario nominal	<ul style="list-style-type: none"> - L'utilisateur demande l'accès à l'application - Le système affiche l'interface Authentification - L'utilisateur introduit son identifiant et le mot de passe - Le système vérifie l'existence de l'utilisateur. [A1] - Le système donne l'accès à l'interface correspondante
Alternative [A1]	Si un champ d'information n'est pas valide ou l'utilisateur n'existe pas, le système affiche un message d'erreur et réaffiche l'interface statistique

TABLE 3.9 – Description de cas d'utilisation " s'authentifier "

Cette description est schématisée avec un diagramme de séquence système. voir l'annexe à la page (VIII).

- Description de cas d'utilisation " Gérer une Publication "

La description suivante est la même pour les acteurs (Responsable ServiceTourisme, Responsable AssociationTouristique et Responsable AgenceVoyage).

Cas d'utilisation	Gérer une Publication
Acteur	-Responsable ServiceTourisme -Responsable AgenceVoyage -Responsable AssociationTouristique
Objectif	Mise à jour des publications
Pré-condition	Authentification
Scénario nominal	- Le système affiche l'interface gestion Publication - L'utilisateur effectue la mise à jour et valide [A1] - Le système enregistre les données
Alternative [A1]	Si un champ d'information n'est pas valide ou la publication existe déjà, le système affiche un message d'erreur et réaffiche l'interface gestion Publication

TABLE 3.10 – Description de cas d'utilisation " Gérer une publication "

Cette description a une représentation sous forme d'un diagramme de séquence système. voir l'annexe à la page (IX).

3.3 Conclusion

A l'issue de cette étape, nous avons pu exprimer et décrire les objectifs et les besoins de notre système, ainsi de les exprimer sous forme de diagramme de cas d'utilisation et diagramme de séquence.

Dans le chapitre suivant, nous entamons la deuxième phase dans laquelle nous décrirons d'une manière détaillée ces besoins.

4

CONCEPTION

Introduction

La démarche de conception est une étape fondamentale dans le processus de développement puisqu'elle fait correspondre la vision applicative (le modèle d'analyse) à la vision technique (l'environnement de développement et d'exécution). [3]

Ce chapitre vise à illustrer la phase de conception et les modèles UML associés.

Nous commençons par établir les diagrammes de séquences des cas d'utilisation, ensuite nous élaborons le diagramme de classe, suivi du passage de ce dernier en Modèle Logique de Données (MLD) et nous terminons avec une petite conclusion.

4.1 Diagrammes de séquences des cas d'utilisation

4.1.1 Diagramme de séquence du cas d'utilisation "s'authentifier"

L'utilisateur doit s'authentifier en saisissant son identifiant et mot de passe, puis le système vérifie les informations introduite, si l'une des coordonnées est non valide le système affiche un message d'erreur sinon il envoie une requête de vérification au serveur qui vérifie l'existence de l'utilisateur dans le SGBD, en cas d'erreur le système affiche un message d'erreur, sinon ce dernier affiche une interface correspondant au rôle de l'utilisateur.

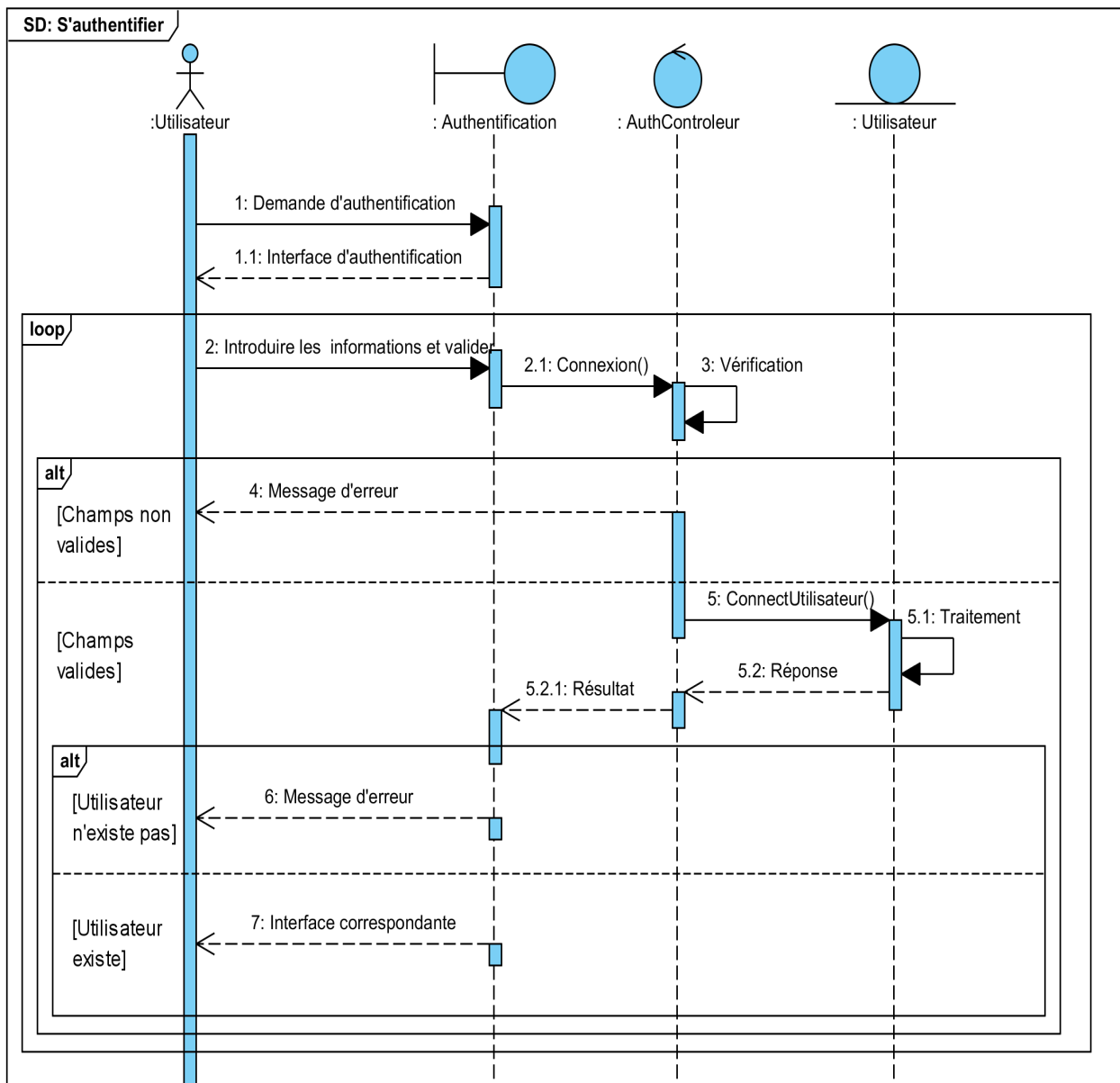


FIGURE 4.1 – Diagramme de séquence du cas d'utilisation " s'authentifier "

4.1.2 Diagramme de séquence du cas d'utilisation "Gérer Service-Touristique"

Pour effectuer des mise à jour de la liste des services touristiques, le responsable de ServiceTourisme doit s'authentifier, ensuite ajouter un service touristique en remplissant un formulaire d'ajout ou modifier, supprimer un service sélectionné dans la base de données.

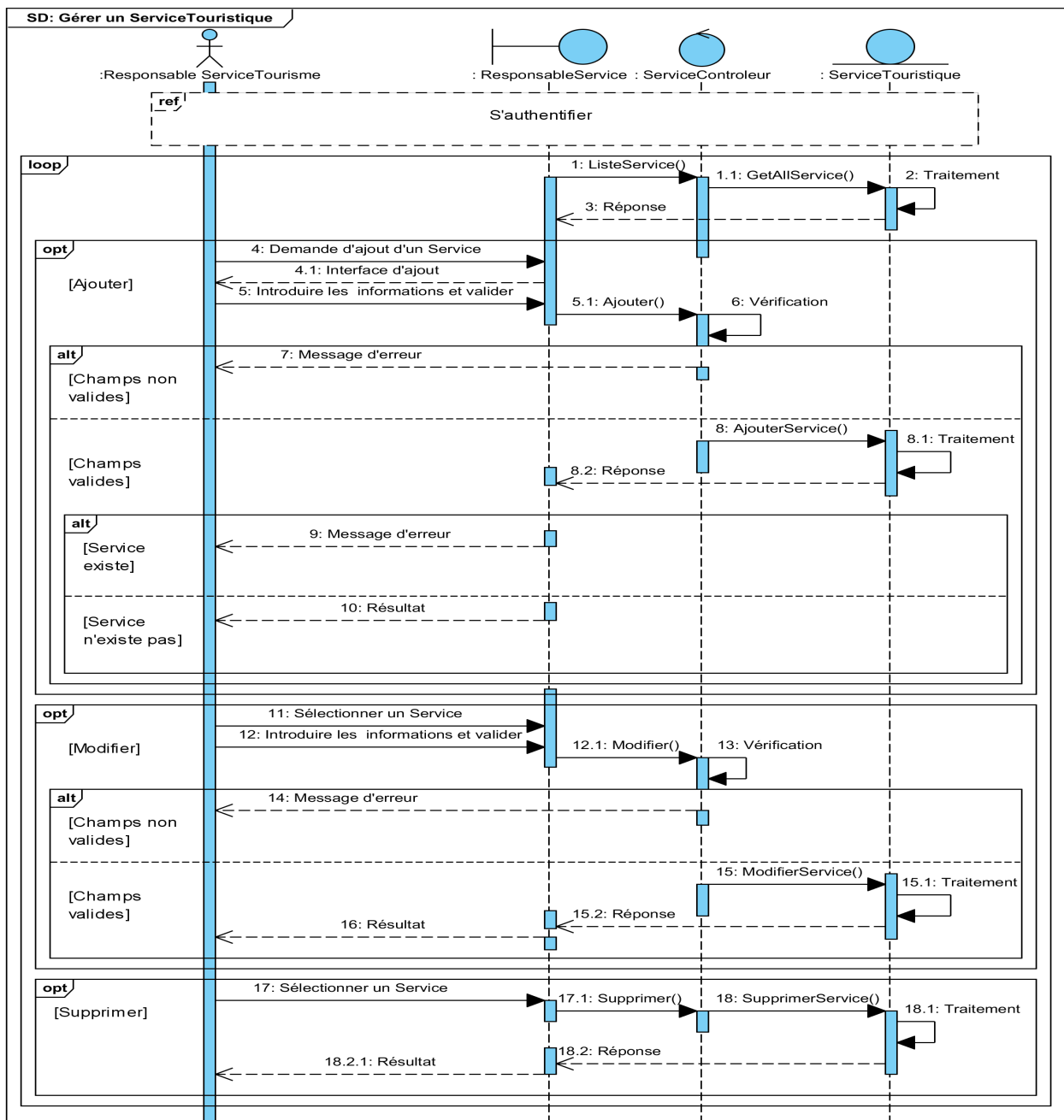


FIGURE 4.2 – Diagramme de séquence du cas d'utilisation " Gérer un ServiceTouristique "

4.1.3 Diagramme de séquence du cas d'utilisation "Gérer un Utilisateur"

Pour que l'Administrateur puisse mettre à jour la liste des utilisateurs, il doit d'abord s'authentifier, ensuite ajouter un utilisateur en remplissant un formulaire d'ajout ou modifier, supprimer un utilisateur sélectionné dans la base de données.

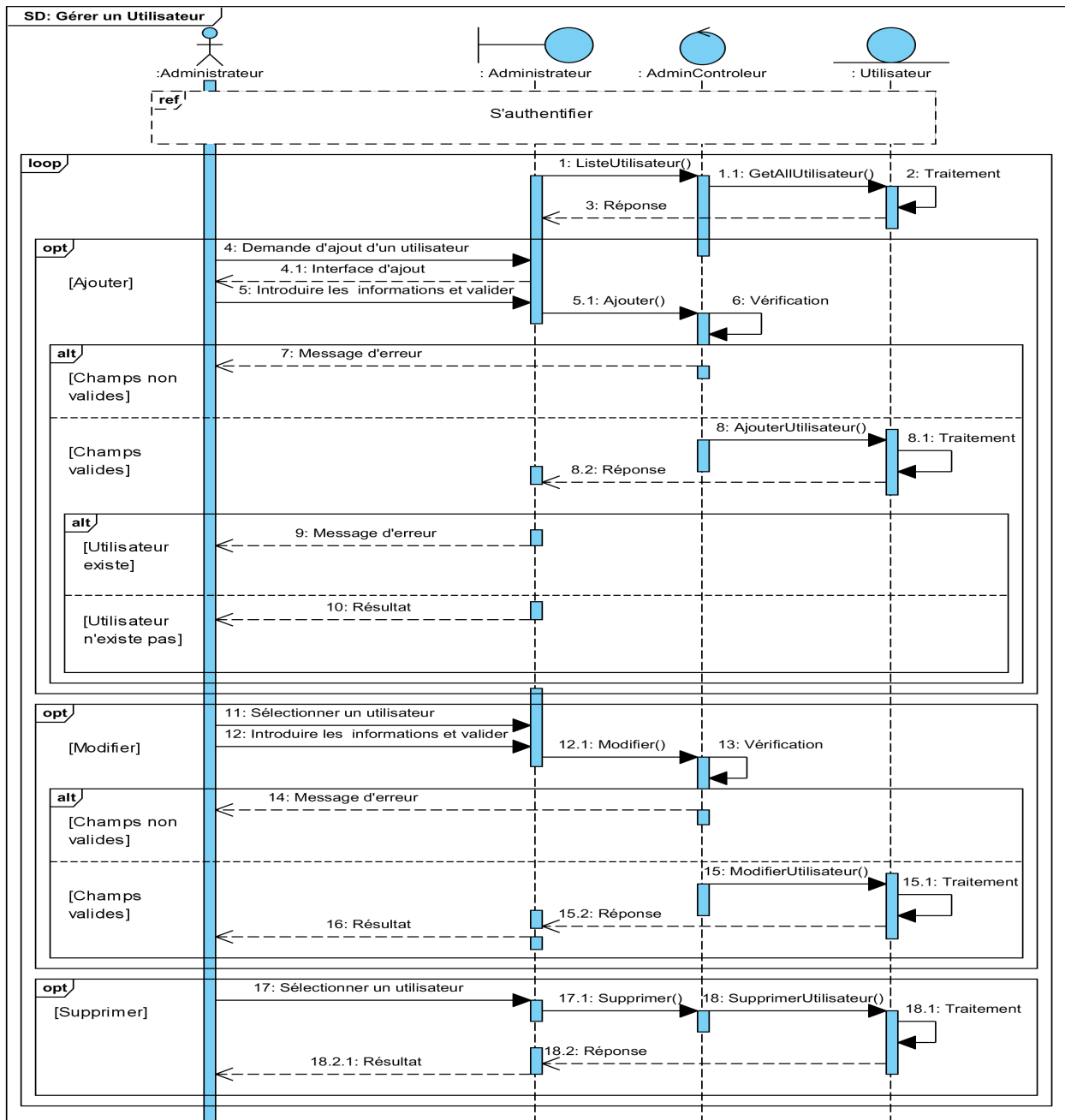


FIGURE 4.3 – Diagramme de séquence du cas d'utilisation "Gérer un Utilisateur"

4.1.4 Diagramme de séquence du cas d'utilisation "Gérer une Publication"

Pour mettre à jour la liste des publications, Le responsable de ServiceTourisme, responsable Agencevoyage et responsable AssociationTouristique doivent s'authentifier, ensuite ajouter une publication en remplissant un formulaire d'ajout ou modifier, supprimer une publication sélectionné dans la base de données.

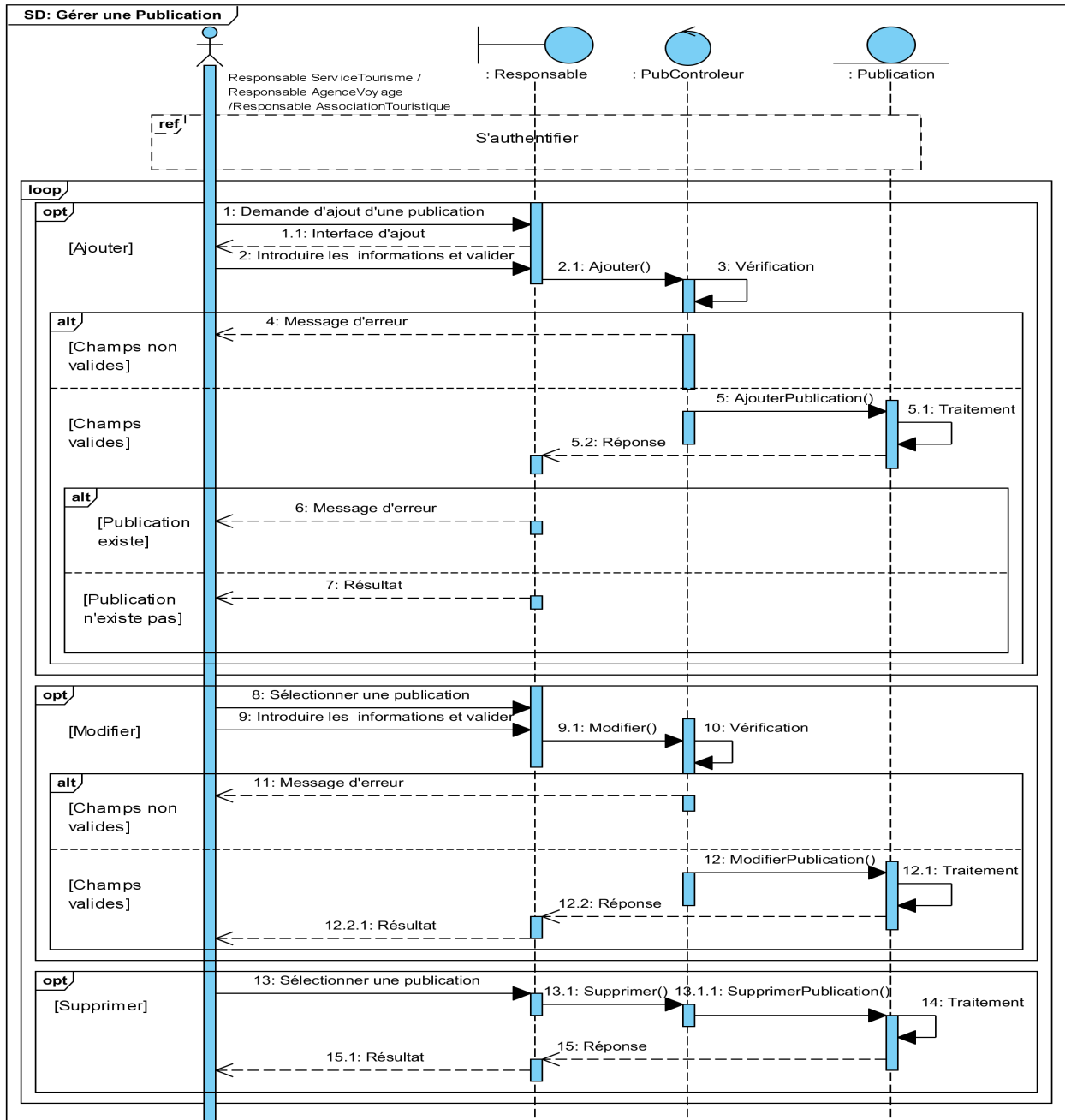


FIGURE 4.4 – Diagramme de séquence du cas d'utilisation "Gérer une Publication"

4.1.5 Diagramme de séquence du cas d'utilisation "Gérer une Statistique"

Pour mettre à jour la liste des statistiques, le responsable d'hôtel doit s'authentifier, ensuite ajouter une statistique en remplissant un formulaire d'ajout ou modifier, une statistique sélectionnée dans la base de données.

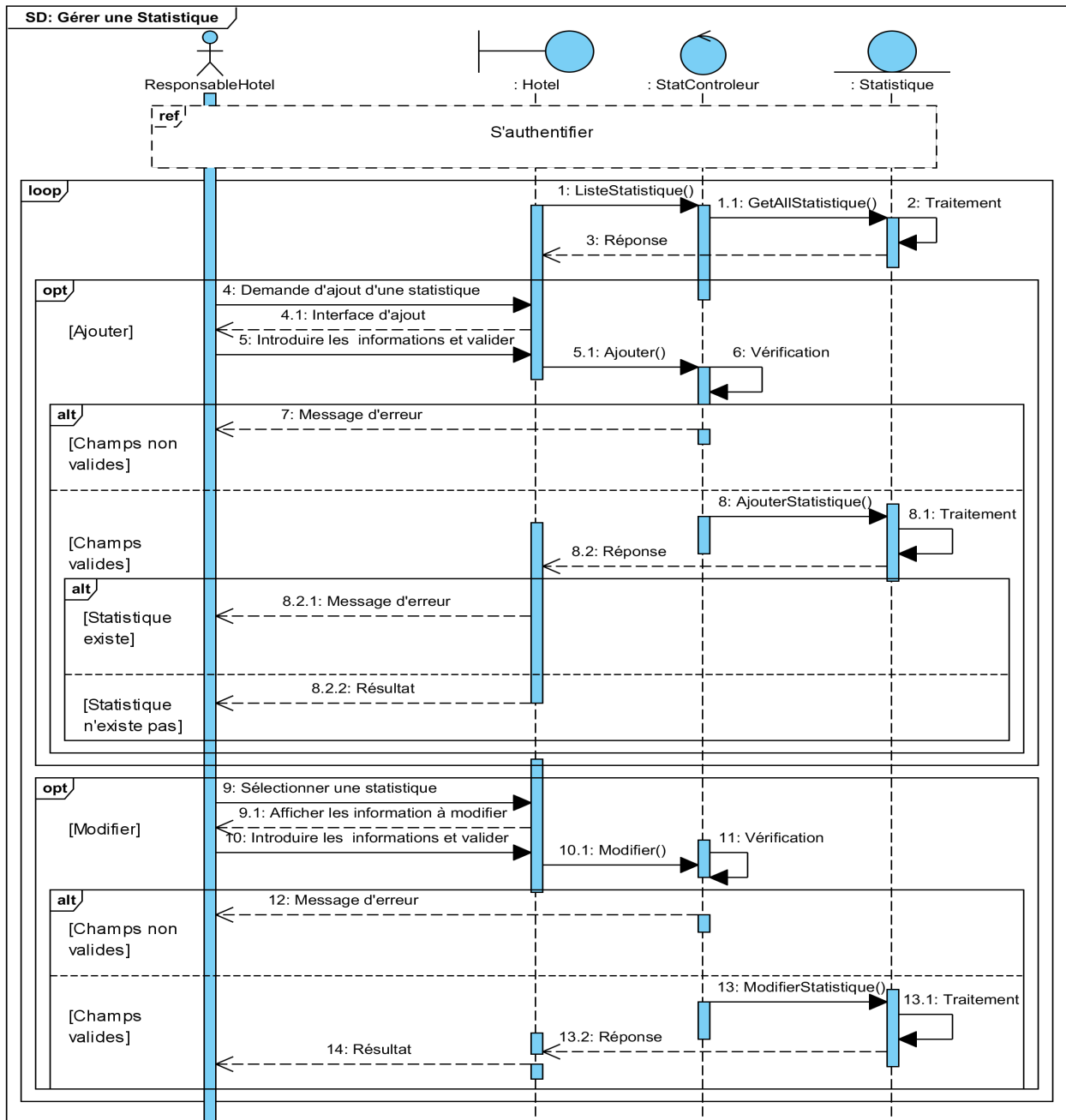


FIGURE 4.5 – Diagramme de séquence du cas d'utilisation "Gérer une Statistique"

4.1.6 Diagramme de séquence du cas d'utilisation "Suivre une Statistique"

Dans ce cas d'utilisation, le responsable de ServiceTourisme doit d'abord s'authentifier, ce dernier peut consulter et suivre les statistiques des hôtels en remplissant un formulaire de recherche.

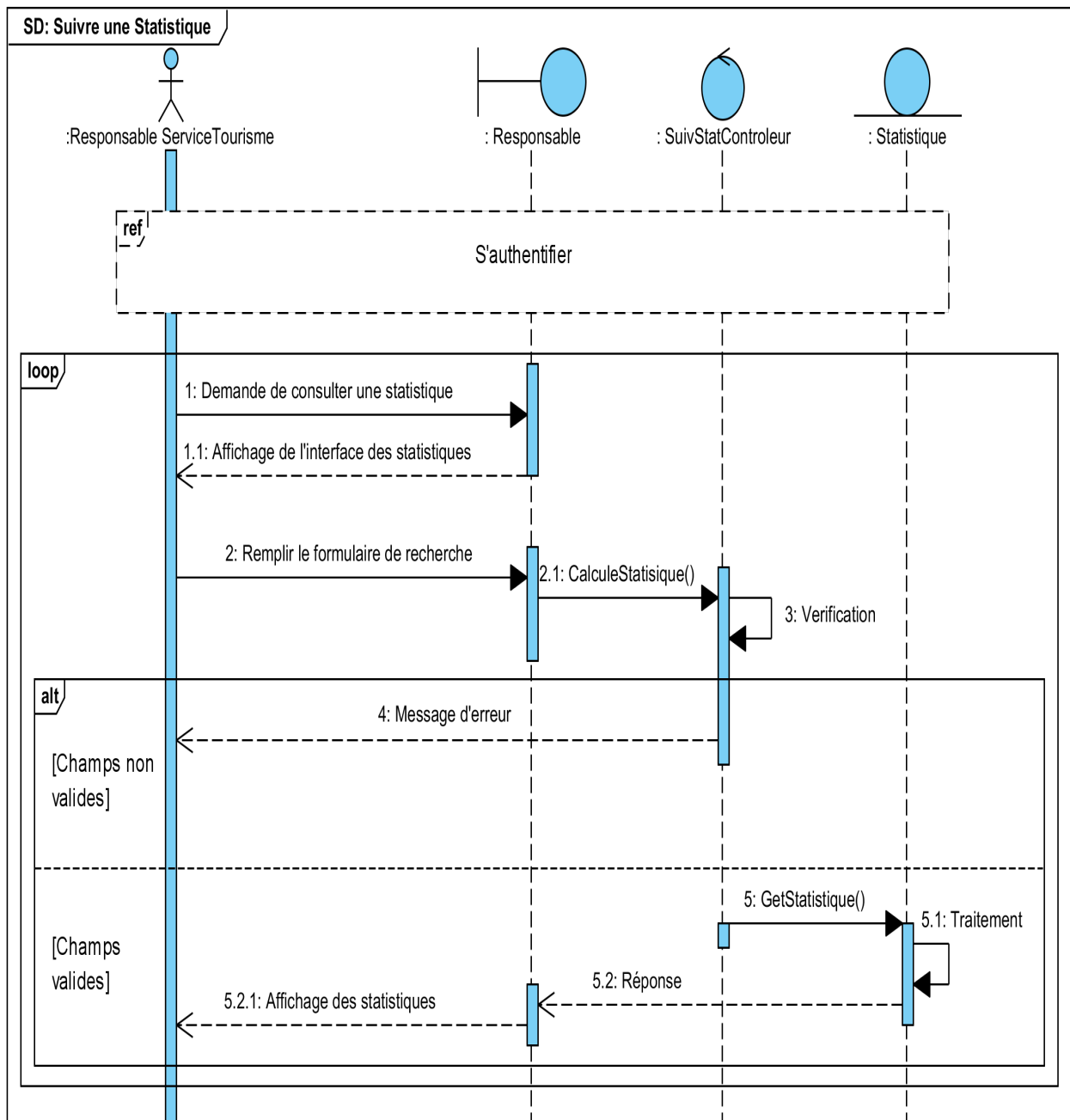


FIGURE 4.6 – Diagramme de séquence du cas d'utilisation "Suivre une Statistique"

4.1.7 Diagramme de séquence du cas d'utilisation "Consulter un ServiceTouristique"

Dans ce cas d'utilisation, le touriste doit d'abord s'authentifier, pour pouvoir consulter les services touristiques disponible.

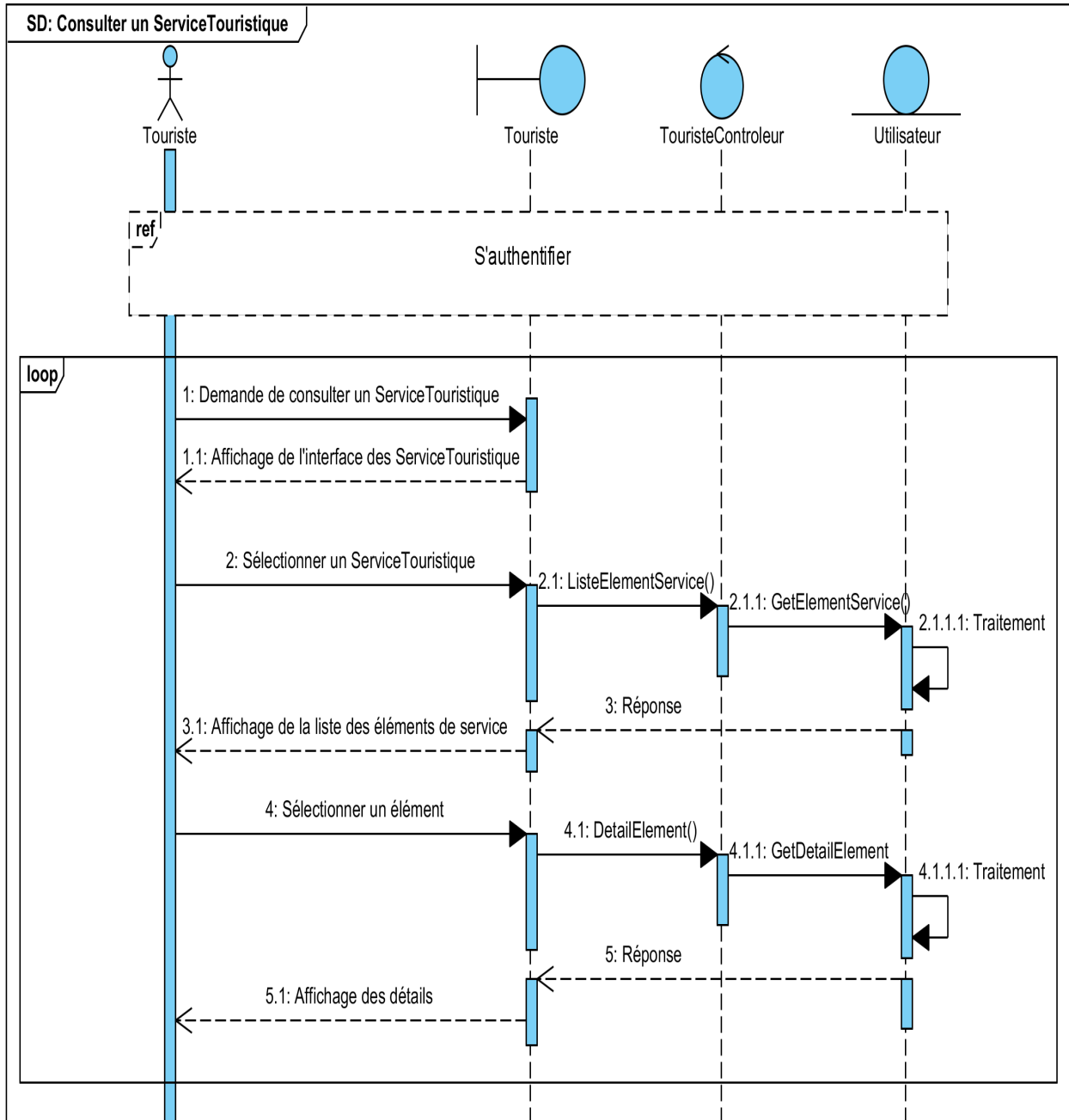


FIGURE 4.7 – Diagramme de séquence du cas d'utilisation "Consulter une ServiceTouristique"

4.1.8 Diagramme de séquence du cas d'utilisation "s'inscrire"

Dans ce cas d'utilisation, le nouveau utilisateur doit remplir le formulaire d'inscription pour pouvoir utiliser l'application.

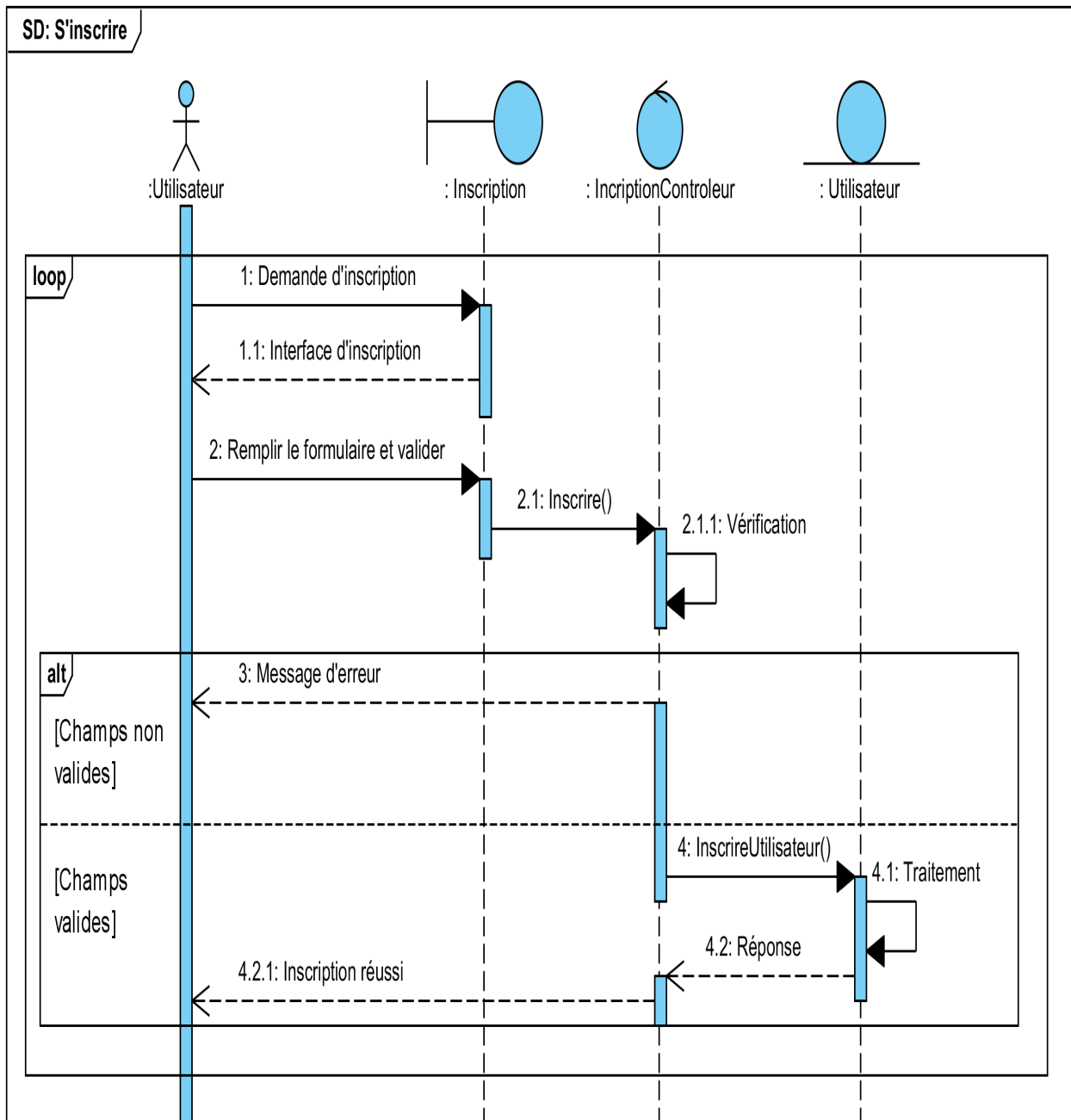


FIGURE 4.8 – Diagramme de séquence du cas d'utilisation "s'inscrire"

4.2 Patrons de conception (Design Pattern)

En informatique, et plus particulièrement en développement logiciel, un patron de conception (design pattern) est un arrangement caractéristique de modules, reconnu comme bonne pratique en réponse à un problème de conception d'un logiciel. Il décrit une solution standard, utilisable dans la conception de différents logiciels. [22]

Les patrons utilisés dans notre projet sont les suivants :

4.2.1 Patron client/serveur à 3-tiers

Notre système est constitué de deux applications conçues selon l'architecture Client/serveur à trois niveaux.

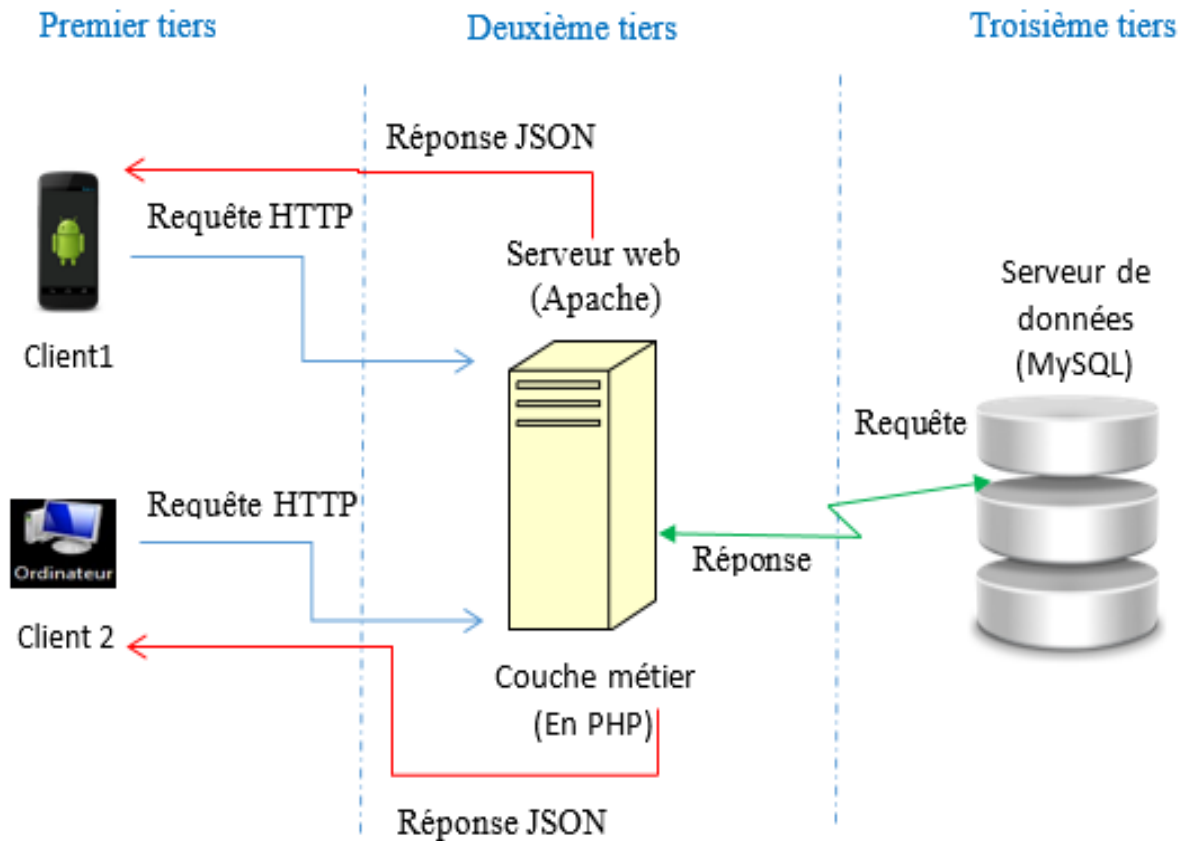


FIGURE 4.9 – Architecture 3- tiers

4.2.2 Patron Modèle-vue-contrôleur

Le patron d'architecture logicielle MVC, est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leurs architectures respectives.

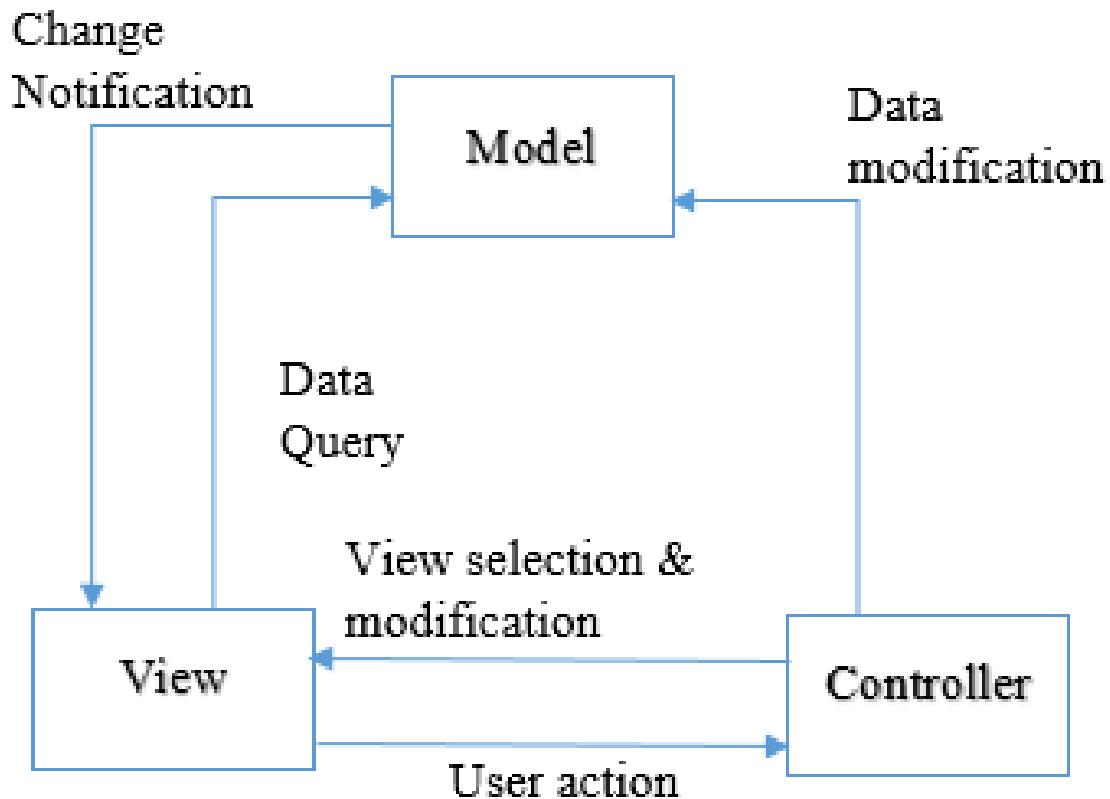


FIGURE 4.10 – Patron MVC (Modèle, Vue, Contrôleur)

Ce modèle de conception impose donc une séparation en 3 couches :

- **Le modèle :** Il représente les données de l'application .Il définit aussi l'interaction avec la base de données et le traitement de ces données.
- **La vue :** Elle représente l'interface utilisateur, ce avec quoi il interagit. Elle n'effectue aucun traitement, elle se contente simplement d'afficher les données que lui fournit le modèle. Il peut tout à fait y avoir plusieurs vues qui présentent les données d'un même modèle.
- **Le contrôleur :** Il gère l'interface entre le modèle et le client. Il va interpréter la requête de ce dernier pour lui envoyer la vue correspondante.

4.3 Diagramme de classe de conception

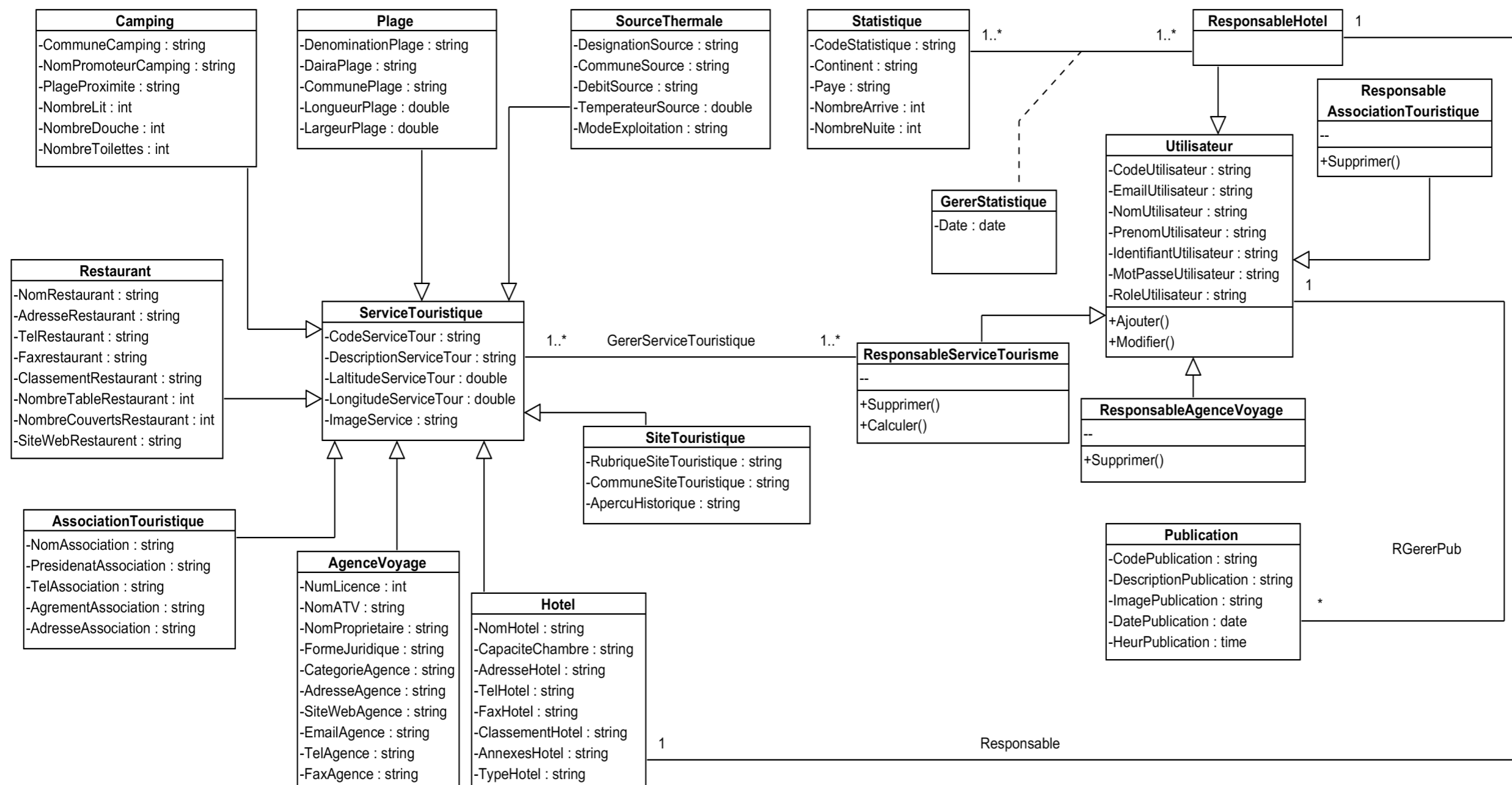


FIGURE 4.11 – Diagramme de classe de conception

4.4 Passage au modèle relationnel

Le modèle relationnel est le modèle logique de donnée qui correspond à l'organisation des données dans les bases de données relationnelles. Un modèle relationnel est composé de relations, encore appelée table. Ces tables sont décrites par des attributs aux champs. Pour décrire une relation, on indique tout simplement son nom, suivi du nom de ses attributs entre parenthèses. L'identifiant d'une relation est composé d'un ou plusieurs attributs qui forment la clé primaire. Une relation peut faire référence à une autre en utilisant une clé étrangère, qui correspond à la clé primaire de la relation référencée. [1]

4.4.1 Règles de passage au modèle relationnel

Cette section présente les règles permettant de décrire un schéma logique dans les modèles relationnel et objet-relationnel à partir d'un diagramme de classe UML.

Nous donnons ci-après quatre règles (de R1 à R4) pour traduire un schéma UML en un schéma relationnel équivalent. Il existe d'autres solutions de transformation, mais ces règles sont les plus simples et les plus opérationnelles. [23]

Transformation des classes (R1)

Chaque classe du diagramme UML devient une relation. Il faut choisir un attribut de la classe pouvant jouer le rôle d'identifiant. Si aucun attribut ne convient en tant qu'identifiant, il faut en ajouter un de telle sorte que la relation dispose d'une clé primaire (les outils proposent l'ajout de tels attributs). [23]

Transformation des associations

Les règles de transformation que nous allons voir dépendent des cardinalités/multiplicités maximales des associations. Nous distinguons trois familles d'associations.

- **Association un à plusieurs (R2)** : Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association. La clé de la relation père migre dans la relation fils. [23]
- **Association plusieurs à plusieurs (R3)** : L'association (classe-association) devient une relation dont la clé primaire est composée par la concaténation des identifiants des classes connectés à l'association. Chaque attribut devient clé étrangère si classe connectée dont il provient devient une relation en vertu de la règle R1.

Les attributs de l'association (classe-association) doivent être ajoutés à la nouvelle relation. Ces attributs ne sont ni clé primaire, ni clé étrangère. [23]

- **Association un à un (R4)** : Il faut ajouter un attribut clé étrangère dans la relation dérivée de l'entité ayant la cardinalité minimale égale à zéro. Dans le cas de UML, il faut ajouter un attribut clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de l'entité (classe) connectée à l'association.

Si les deux cardinalités (multiplicités) minimales sont à zéro, le choix est donné entre les deux relations dérivées de la règle R1. Si les deux cardinalités minimales sont à un, il est sans doute préférable de fusionner les deux entités (classes) en une seule. [23]

Transformation de l'héritage

Trois décompositions sont possibles pour traduire une association d'héritage en fonction des contraintes existantes dont la décomposition par distinction, décomposition descendante, décomposition. [23]

Dans notre projet nous allons utiliser que le deuxième cas.

- **décomposition descendante (push-down)** : s'il existe une contrainte de totalité ou de partition sur l'association, il est possible de ne pas traduire les relations issues de la sur-classe. Il faut alors faire migrer tous ses attributs dans les relations issues des sous classes. [23]

4.4.2 Modèle logique des données

- **Restaurant** (CodeServiceTour, NomRestaurant, AdresseRestaurant, TelRestaurant, FaxRestaurant, ClassementRestaurant, NombreTableRestaurant, NombreCouvertsRestaurant, SiteWebRestaurent, DescriptionServiceTour, LaltitudeServiceTour, LongitudeServiceTour, ImageService)
- **Hotel** (CodeServiceTour, NomHotel, CapaciteChambre, AdresseHotel, TelHotel, FaxHotel, ClassementHotel, AnnexesHotel, TypeHotel, DescriptionServiceTour, LaltitudeServiceTour, LongitudeServiceTour, ImageService)
- **SiteTouristique** (CodeServiceTour, RubriqueSiteTouristique, CommuneSiteTouristique, ApercuHistorique, DescriptionServiceTour, LaltitudeServiceTour, LongitudeServiceTour, ImageService)
- **Camping** (CodeServiceTour, CommuneComping, NomPromoteurComping, PlageProximite, NombreLit, NombreDouche, NombreToilettes, DescriptionServiceTour, LaltitudeServiceTour, LongitudeServiceTour, ImageService)
- **Plage** (CodeServiceTour, DenominationPlage, DairaPlage, CommunePlage, LongueurPlage, LargeurPlage, DescriptionServiceTour, LaltitudeServiceTour, LongitudeServiceTour, ImageService)

- **AssociationTouristique** (CodeServiceTour, NomAssociation, PresidenatAssociation, TelAssociation, AgrementAssociation, AdresseAssociation, DescriptionServiceTour, LaltitudeServiceTour, LongitudeServiceTour, ImageService)
- **AgenceVoyage** (CodeServiceTour, NumLicence, NomATV, NomProprietaire, FormeJuridique, CategorieAgence, AdresseAgence, SiteWebAgence, EmailAgence, TelAgence, FaxAgence, DescriptionServiceTour, LaltitudeServiceTour, LongitudeServiceTour, ImageService)
- **SourceThermale** (CodeServiceTour, DesignationSource, CommuneSource, DebitSource, TemperatureSource, ModeExploitation, DescriptionServiceTour, LaltitudeServiceTour, LongitudeServiceTour, ImageService)
 - **Statistique** (CodeStatistique, Continent, Paye, NombreArrive, NombreNuite)
- **Publication** (CodePublication, DescriptionPublication, ImagePublication, DatePublication, HeurPublication, CodeUtilisateur#)
- **ResponsableServiceTourisme** (CodeUtilisateur, EmailUtilisateur, NomUtilisateur, PrenomUtilisateur, IdentifiantUtilisateur, MotPasseUtilisateur, RoleUtilisateur)
- **ResponsableAgenceVoyage** (CodeUtilisateur, EmailUtilisateur, NomUtilisateur, PrenomUtilisateur, IdentifiantUtilisateur, MotPasseUtilisateur, RoleUtilisateur, CodeServiceTour#)
- **ResponsableAssociationTouristique** (CodeUtilisateur, EmailUtilisateur, NomUtilisateur, PrenomUtilisateur, IdentifiantUtilisateur, MotPasseUtilisateur, RoleUtilisateur, CodeServiceTour#)
- **ResponsableHotel** (CodeUtilisateur, EmailUtilisateur, NomUtilisateur, PrenomUtilisateur, IdentifiantUtilisateur, MotPasseUtilisateur, RoleUtilisateur, CodeServiceTour#)
- **GererServiceTouristique** (CodeUtilisateur#, CodeServiceTour#)
- **GererStatistique** (CodeStatistique#, CodeUtilisateur#, Date)

Conclusion

Nous avons présenté dans ce chapitre la phase de conception de notre projet qui contient les diagrammes de séquence, qui nous ont aidés à décrire d'une façon détaillé, le fonctionnement de système dans le but de faciliter la réalisation et la maintenance. Ensuite nous avons conçu le diagramme de classe de notre système qui illustre d'une manière globale la structure des éléments qui constitue la base de données associée à notre application.

Dans le chapitre suivant nous entamons l'étape finale de notre projet qui est l'étape de réalisation et développement.

5

RÉALISATION ET DÉVELOPPEMENT

Introduction

Dans ce chapitre, nous allons faire des choix sur le codage. Nous sommes arrivés pratiquement à la fin du processus de développement. Nous commençons par la présentation de l'environnement de développement, les outils et langages de programmation, ainsi que les services web utilisés dans notre application. Enfin, nous terminons par la présentation des réalisations effectuées au cours de ce projet.

5.1 Environnement de développement

5.1.1 Android Studio

Android Studio est un nouvel environnement pour le développement et programmation entièrement intégré qui a été récemment lancé par Google pour les systèmes Android, il a été conçu pour fournir un environnement de développement et une alternative à Eclipse qui est l'IDE le plus utilisé.

Android Studio permet de voir chacun des changements visuels que vous effectuez sur votre application et en temps réel, vous pourrez voir aussi son effet sur différents appareils Android.

Android Studio offre aussi d'autres choses :

- Un environnement de développement robuste.
- Une manière simple pour tester les performances sur d'autres types d'appareils.
- Des assistants et des modèles pour les éléments communs trouvés sur tous les programmeurs Android.
- Un éditeur complet avec une panoplie d'outils pour accélérer le développement de votre application. [24]

5.1.2 Eclipse

Eclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

La spécificité d'Eclipse vient du fait de son architecture totalement développée autour de la notion de plug-in : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in. [25]

5.1.3 WampServer

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PhpMyAdmin pour gérer plus facilement vos bases de données. [26]

5.1.4 PhpMyAdmin

PhpMyAdmin est un outil logiciel gratuit écrit en PHP , destiné à gérer l'administration de MySQL sur le Web. Prend en charge une large gamme d'opérations sur MySQL tel que la gestion des bases de données, des tableaux, des colonnes, des relations, des index, des utilisateurs, des autorisations, etc. ses opérations peuvent être effectuées via l'interface utilisateur, alors il offre aussi la possibilité d'exécuter directement une instruction SQL. [27]

5.1.5 Apache

Le logiciel libre Apache HTTP est un serveur HTTP créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du World Wide Web. Il est distribué selon les termes de la licence Apache.

5.2 Langages de développement

5.2.1 JAVA

C'est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (JSP). [28]

5.2.2 PHP

PHP (Hypertext Preprocessor) est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web. Il peut être intégré facilement au HTML. [29]

5.2.3 SQL

Le SQL (Structured Query Language) est un langage permettant de communiquer avec une base de données. Ce langage informatique est notamment très utilisé par les développeurs web pour communiquer avec les données d'un site web. SQL.sh recense des cours de SQL et des explications sur les principales commandes pour lire, insérer, modifier et supprimer des données dans une base. [30]

5.2.4 XML

Le XML (eXtensible Markup Language) est un langage informatique qui sert à enregistrer des données textuelles. Ce langage a été standardisé par le W3C en février 1998 et est maintenant très populaire. Ce langage, grosso-modo similaire à l'HTML de par son système de balisage, permet de faciliter l'échange d'information sur l'internet. Contrairement à l'HTML qui présente un nombre fini de balises, le XML donne la possibilité de créer de nouvelles balises à volonté. [31]

5.2.5 JSON

JSON (JavaScript Object Notation – Notation Objet issue de JavaScript) est un format léger d'échange de données. Il est facile à lire ou à écrire pour des humains. Il est aisément analysable ou générable par des machines. Il est basé sur un sous-ensemble du langage de programmation JavaScript. JSON est un format texte complètement indépendant de tout langage, mais les conventions qu'il utilise seront familières à tout programmeur habitué aux langages descendant du C, comme par exemple : C lui-même, C++, C#, Java, JavaScript, Perl, Python et bien d'autres. Ces propriétés font de JSON un langage d'échange de données idéal. [32]

JSON se base sur deux structures :

- Une collection de couples nom/valeur. Divers langages la réifient par un objet, un enregistrement, une structure, un dictionnaire, une table de hachage, une liste typée ou un tableau associatif.
- Une liste de valeurs ordonnées. La plupart des langages la réifient par un tableau, un vecteur, une liste ou une suite. [32]

5.3 Les services web

5.3.1 Définition d'un service web

Selon la définition du W3C (World Wide Web Consortium), un Web service (ou service Web) est une application callable via Internet par une autre application d'un autre site Internet permettant l'échange de données (de manière textuelle) afin que l'application appelante puisse intégrer le résultat de l'échange à ses propres analyses. Les requêtes et les réponses sont soumises à des standards (SOAP, WSDL et UDDI) et normalisées à chacun de leurs échanges.

Un service web est une unité de logique d'application qui fournit des données, et des services aux autres applications, les applications accèdent aux web services via des protocoles, et des formats de données omniprésentes telles que HTTP, XML et SOAP. [33]

5.3.2 Service web utilisé dans notre application

Google Maps Distance Matrix API

Le service Google Maps Distance Matrix API fournit les distances et les durées des trajets pour une matrice d'origines et de destinations en fonction de l'itinéraire recommandé entre les points de départ et d'arrivée. On accède Google Maps Distance Matrix API via une interface HTTP, avec des requêtes construites en tant que chaînes d'URL utilisant les paramètres origines et destinations en plus de notre clé d'API. [34]

L'exemple suivant illustre une requête pour des données de distance entre Bejaia et Adekar, au format JSON :

https://maps.googleapis.com/maps/api/distancematrix/json?units=imperial&origins=36.69421,4.669073&destinations=36.75089,5.056733&key=AlzaSyDDZSQsyPn2qhtwgEATulBk_KHMNd-21RA

Un exemple de réponse est fourni ci-dessous, au format JSON :

```
{
  - destination_addresses: [
    "Boulevard de l'A.L.N, Béjaïa 06000, Algérie"
  ],
  - origin_addresses: [
    "N12, Adegar, Algérie"
  ],
  - rows: [
    - {
      - elements: [
        - {
          - distance: {
            text: "28,7 miles",
            value: 46230
          },
          - duration: {
            text: "1 heure 5 min",
            value: 3884
          },
          status: "OK"
        }
      ]
    }
  ],
  status: "OK"
}
```

FIGURE 5.1 – Exemple d'un format JSON

Google Maps Android API

Google Maps Android API nous permet d'ajouter des cartes utilisant des données Google Maps à notre application. Cette API gère automatiquement l'accès aux serveurs Google Maps, le téléchargement de données, l'affichage de la carte et la réponse aux gestes effectués sur la carte. nous pouvons également utiliser les appels d'API pour ajouter des marqueurs, des polygones et des superpositions à une carte de base, ainsi que pour modifier la vue de l'utilisateur pour une zone donnée. Les objets suivants fournissent des informations supplémentaires sur les points géographiques et permettent à l'utilisateur d'interagir avec la carte. Cette API nous permet d'ajouter les éléments graphiques suivants à une carte :

- Description Icônes ancrées à des emplacements spécifiques sur la carte (Marqueurs).
- Ensembles de segments de lignes (polylignes).
- Segments fermés (polygones).
- Éléments graphiques bitmap ancrés à des emplacements spécifiques sur la carte (superpositions au sol).

Ensembles d'images qui sont affichées au-dessus des tuiles de la carte de base (superpositions de tuiles). [35]

L'exemple suivant illustre une Capture d'écran d'une carte de Google Map :



FIGURE 5.2 – Carte de Google map.

5.4 Schéma de navigation de notre système

pour consulter les différents schémas de navigation de notre système. voir l'annexe à les page (X-XV).

5.5 Quelques interfaces de notre système

5.5.1 Interfaces de l'application Android

Dans ce qui suit, nous allons présenter les interfaces de notre application Android.

Interface d'authentification

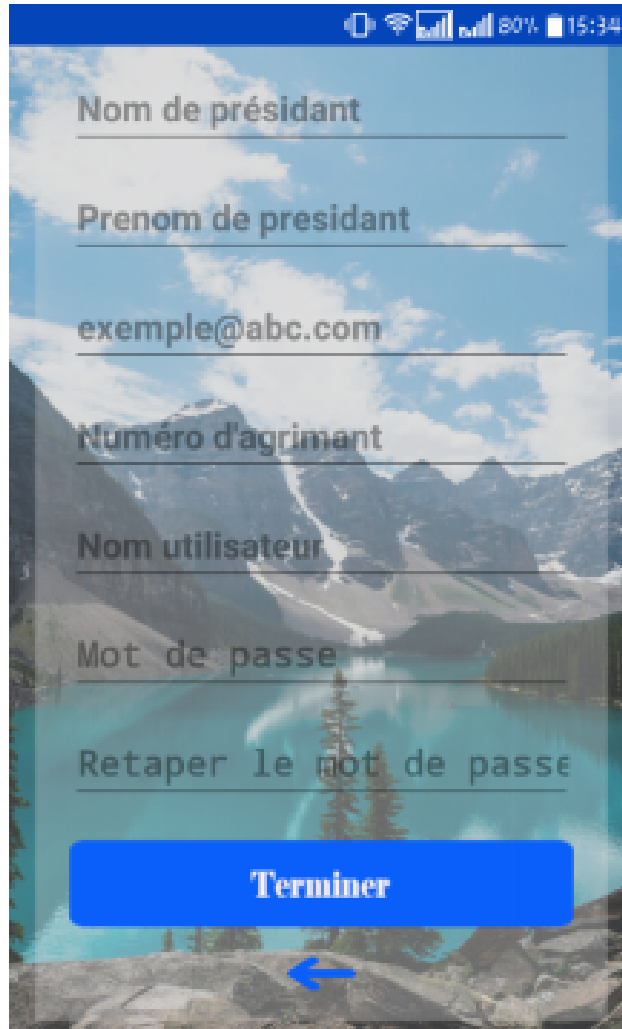
Cette interface offre aux utilisateurs un service d'inscription et d'authentification.



FIGURE 5.3 – Interface d'authentification

Interface d'inscription (Responsable de l'association touristique)

Sur cette interface l'utilisateur doit remplir le formulaire d'inscription, on a choisi comme exemple l'inscription pour le responsable de l'association touristique.



The image shows a mobile application interface for registration. The background is a scenic landscape with mountains and a lake. The form consists of several input fields with labels in French: 'Nom de président', 'Prenom de president', 'exemple@abc.com', 'Numéro d'agrimant', 'Nom utilisateur', 'Mot de passe', and 'Retaper le mot de passe'. A blue button labeled 'Terminer' is at the bottom. A blue arrow points to the left at the bottom of the screen. The top status bar shows signal strength, Wi-Fi, 80% battery, and the time 15:34.

FIGURE 5.4 – Interface d'inscription (Responsable de l'association touristique)

Interface d'accueil

Cette interface illustre un aperçu sur la page d'accueil de l'application.



FIGURE 5.5 – Interface d'accueil

Interface service touristique (Site touristique)

Une fois l'utilisateur se connecte, il pourra accéder aux différents services disponible (Restaurants, Hôtels, Site touristique, Plages, Campings, Sources thermales)

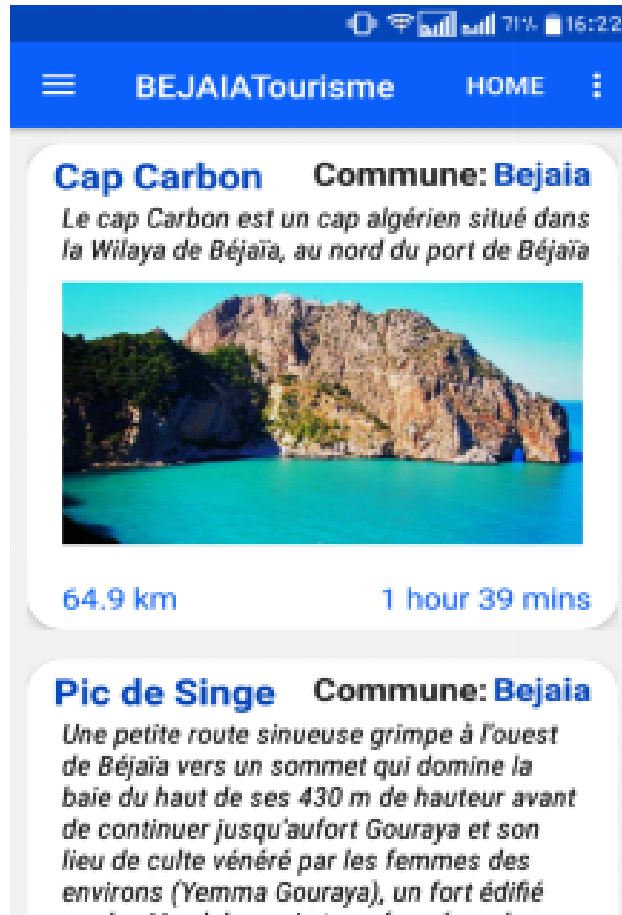


FIGURE 5.6 – Interface service touristique (Site touristique)

Interface détail de service touristique (Site touristique)

Cette interface affiche les détails sur un service touristique choisi.



FIGURE 5.7 – Interface détail de service touristique (Site touristique)

Interface localisation de service touristique (Site touristique)

Cette interface illustre la localisation d'un service touristique choisi et la position courante de l'utilisateur.



FIGURE 5.8 – Interface localisation de service touristique (Site touristique)

Interface gérer statistique

Sur cette interface le responsable de l'hôtel aura la possibilité d'ajouter une statistique.



FIGURE 5.9 – Interface gérer statistique

Interfaces gérer publication

Les interfaces ci-dessous mettent en disposition de responsable de l'association touristique ou de responsable de l'agence de voyage la possibilité d'ajouter, modifier ou supprimer une publication.

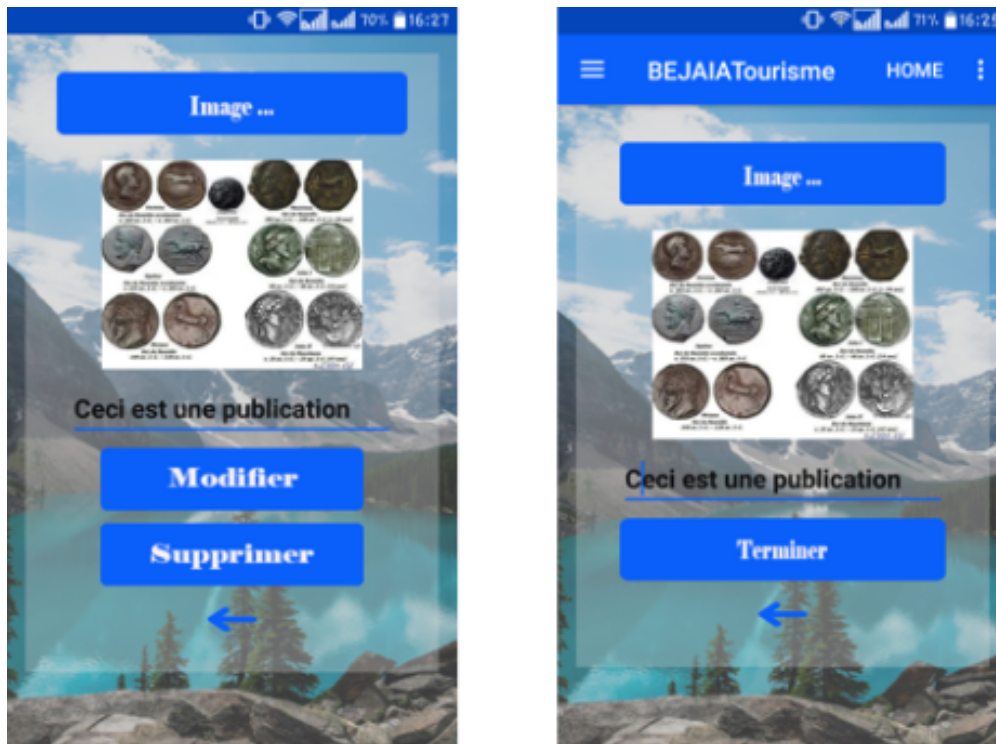


FIGURE 5.10 – Interfaces gérer publication

5.5.2 Interfaces de l'application d'administrateur

Nous allons présenter quelques interfaces de l'application d'administration.

Interface d'authentification

Pour que l'utilisateur puisse utiliser l'application, il doit d'abord s'authentifier en introduisant son identifiant et son mot de passe.



FIGURE 5.11 – Interface d'authentification

Interface gérer un service touristique (Restaurant)

Après l'authentification, l'utilisateur aura l'accès aux différentes fonctionnalités de l'application, Il pourra effectuer des mise à jour sur les services touristiques, ainsi de consulter les statistiques

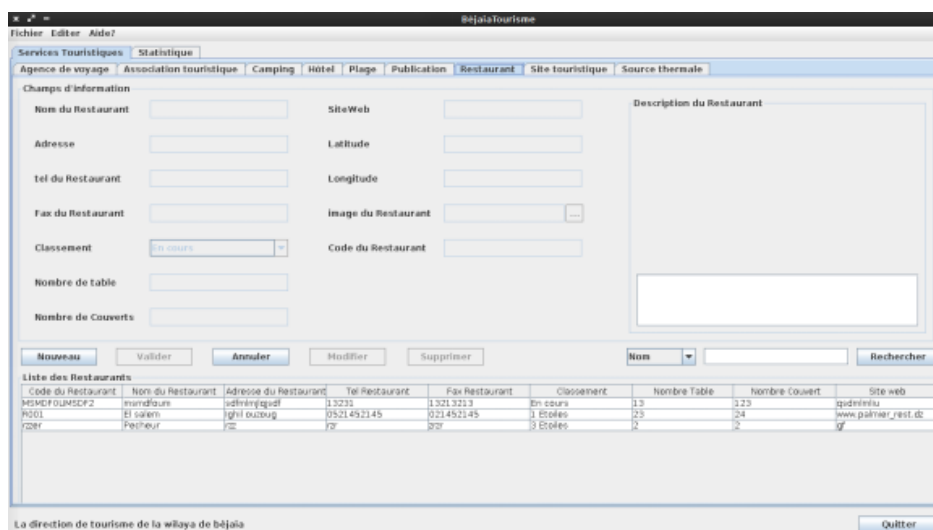


FIGURE 5.12 – Interface gérer un service touristique (Restaurant)

Interface gérer les utilisateurs (responsable agence)

après l'authentification l'administrateur pourra effectuer les opérations dont il a le droit.

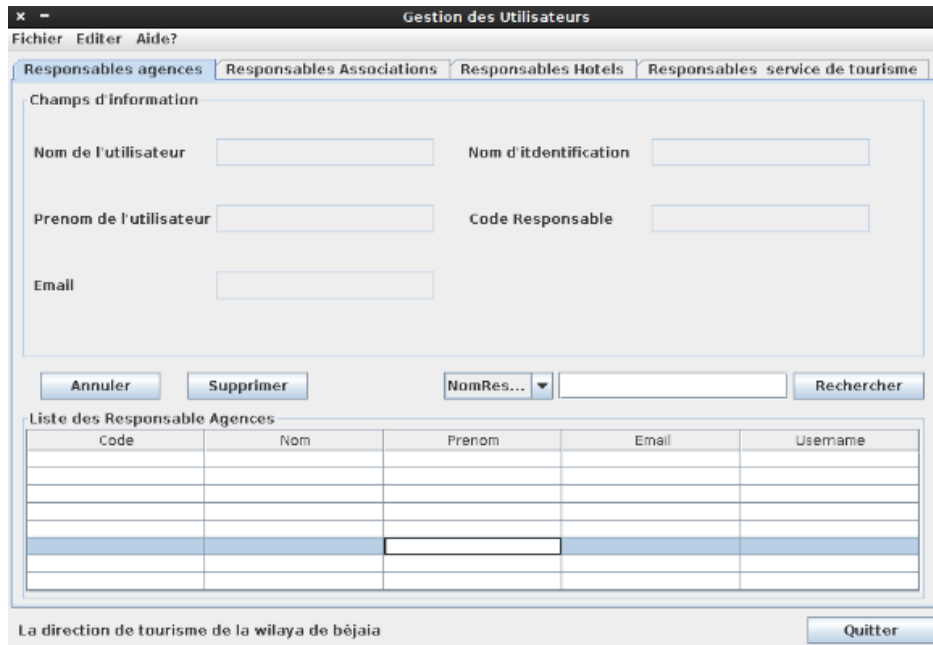


FIGURE 5.13 – Interface gérer les utilisateurs

Conclusion

La phase de réalisation est l'étape la plus importante dans le cycle de vie d'une application. Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de nos applications en spécifiant l'environnement, les outils et les langages de développement associés à notre système. En effet, nous avons achevé l'implémentation tout en respectant la conception élaborée.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Face à l'importance de tourisme au sein de la wilaya de Bejaïa, nous avons étudié, conçu et réalisé à travers de ce travail un système (application) mobile qui permet aux touristes tout au long de leur visite, de rendre le déplacement et la localisation des lieux touristique plus facile et plus rapide.

Nous avons dans un premier temps exposé l'organisme d'accueil, en faisant un aperçu sur la wilaya de Bejaïa. Nous avons présenté dans la deuxième partie l'univers Android, en parlant d'abord sur l'informatique et les applications mobiles, puis nous avons expliqué c'est quoi le système Android et ses applications. Ensuite nous avons entamé les différentes étapes du processus de développement UP, afin de mettre en œuvre notre solution à la problématique.

Nous avons commencé par l'identification des besoins qu'on a ensuite modélisé sous, forme de diagrammes de cas d'utilisation, diagrammes de séquences, diagramme de classe et terminer par la réalisation de notre application.

Lors de ce travail, Nous avons pu mettre en pratique nos connaissances théoriques acquises durant notre formation.

Ce projet a fait l'objet d'une expérience intéressante, Il nous a permis de réaliser que le projet de réalisation d'une application mobile est un ensemble de plusieurs actions planifiées et dépendantes les unes des autres. Toutes les étapes de ce projet nous a permis d'enrichir notre expérience notamment dans les différents outils et langages dédiés à la programmation mobile.

Nous avons retenu également que la réalisation d'une application mobile demande une bonne organisation et une cohérence entre les différents acteurs du projet. Etant donné que tout travail informatique a été toujours l'œuvre d'une équipe.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Bien que notre application n'est pas encore fini, nous avons l'attention de la compléter et l'améliorer en terme design (ergonomie), et quelques fonctionnalités tel que les notifications, l'itinéraire, recherche avancé, sécurité et la déployer au niveau de Play store afin de pouvoir l'exploiter.

Bibliographie

- [1] R. P. Franck Vallée, *UML 2 en action, de l'analyse des besoins à la conception*. 4^{ème} édition, EYROLLES, 2007.
- [2] R. Pascal, *UML 2 Modéliser une application web*. 3^{ème} édition, EYROLLES, 2007.
- [3] A. V. André Pascal, *Développement de logiciels avec UML 2 et OCL*. ellipses, 2013.
- [4] D. Laurent, *de UML en JAVA Conception et réalisation d'une application Web dans le domaine de la sémantique*. 2012.
- [5] “<https://www.grizzlydev.com/developpement/mobilite-informatique/>.” consulté le 31 Mars 2017.
- [6] “http://gdt.oqlf.gouv.qc.ca/ficheoqlf.aspx?id_fiche=8360495.” consulté le 30 Mars 2017.
- [7] “<https://www.techopedia.com/definition/4619/personal-digital-assistant-pda>.” consulté le 31 Mars 2017.
- [8] “<https://www.techopedia.com/definition/2977/smartphone>.” consulté le 31 Mars 2017.
- [9] “<https://www.techopedia.com/definition/2662/tablet-pc>.” consulté le 30 Mars 2017.
- [10] “<https://www.techopedia.com/definition/2953/mobile-application>.” consulté le 2 Avril 2017.
- [11] “https://www.synbioz.com/blog/le_guide_ultime_du_developpement_mobile.” consulté le 2 Avril 2017.
- [12] “<http://www.supinfo.com/articles/single/145-application-mobile-native-web-hybride>consulté.” consulté le 2 Avril 2017.
- [13] S. P. Hébuterne Sylvain, *Android Guide de développement d'applications Java pour Smartphones et Tablettes*. 2^{ème} édition, 2014.
- [14] B. Nazim, *Android 7 Les fondamentaux du développement d'application Java*. Février 2017.
- [15] H. Sylvain, *Développez une application Android programmation en Java sous Android Studio*. Septembre 2015.

- [16] N. Elenkov, *Android Security Internals An in-Depth Guide to Android's Security Architecture*. 2015.
 - [17] U. Serge, *Développez une application Android Programmation en Java sous Eclipse*. 2^{ème} édition, juin 2013.
 - [18] “[http ://www.definitions-marketing.com/definition/application-android/](http://www.definitions-marketing.com/definition/application-android/).” consulté le 2 Avril 2017.
 - [19] R. Meier, *Professional Android Application Development*. 2012.
 - [20] B. Nazim, *Android 5 les fondamentaux du développement d'applications Java*. Avril 2015.
 - [21] “[http ://sabricole.developpez.com/uml/tutoriel/unifiedprocess/](http://sabricole.developpez.com/uml/tutoriel/unifiedprocess/).” consulté le 6 Mai 2017.
 - [22] P. Kuchana, *Software architecture design patterns in Java*. Avril 2004.
 - [23] C. Soutou, *UML 2 pour les bases de données avec 20 exercices corrigés*. EYROLLES, Mars 2007.
 - [24] “[http ://www.techno-science.net/?onglet=glossaire&definition=517](http://www.techno-science.net/?onglet=glossaire&definition=517).” consulté le 06 Juin 2017.
 - [25] “[https ://android-studio.fr.uptodown.com/windows](https://android-studio.fr.uptodown.com/windows).” consulté le 06 Juin 2017.
 - [26] “[https ://www.phpmyadmin.net/](https://www.phpmyadmin.net/).” consulté le 06 Juin 2017.
 - [27] “[http ://www.wampserver.com/](http://www.wampserver.com/).” consulté le 08 Juin 2017.
 - [28] “[http ://glossaire.infowebmaster.fr/xml/](http://glossaire.infowebmaster.fr/xml/).” consulté le 08 Juin 2017.
 - [29] “[http ://www.futura-sciences.com/tech/definitions/internet-java-485/](http://www.futura-sciences.com/tech/definitions/internet-java-485/).” consulté le 08 Juin 2017.
 - [30] “[http ://php.net/manual/fr/intro-what-is.php](http://php.net/manual/fr/intro-what-is.php).” consulté le 08 Juin 2017.
 - [31] “[http ://sql.sh/](http://sql.sh/).” consulter le 08 Juin 2017.
 - [32] “[http ://slideplayer.fr/slide/173472/](http://slideplayer.fr/slide/173472/).” consulté le 08 Juin 2017.
 - [33] “[http ://www.json.org/json-fr.html](http://www.json.org/json-fr.html).” consulté le 10 Juin 2017.
 - [34] “[https ://developers.google.com/maps/documentation/distance-matrix/start?hl=fr](https://developers.google.com/maps/documentation/distance-matrix/start?hl=fr).” consulté le 10 Juin 2017.
 - [35] “[https ://developers.google.com/maps/documentation/android-api/intro](https://developers.google.com/maps/documentation/android-api/intro).” consulté le 10 Juin 2017.
-

A

ANNEXE

DIAGRAMMES DE SÉQUENCE SYSTÈME

A.1 Diagramme de séquence système " Gérer un Utilisateur "

la figure suivante représente le diagramme de séquence système "Gérer un Utilisateur".

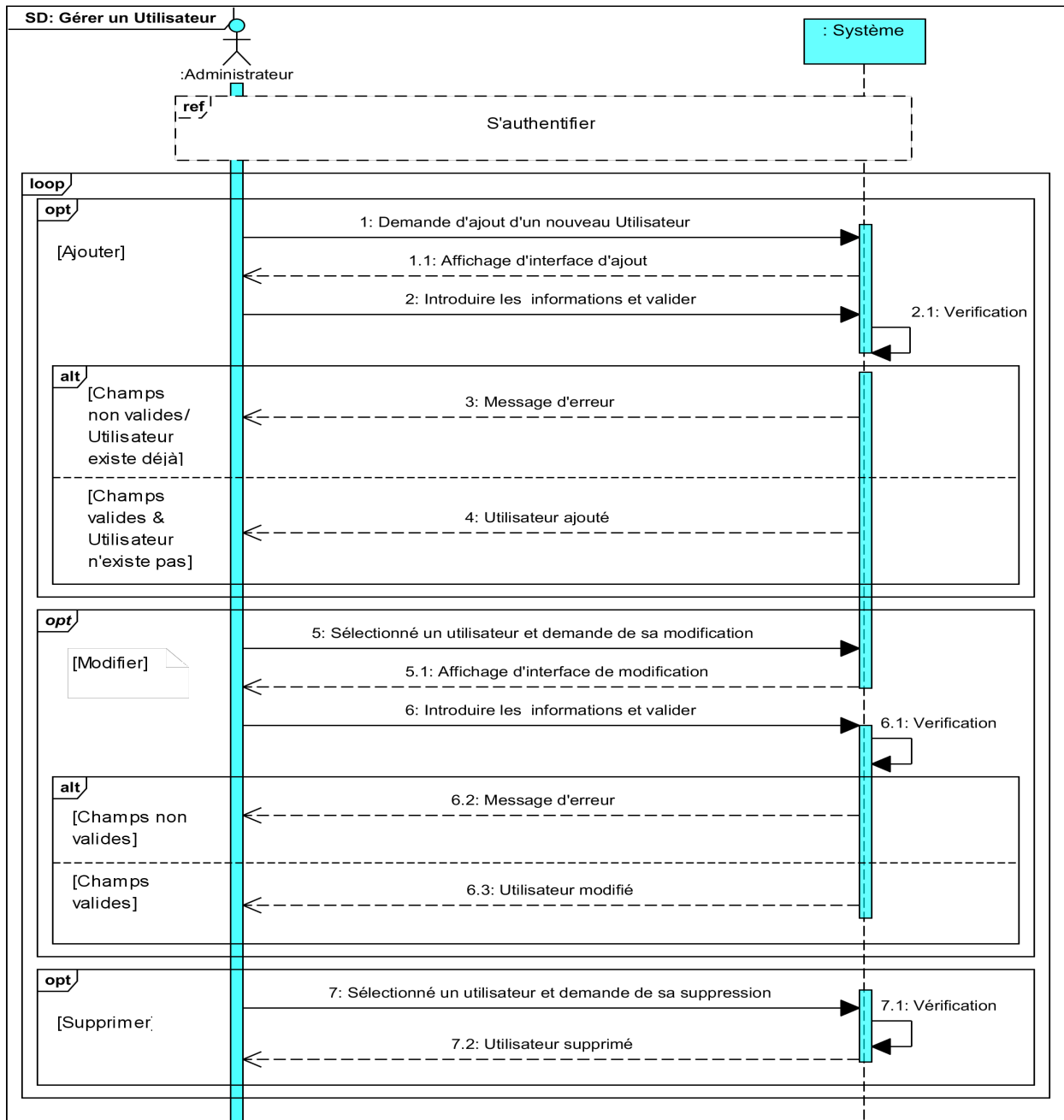


FIGURE A.1 – Diagramme de séquence système " Gérer un Utilisateur "

A.2 Diagramme de séquence système " Gérer un ServiceTouristique "

la figure suivante représente le diagramme de séquence système "Gérer un ServiceTouristique"..

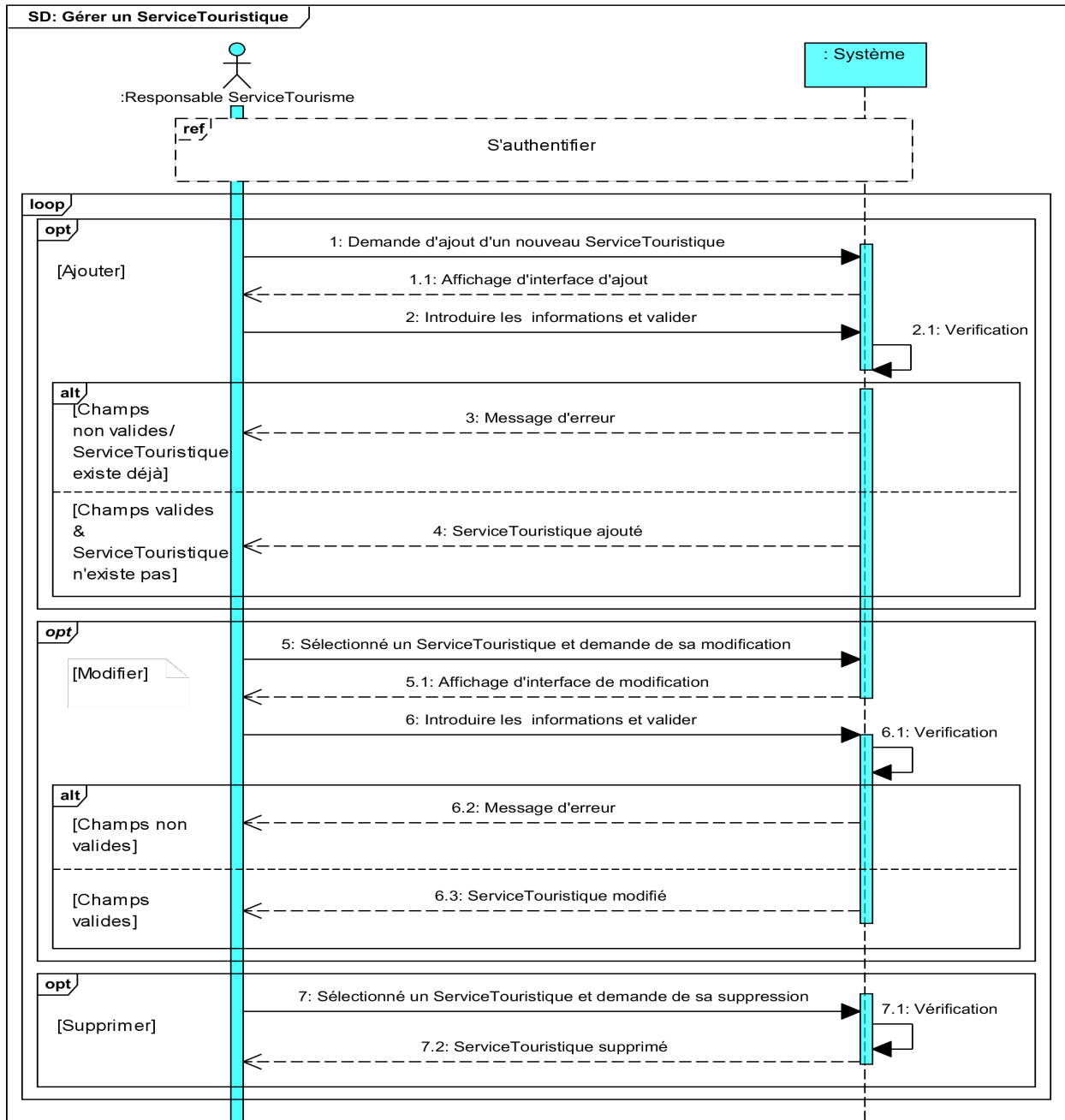


FIGURE A.2 – Diagramme de séquence système " Gérer un ServiceTouristique "

A.3 Diagramme de séquence système " Suivre une Statistique "

La figure suivante représente le diagramme de séquence système "suivre une statistique".

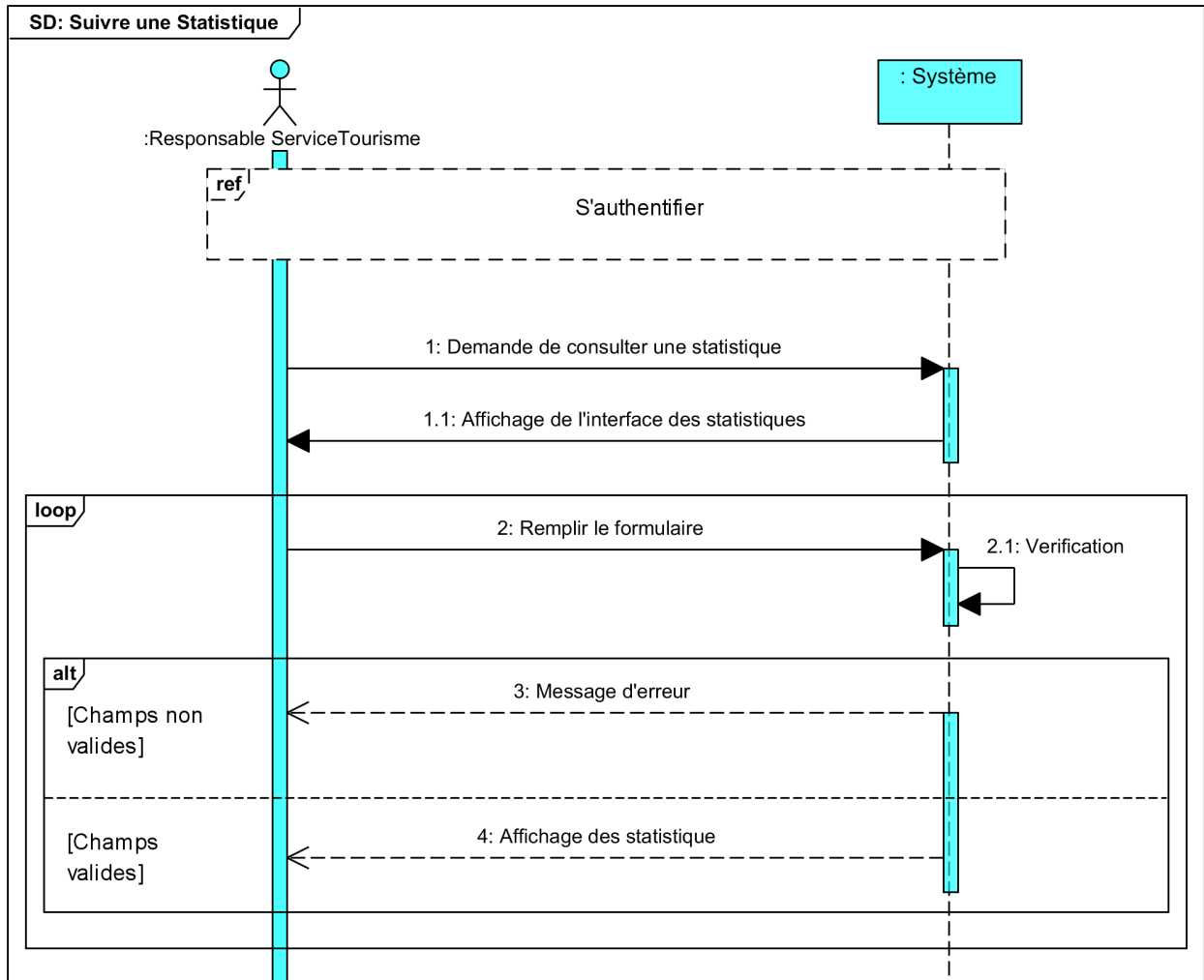


FIGURE A.3 – Diagramme de séquence système " Suivre une Statistique "

A.4 Diagramme de séquence système " Gérer une Statistique "

La figure suivante représente le diagramme de séquence système "Gérer une Statistique".

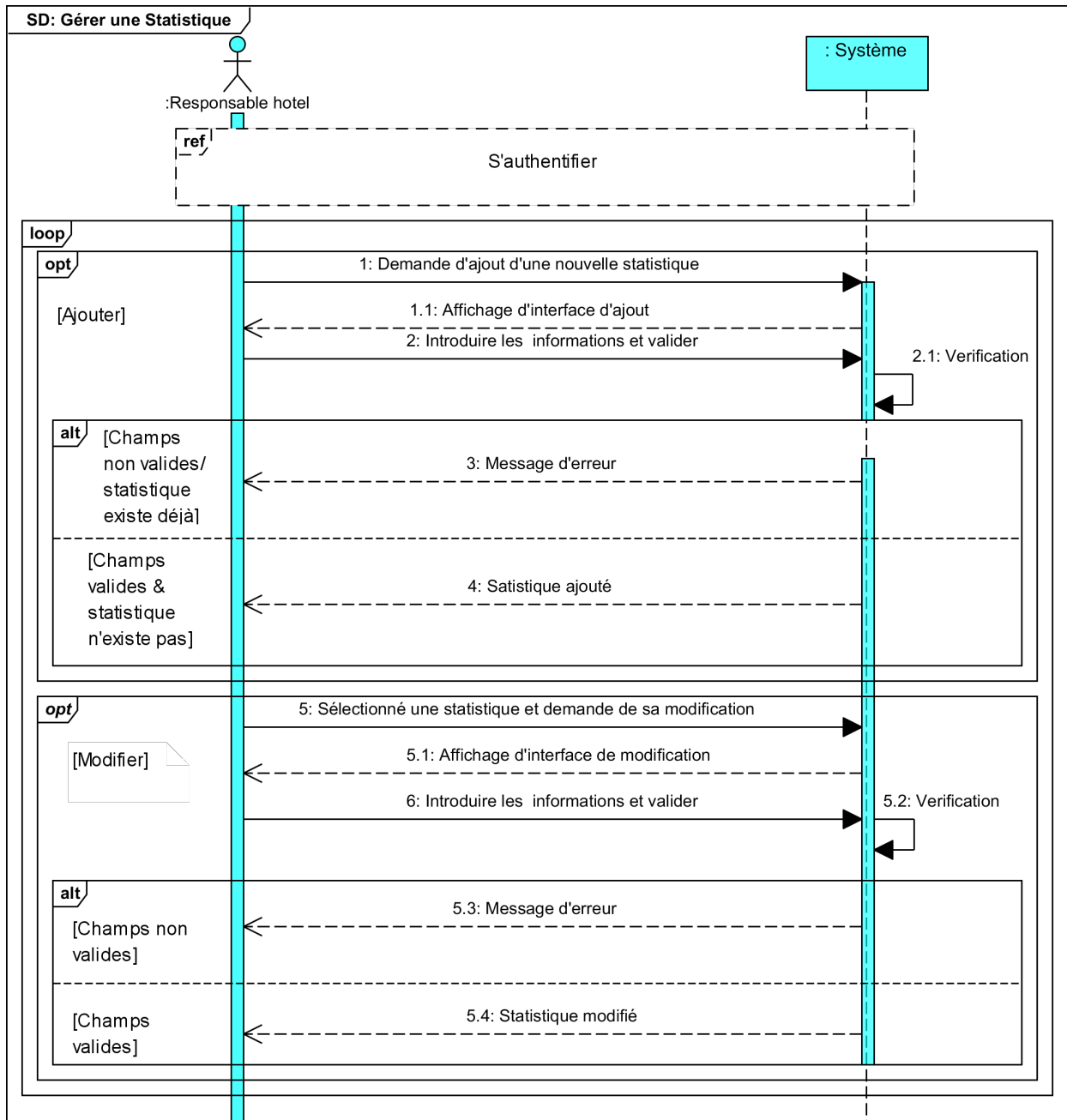


FIGURE A.4 – Diagramme de séquence système " Gérer une Statistique "

A.5 Diagramme de séquence système " Consulter un ServiceTouristique "

La figure suivante représente le diagramme de séquence système "Consulter un ServiceTouristique".

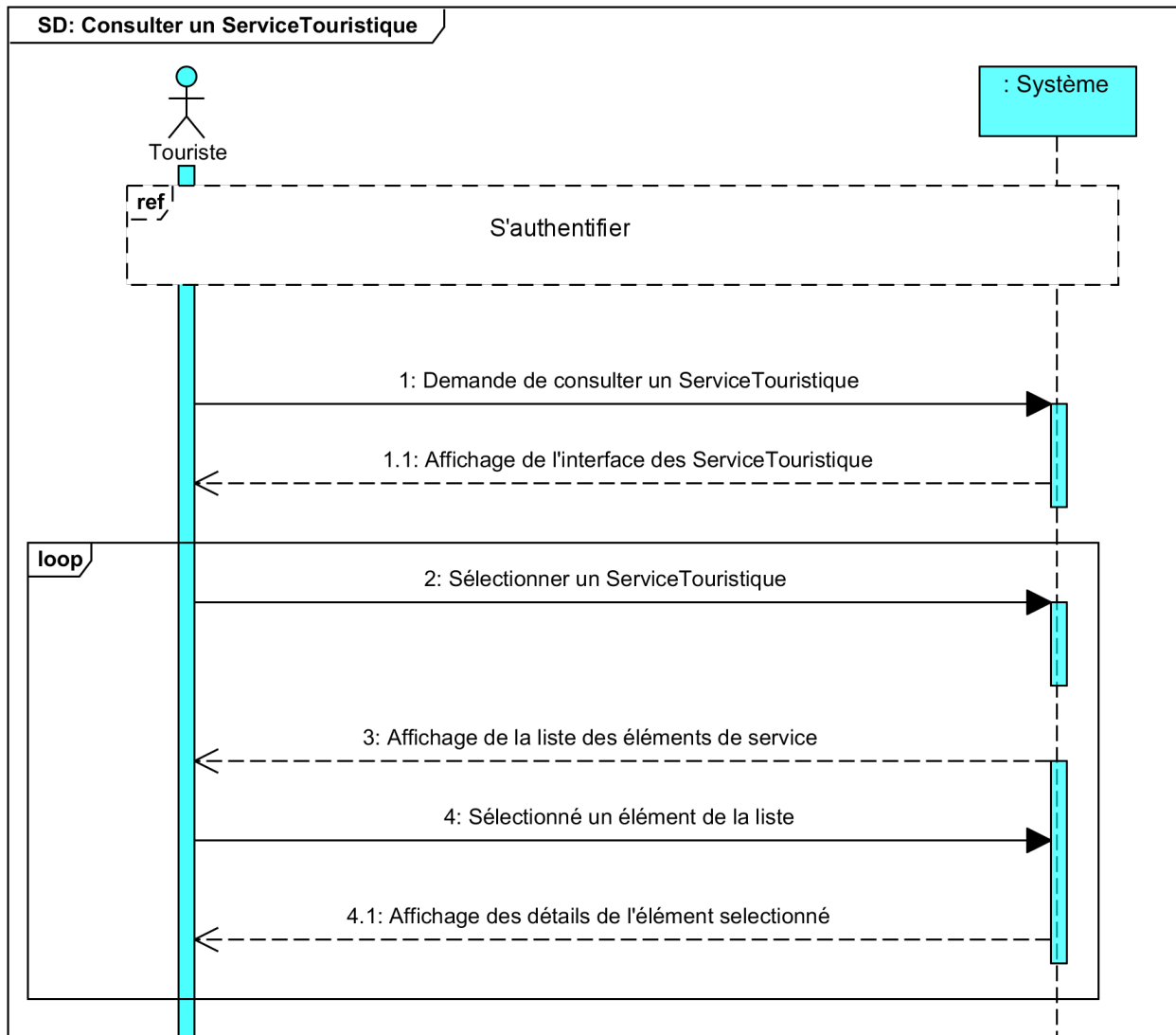


FIGURE A.5 – Diagramme de séquence système " Consulter un ServiceTouristique "

A.6 Diagramme de séquence système "s'inscrire"

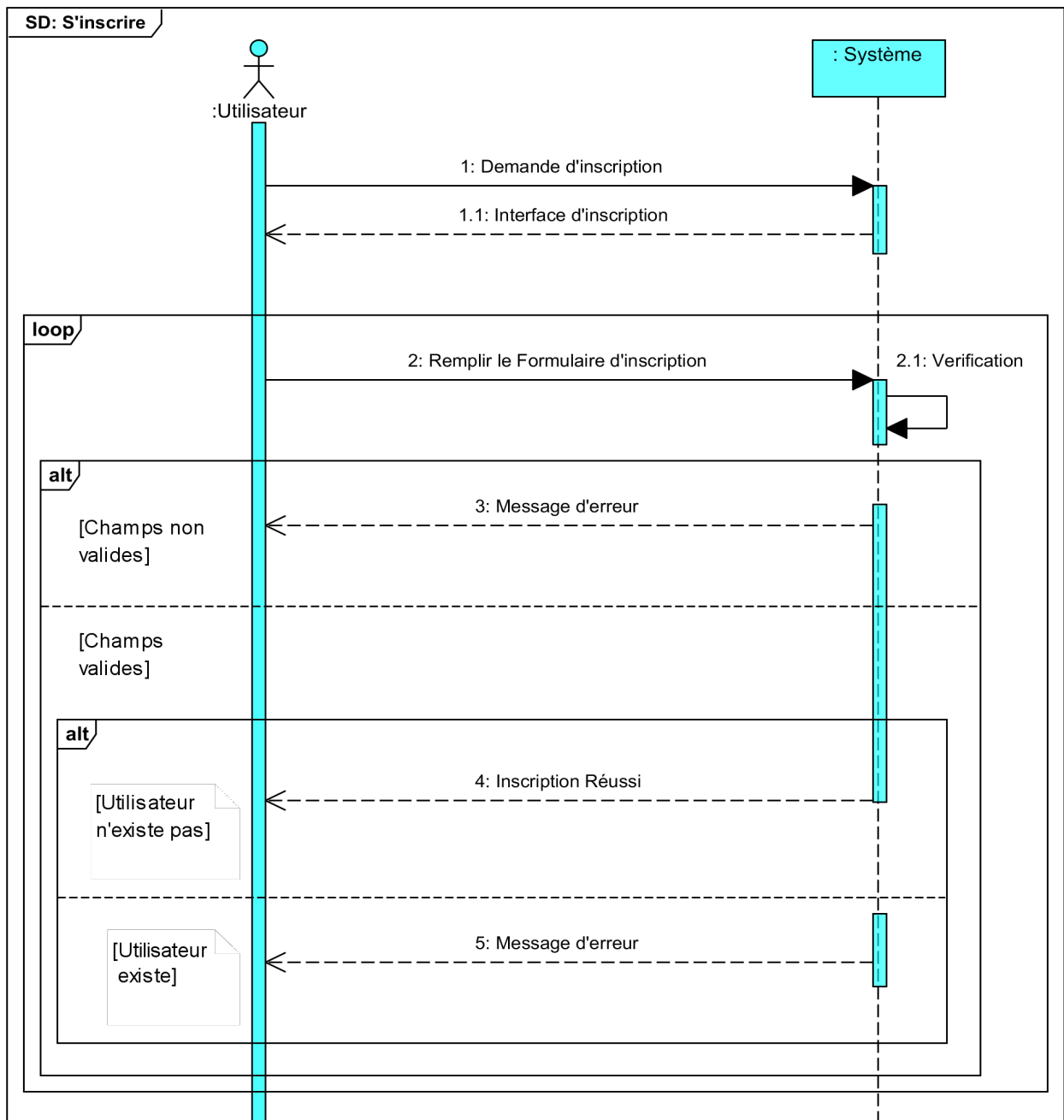


FIGURE A.6 – Diagramme de séquence système " s'inscrire"

A.7 Diagramme de séquence système " s'authentifier "

la figure ci-dessous représente le diagramme de séquence système pour "L'authentification".

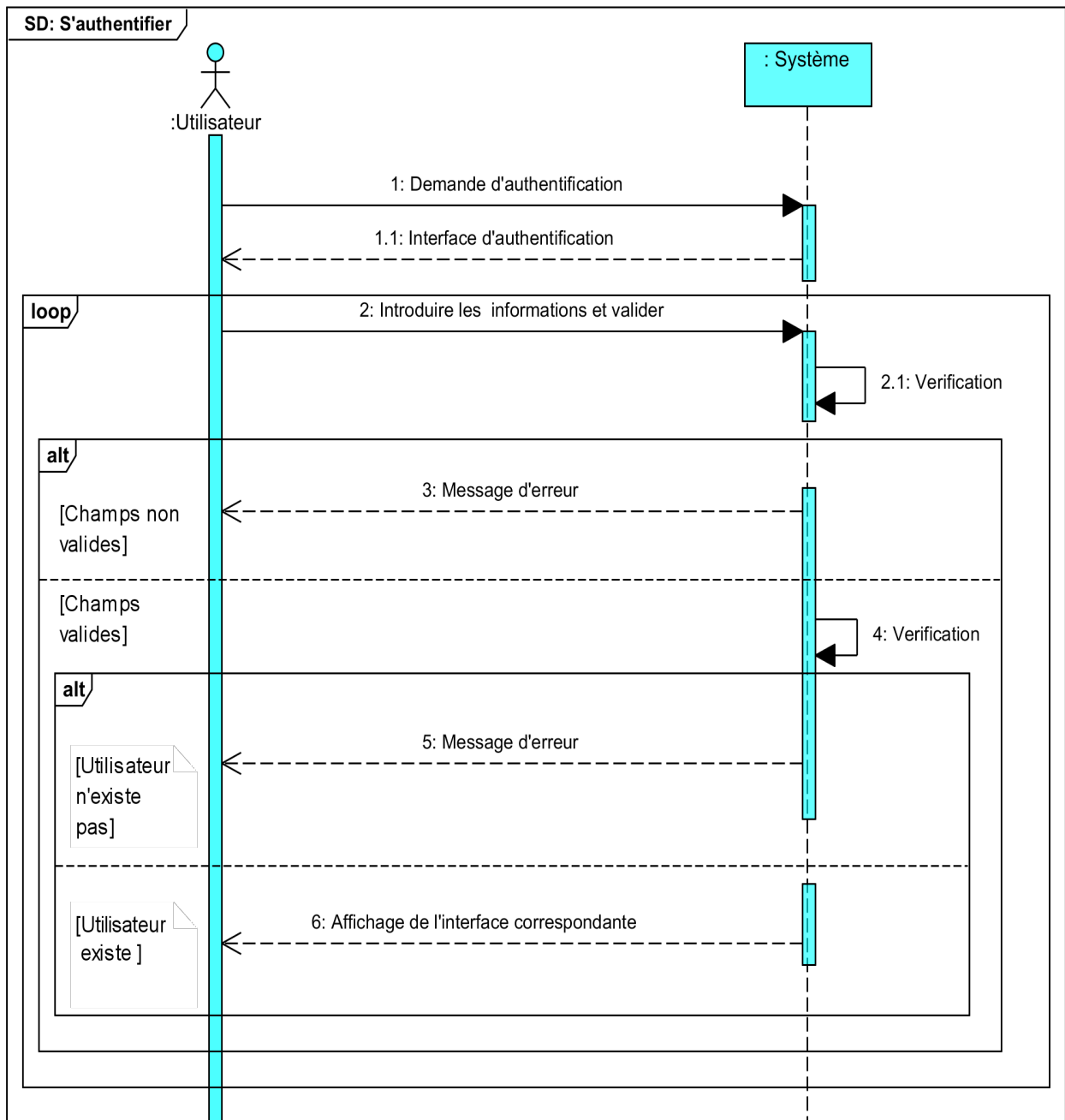


FIGURE A.7 – Diagramme de séquence système " s'authentifier "

A.8 Diagramme de séquence système " Gérer une Publication "

cette figure illustre le diagramme de séquence système pour le cas d'utilisation "Gérer une Publication".

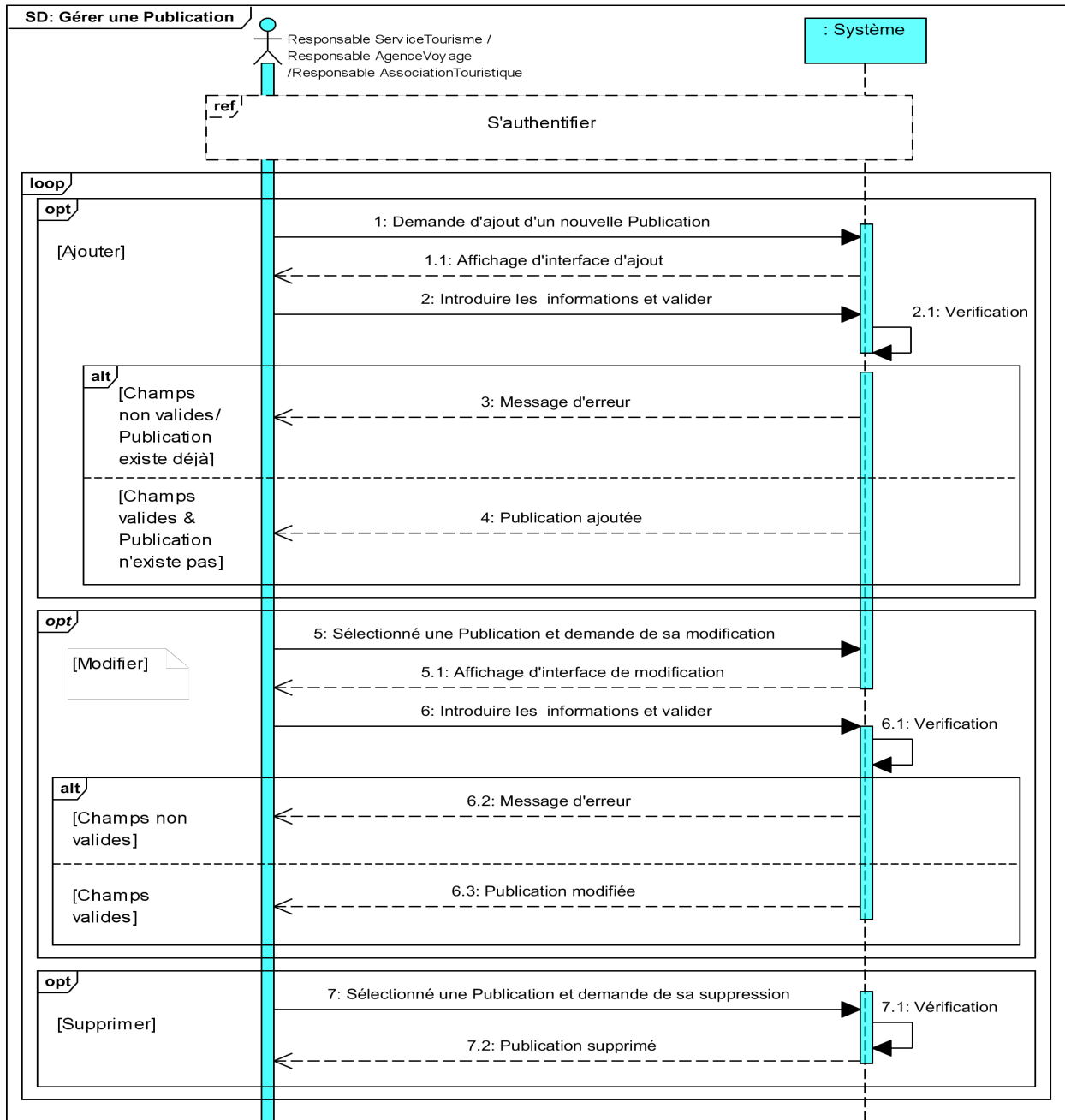


FIGURE A.8 – Diagramme de séquence système " Gérer une Publication "

B

ANNEXE

SCHÉMAS DE NAVIGATION DU SYSTÈME

Schémas de navigation de notre système

B.1 Côté touriste

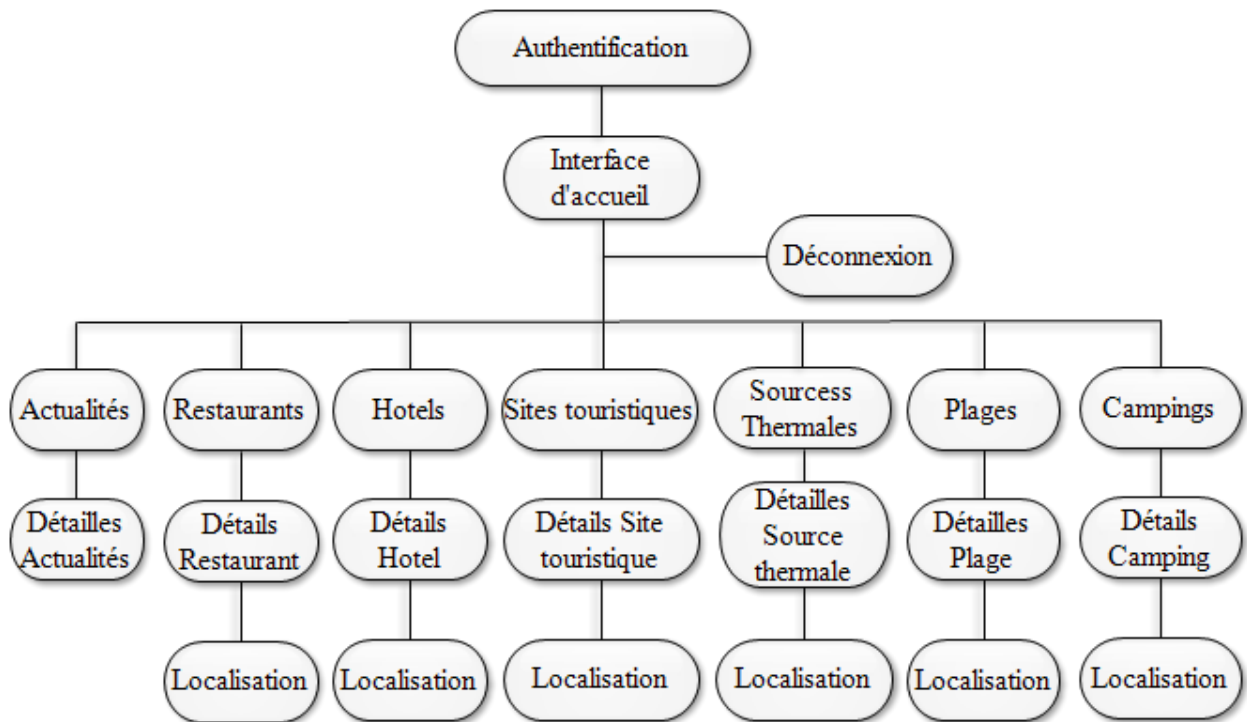


FIGURE B.1 – Schémas de navigation de notre système "côté touriste"

B.2 Côté Responsable agence de voyage et Responsable association touristique

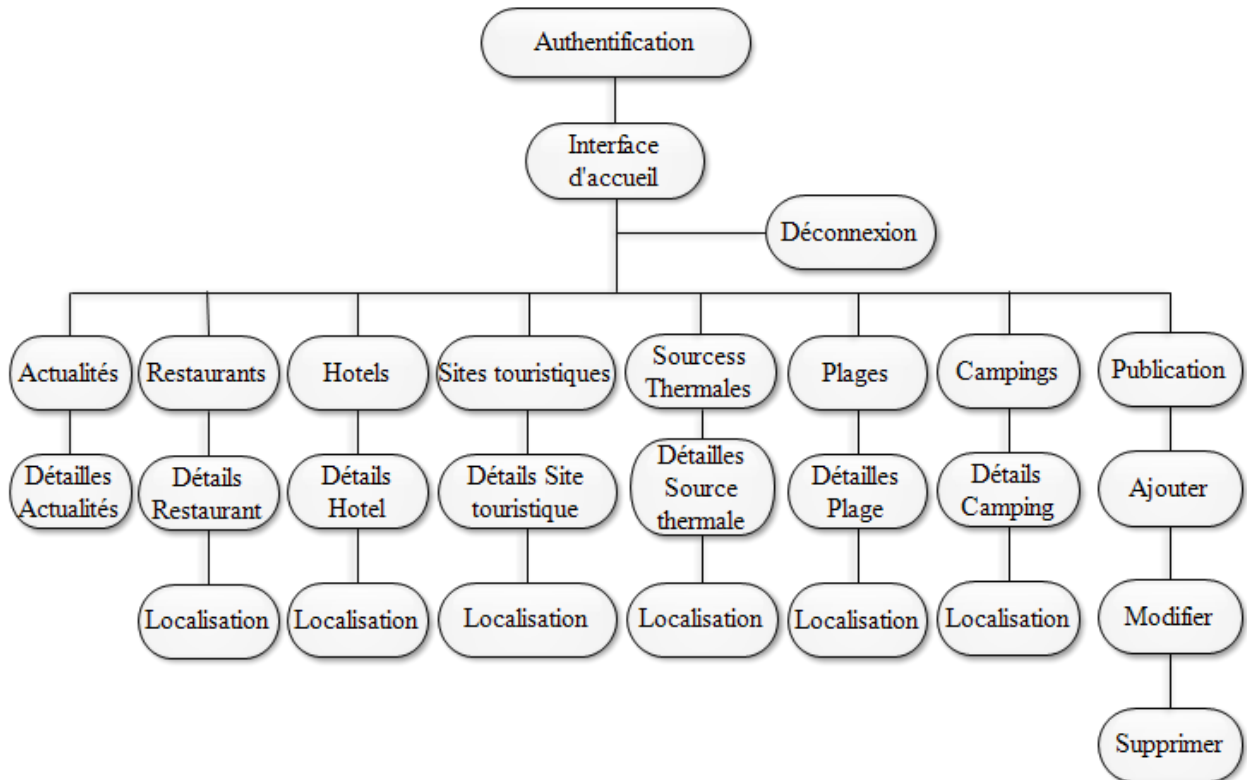


FIGURE B.2 – Schémas de navigation de notre système "côté responsable agence de voyage et responsable association touristique"

B.3 Côté Responsable d'hôtel

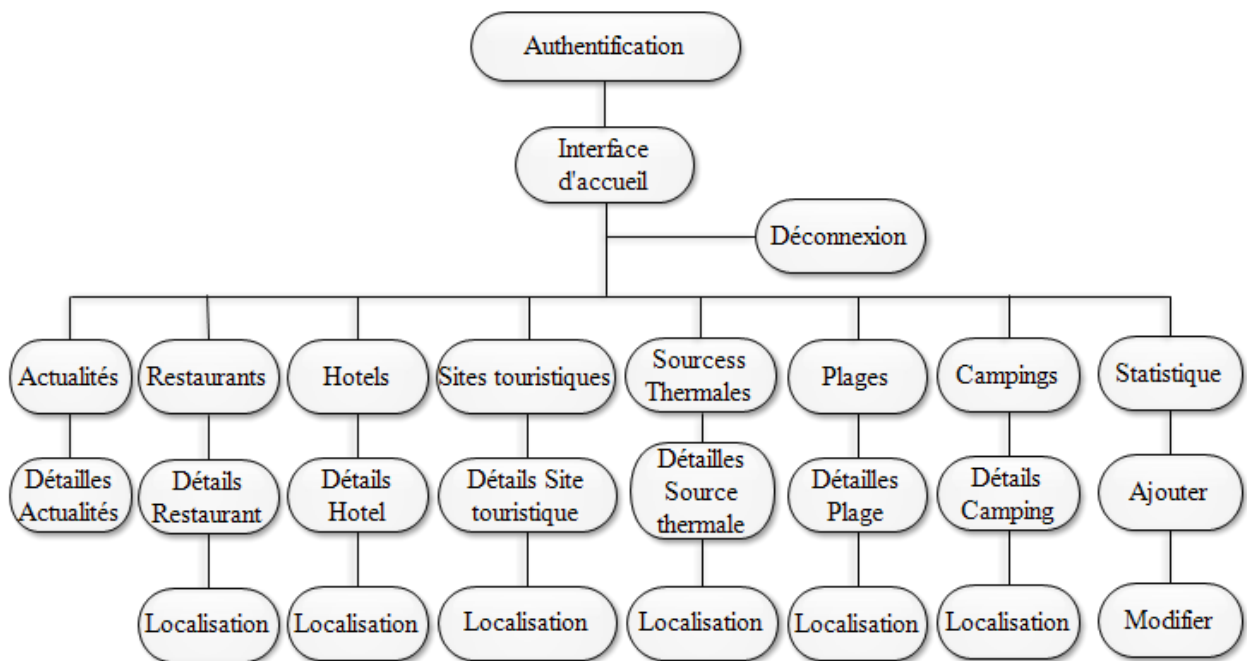


FIGURE B.3 – Schémas de navigation de notre système "côté responsable d'hôtel"

B.4 Côté administrateur

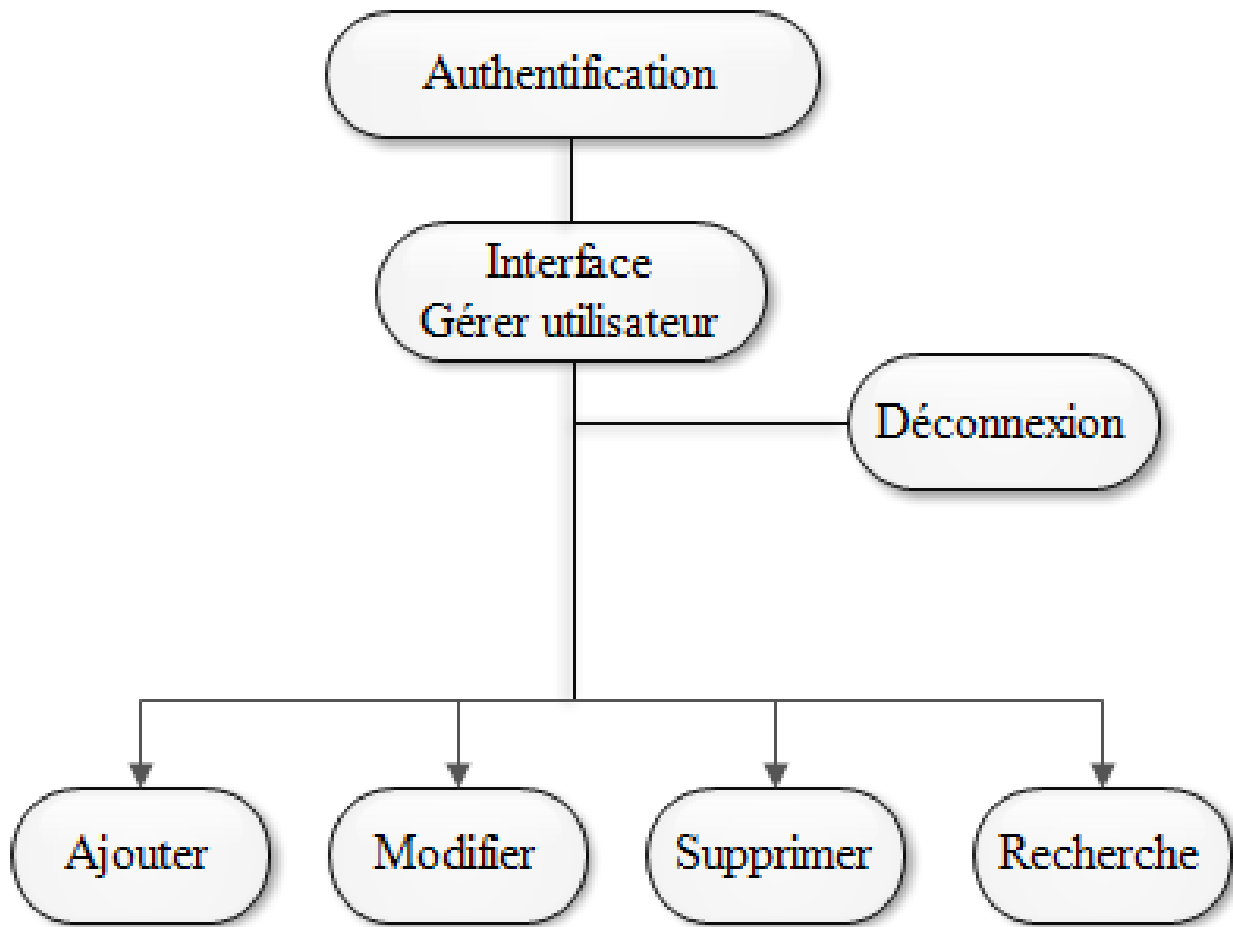


FIGURE B.4 – Schémas de navigation de notre système "côté administrateur"

B.5 Côté responsable de service tourisme

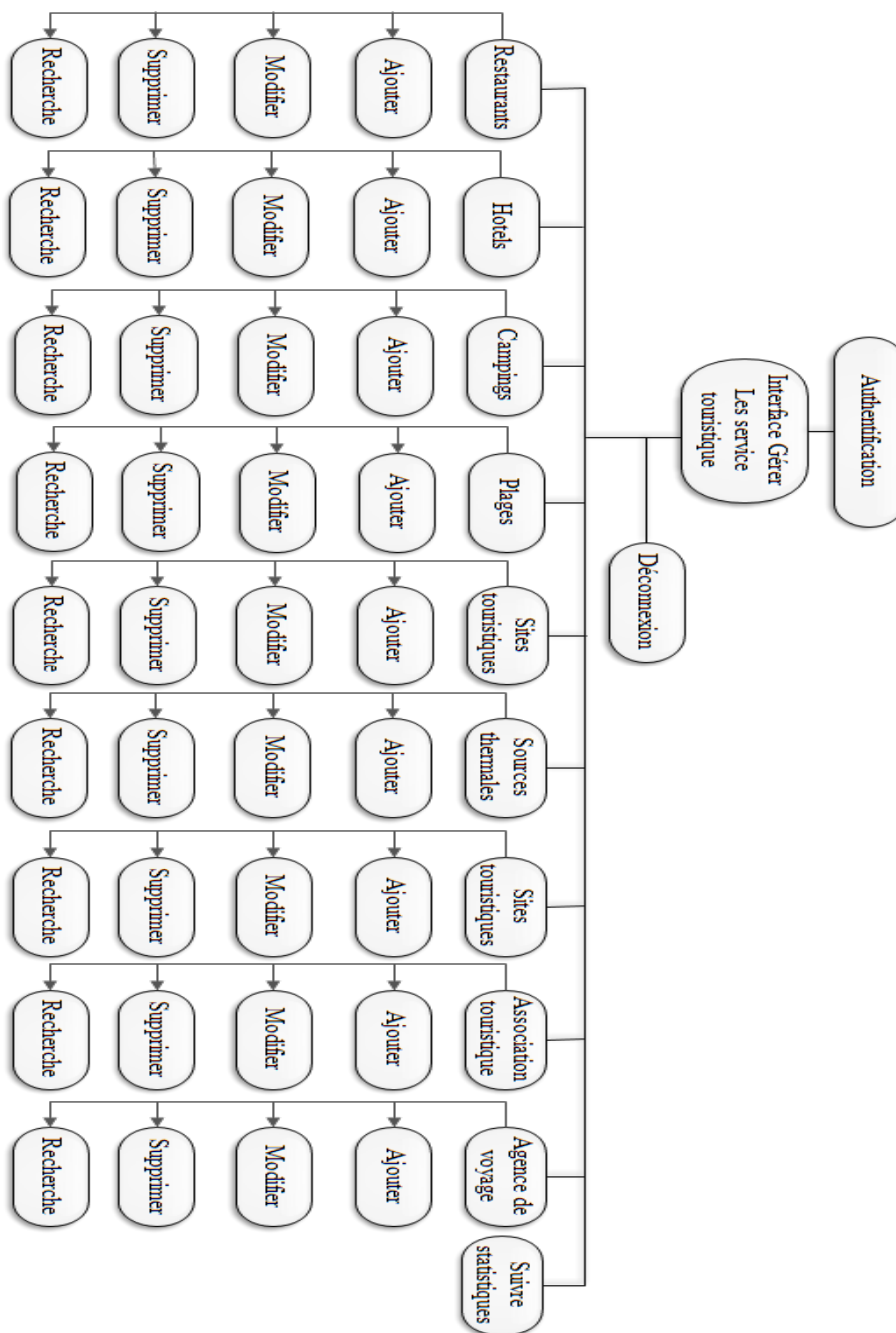


FIGURE B.5 – Schémas de navigation de notre système "côté responsable de service tourisme"

Résumé

Le but de ce travail est de réaliser une application mobile fiable et aisée, munie de fonctionnalités nécessaires et adéquates au besoins des touristes.

Pour mettre en œuvre notre solution, nous avons utilisé un processus de développement appelé Processus Unifié (UP), qui se base sur UML comme langage de modélisation conçu pour fournir une méthode normalisée pour la conception, et la construction des documents nécessaires au bon développement.

Cette conception est mise en œuvre sous l'environnement de développement Intégré (Eclipse et Android Studio), java et php comme langages de programmation, un système de gestion de base de données MySQL et le serveur apache, sans oublier quelques services web tel que Google Maps Android API.

Mots clés : UP,UML,java, PHP, XML,SQL,WampServer,PhpMyadmin, Apache, MySQL, Android,Json,Service Web, MVC.

Abstract

The aim of our project is to release and develop a reliable and easy application, equipped with necessary functionalities and adapted to the needs of the tourists.

To develop our solution, we used a development process called Unified Process (UP), which is based on UML as a modeling language designed to provide a standardized method for designing and building the documents necessary for good development. This design is implemented under the Integrated Development Environment (Eclipse and Android Studio), java and php as programming languages, a MySQL database management system and the apache server, without forgetting a few web services such as Google Maps Android API.

Keywords : UP,UML,java, PHP, XML,SQL,WampServer,PhpMyadmin, Apache, MySQL, Android,Json,Service Web, MVC.