

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire
Abd elhafid boussouf Mila

Institut des sciences et de la technologie

Département de Mathématiques et Informatique

**Mémoire préparé En vue de l'obtention du diplôme de
Master**

EN :Filière informatique

Spécialité sciences et technologies de l'information et de la communication stic

Application à base des web services pour les régimes alimentaires

Préparé par :

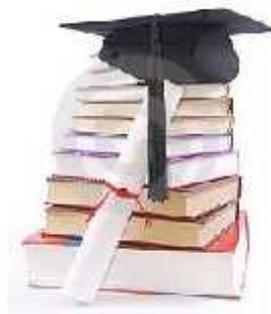
Baouche Zina
Athimene Rokia

Soutenue devant le jury

Président :Madame Talai MeriemeMAA.
Examinateur : Madame Bouchemal Nardjas.....MCB.
Examinateur : Madame Harrich Ouahiba.....MAA.
Encadreur : Madame Abderrezek Samira.....M.A.A.

Année universitaire : 2014/2015

Remerciement



Nous remercions tout d'abord Allah le tout puissant de nous avoir aidé et donné la patience et le courage durant ces longues et dures années d'études,

Nos encadrateurs, Madame « Madame Abderrezek Samira » tout au long de la réalisation de ce mémoire, ainsi pour l'orientation, la confiance, les précieux conseils, la patience et l'aide,

Nous vifs remerciment vont également les membres du jury

«Harrich Ouahiba»

«Madame Talai Merieme» « Bouchemal Nardjas »

Nous tenons à remercier L'ensemble de nos professeurs de Master 2 STIC,

Toute personne ayant de près ou de loin contribué à l'élaboration de ce travail un grand merci.

Zina & Rokia

Sommaire

Table des matières

Introduction Générale

Introduction	1
1.1. Discours alimentaire selon les époques et les cultures.....	2
1.1.1. L'Antiquité.....	3
1.1.2. Le haut Moyen-Âge.....	4
1.1.3. Du bas Moyen-Âge au XVIIème siècle.....	4
1.1.4. XVIIème et XVIIIème siècles.....	5
1.1.5. Le XIXème siècle.....	5
1.1.6. XXème et XXIème siècle.....	6
1.2. Nutrition et alimentation	6
1.2.1. Définitions.....	6
1.2.2. Nature et rôle des constituants alimentaires.....	7
1.2.2.1. Les aliments organiques.....	7
1.2.2.2. Les aliments minéraux.....	8
1.2.2.3. Aliments simples, aliments composés.....	9
1.2.2.4. Aliments complets, aliments incomplets.....	10
1.2.2.5. Rôle des aliments ingérés chez l'homme.....	11
1.2.2.6. Les aliments.....	12
1.2.2.7. Besoins nutritionnels et apports recommandés.....	13
1.2.3. Elaboration des rations alimentaires.....	15
1.2.4. L'alimentation équilibrée.....	16
1.2.4.1. Définition	16
1.2.4.2. Maîtriser quantité et qualité.....	17
1.2.4.3. Les cinq clés de l'équilibre alimentaire.....	17
1.2.4.4. Equilibre et régime alimentaires.....	18
1.2.4.5. Déséquilibre alimentaire et maladies nutritionnelles.....	19
1.3. Le régime alimentaire.....	19

1.3.1. Ce que nous dit l'histoire des régimes.....	19
1.3.2. Définitions.....	21
1.3.3. Les critères de classifications des régimes alimentaires.....	22
1.3.3.1. La diversité des régimes alimentaires.....	22
1.3.4. Le suivre d'un régime.....	23
1.3.5. Un régime n'est pas sans risque.....	24
1.3.6. Le bon régime.....	24
1.4. La structure proposée pour un régime alimentaire.....	24
Conclusion.....	25

Chapitre2

Introduction	26
2.1. L'évolution des réseaux informatiques.....	26
2.1.1. Le système central.....	27
2.1.2 Le modèle client/serveur.....	27
2.1.3. Le modèle internet.....	30
2.1.4. Le modèle web services.....	32
2.2. L'architecture orientée service.....	33
2.2.1. Service.....	33
2.2.2. Histoire de la SOA.....	33
2.2.3. Définition.....	33
2.2.4. Concepts de SOA.....	35
2.2.5. Les composants de la SOA.....	35
2.2.6. L'accès aux services.....	37
2.2.7. Avantages de l'architecture orientée service.....	37
2.3. Les Web Services.....	38
2.3.1. L'apparition des Web Services.....	38
2.3.2. Définitions.....	40
2.3.3. Les caractéristiques et l'intérêt des services Web.....	41
2.3.4. Les applications des services Web.....	42
2.3.5. Les types de services web.....	43

2.3.7. Architecture des services des web.....	43
2.3.7.1. Architecture de base.....	43
2.3.7.2. Architecture en couche.....	44
2.3.8. Le cycle de vie d'un service web et le fonctionnement d'un service web.....	47
2.3.8.1. Le cycle de vie d'un service web.....	47
2.3.8.2. Le fonctionnement d'un service web.....	48
2.3.9. Les standards du service Web.....	49
2.3.9.1. XML – EXtensible Markup Language.....	49
2.3.9.2. Le protocole SOAP (simple Object Access Protocol)	52
2.3.9.3. WSDL (Web Service Description Language)	58
2.3.9.4. UDDI (Universal Description, Discovery and Integration)	61
2.3.10. Problématique de recherche sur les services Web.....	63
2.3.10.1. Sélection des services Web.....	63
2.3.10.3. La découverte dynamique d'un service Web.....	64
2.3.10.4. Composition des services Web.....	65
2.3.11. Les avantages et les inconvénients des services web.....	67
2.3.11.1. Les avantages.....	67
2.3.11.2. Les inconvénients.....	68
Conclusion.....	69

Chapitre3

Introduction	70
3.1. Processus de développement logiciel.....	70
3.1.1. Processus unifié (unified process)	70
3.2. Avantages d'un processus itératif contrôlé.....	72
3.3. Le processus 2TUP.....	72
3.3.1 Les Diverses phases de 2TUP	73
3.3.1.1. Etude préliminaire.....	73
3.3.1.2. Branche gauche (fonctionnels)	74
3.3.1.4. La branche du milieu.....	74
Conclusion.....	75

Chapitre4

Introduction	76
4.1. L'élaboration du cahier de charges.....	76
4.1.1. Présentation du projet.....	76
4.1.2. Choix techniques.....	76
4.1.3. Recueil des besoins fonctionnels.....	77
4.1.3.1. Les besoins fonctionnels au niveau des fournisseurs des régimes alimentaires	77
4.1.3.2. Les besoins fonctionnels au niveau des utilisateurs des régimes alimentaires	77
4.1.4. Recueil des besoins opérationnels.....	77
4.2. Description du contexte.....	78
4.2.1. Identifications des acteurs.....	78
4.2.1.1. Au niveau des fournisseurs des régimes alimentaires.....	78
4.2.1.2. Au niveau de l'utilisateur des régimes alimentaires.....	79
4.2.1.3. Au niveau de l'UDDI.....	79
4.2.2. Identifications des messages.....	80
4. 2.2. 1. Au niveau de fournisseur des régimes alimentaires.....	80
4. 2.2. 2. Au niveau de l'utilisateur des régimes alimentaires.....	80
4. 2.2. 3. Au niveau de l'UDDI.....	81
4.2.3. Diagramme de contexte dynamique.....	83
4.3. Modélisation du contexte.....	85
Conclusion.....	86

Chapitre6

Introduction	157
6.1. Spécification technique du point de vue matériel.....	157
6.1.1. Configuration matérielle.....	157
6.1.2. Spécification d'architecture web service.....	158
6.2. Capture des spécifications logicielles.....	158
6.2.1. Les exploitants.....	158
6.2.2. Identification des cas d'utilisation techniques.....	159

6.2.3. Description des cas d'utilisation techniques.....	159
6.2.3.1. Cas d'utilisation : «Gérer la sécurité »	159
6.2.3.2. Cas d'utilisation : « Gérer l'intégrité des données »	160
6.3. Organisation du modèle de spécification logicielle.....	162
Conclusion.....	163

Chapitre7

Introduction	164
7.1. Découpage en catégories.....	164
7.1.1. La répartition des classes candidates en catégories.....	164
7.1.2. Elaboration des diagrammes de classes préliminaires par catégorie.....	165
7.1.3. Dépendance entre catégories.....	167
7.2. Développement du modèle statique.....	168
7.3. Développement du modèle dynamique.....	169
7.3.1. Diagrammes de séquences.....	169
7.3.1. 1. Diagramme de séquence du cas «Ajouter régime»	170
7.3.1. 2. Diagramme de séquence du cas «Modifier régime»	172
7.3.1. 3. Diagramme de séquence du cas «Supprimer régime»	174
7.3.1.3. Diagramme de séquence du cas «Invoquer régime»	175
7.3.2. Diagramme état de transition.....	176
Conclusion.....	176
Introduction.....	177
8.1. Développement du modèle du déploiement.....	177
8.2. Dictionnaire des données.....	180
8.2.2. L'algorithme propose pour l'extraire d'un régime.....	185
8.3. Le modèle relationnel.....	187
8.3.1. Les règle de passage.....	188
8.3.2. Les règles de gestion.....	188
8.3.3. Les tables de la base des données.....	189
Conclusion.....	192
Introduction	193

9.1. Présentation des outils de développement de l'application.....	193
9.1.1. Le langage de programmation java.....	193
9.1.2. Eclipse	193
9.1.3. Implémentation de la base de données	194
9.1.4. Le serveur.....	194
9.2.1. Le PHP.....	195
9.2.2. My SQL.....	195
9.2.3. L'interface My SQL /PHP.....	196
9.2.4. L'interface PHP my admin.....	196
9.3. Quelques interfaces de l'application.....	197
Conclusion.....	200

Liste des tableaux

Chapitre1

Tableau 1. Composition chimique d'un homme de 72 kg.....	9
Tableau 2. Composition chimique de quelques aliments courants.....	9
Tableau 3. Rôle des aliments ingérés chez l'homme.....	11
Tableau 4. Groupes d'aliments.....	12
Tableau 5. Besoin en aliments énergétiques.....	14
Tableau 6. Besoin en aliments non énergétiques.....	15

Chapitre2

Tableau 1. Historique des services web.....	40
Tableau 2. Avantages / Inconvénients.....	52

Chapitre4

Tableau 1. Modélisation de contexte de notre système	83
Tableau 2. Gérer le rôle des acteurs.....	86

Chapitre5

Tableau 1. Liste des acteurs et des messages par cas d'utilisation de notre système.....	89
--	----

Introduction générale

Actuellement, le monde connaît une avance technologique considérable dans tous les secteurs et cela grâce à l'informatique qui est une science qui étudie les techniques du traitement automatique de l'information. Elle joue un rôle important dans le développement de l'entreprise et d'autres établissements.

L'avènement de nouveaux usages de l'informatique dite mobile et ambiante ne permet plus de concevoir des applications logicielles dédiées à des plates-formes prédéfinies, standardisées, composées d'un ensemble de dispositifs qui sont a priori connus. Les logiciels nécessitent toujours plus de capacité d'adaptation face à une multitude de contextes d'utilisation.

Que dire alors de l'enjeu proposé par une informatique qui se voudrait d'adapter dynamiquement une application logicielle à un environnement d'exécution découvert dynamiquement, évoluant tout aussi dynamiquement, et partiellement connue a priori. Le paradigme qui permet de gérer une telle application par assemblage de composants se révèle alors particulièrement pertinent lorsqu'il est associé à un ensemble de composants orientés services pour la découverte dynamique de dispositifs.

La notion de SOA (Architecture Orientée Services) définit un modèle d'interactions au niveau logiciel mettant en œuvre des connexions entre des composants logiciels «fournisseurs» à l'attention de composants logiciels «consommateurs». Cette approche a largement été adoptée pour le développement de systèmes de traitement de l'information favorisant par là même, la distribution de fonctionnalités indépendantes, leur réutilisabilité et leur intégration.

Les services Web sont des technologies émergentes permettant une interopérabilité entre les différents acteurs (fournisseurs et demandeurs de services) du fait de leur architecture reposant sur des technologies standard. Ils ont pour vocation de favoriser une architecture orientée services, intégrant des systèmes hétérogènes complexes, fortement distribués et pouvant coopérer sans recourir à une intégration spécifique et coûteuse l'architecture à services web apparaît pour permettre à des applications hétérogènes s'exécutant au sein de différentes

entreprises de pouvoir interopérer à travers des services offerts par les applications. L'hétérogénéité des applications n'est pas seulement considérée au niveau des langages d'implémentation des applications, mais aussi au niveau des modèles d'interaction, protocoles de communication et niveaux de qualité des services.

1. Motivation

Le but de cette étude est de concevoir une application basée sur des web services, pour les régimes alimentaires. Nous souhaitons réaliser une application qui répond aux besoins des demandeurs à savoir leurs états de santé, leurs religions, leurs régions, ..., d'un côté et de l'autre pour aider les nutritionnistes à prescrire ces régimes. Pour atteindre ce but nous proposons de concevoir une application pour la gestion des « régimes alimentaires » au sein d'une application client-serveur qui présente le fournisseur des services web (dans notre cas le régime alimentaire), puis nous réaliserons un site web pour les régimes alimentaires qui est le consommateur des services fournis par la première application. Donc ce système va offrir la possibilité de la création puis la publication et enfin l'invocation d'un régime alimentaire.

Notre projet de fin d'étude sera réalisé après une recherche sur le domaine des régimes alimentaires.

2. Problématique

Le régime alimentaire représente une importance majeure pour la sauvegarde et la prévention de la santé de l'homme. La relation entre notre santé et les aliments que nous mangions est devenue une évidence non discutable.

Actuellement, la prescription d'un régime alimentaire nécessaire pour une personne, qu'elle soit malade ou non, peut se faire par une consultation directe d'un nutritionniste ou un médecin qui proposent des régimes très étudiés et plus confiants ; mais ce moyen est non disponible à tout moment.

Il existe une autre alternative plus facile et très rapide pour obtenir un régime alimentaire et qui consiste à faire une recherche sur internet et choisir une recette parmi plusieurs qui sont trouvées. L'inconvénient de cette source réside dans l'assurance, la confiance et la justesse de l'information. Le régime alimentaire est fortement lié à des critères dynamiques et variantes assez difficiles à gérer à cause

de la complexité due à la cohabitation de plusieurs problèmes au niveau de chaque régime dont nous citons les suivants :

- Difficulté de donner un régime acceptable sans toucher à l'un de ses critères, ou d'éliminer un critère par rapport à l'autre.
- Volume important des informations traitées.
- L'ambiguïté et la variance dans les règles suivies pour aboutir à un régime alimentaire.

Au vu des considérations citées ci-dessus, l'automatisation de la tâche en utilisant les services web s'avère une approche prometteuse pour apporter une valeur ajoutée au domaine.

3. Organisation de travail

Le document est structuré en deux parties essentielles. La première partie présente le domaine d'étude qui est en fait une synthèse de la documentation faite autour du domaine d'étude et qui contient des définitions et des concepts fondamentaux liés au régime alimentaire. Après En définit la notion de base de web services Enfin nous expliquons le processus de développement que nous allons adopter pour aboutir à notre système.

La deuxième partie du mémoire est consacré à l'étude de cas. Elle constitue l'essentiel du travail d'ingénierie des systèmes d'information que nous avons effectué.

Elle s'articule autour des phases essentielles de la méthode 2TUP, et qui sont :

- **L'étude préliminaire.**

Dans cette phase, nous allons élaborons le cahier des charges qui contient les activités plus formelle de capture des besoins fonctionnels et de capture des besoins techniques.

- **La capture des besoins fonctionnels.**

Dans cette phase, nous complétons le recueil des besoins ébauchés lors de l'étude préliminaire. Cette étape représente la branche gauche du cycle en Y puisqu'elle décrit les différentes fonctionnalités du système et la façon de les utiliser.

- **La capture des besoins techniques.**

Dans cette phase nous citons les spécifications techniques du système (l'architecture physique ainsi que les cas d'utilisation techniques).

- **La phase d'analyse.**

Dans cette phase, nous découpons notre système en paquetages et définissons les classes d'analyse du système, le modèle statique et le modèle dynamique.

- **La conception préliminaire.**

Dans cette phase, nous fusionnons les résultats des études fonctionnelles et techniques pour présenter le modèle de conception

- **La conception détaillée.**

Dans cette phase, nous allons présenter la conception des classes, des associations qui construiront notre système.

- **Le dossier technique**

Dans ce chapitre, nous allons présenter les outils de développement que nous avons utilisé et expose quelques interfaces de l'application.

Résumé

La troisième génération du Web est orientée service. Cette orientation favorise l'interopérabilité des applications et des systèmes. Les services Web sont des technologies émergentes et prometteuses pour le développement, le déploiement et l'intégration d'applications Internet. Ces technologies, basées sur XML, fournissent une infrastructure pour décrire (WSDL), découvrir (UDDI), invoquer des services (SOAP).

Le présent mémoire synthétise le travail effectué dans le cadre de recherche dans le domaine des régimes alimentaires. Malgré l'avancée technologique spectaculaire qu'a connue l'informatique mais il reste un manque dans l'automatisation de certaines tâches comme les régimes alimentaires

Ainsi nous avons permis en place un système d'information automatisé pour la gestion des régimes alimentaires et d'autre part un système à base des services web pour l'utilisation des régimes fournis par la première application. Et cela quel que soit l'état de santé de la personne concernée.

Nous avons adopté pour cela la méthode 2TUP comme une méthode de conception, UML comme langage de modélisation, JAVA comme langage de programmation et le SGBD MySQL, Oracle pour l'implémentation de la base de données.

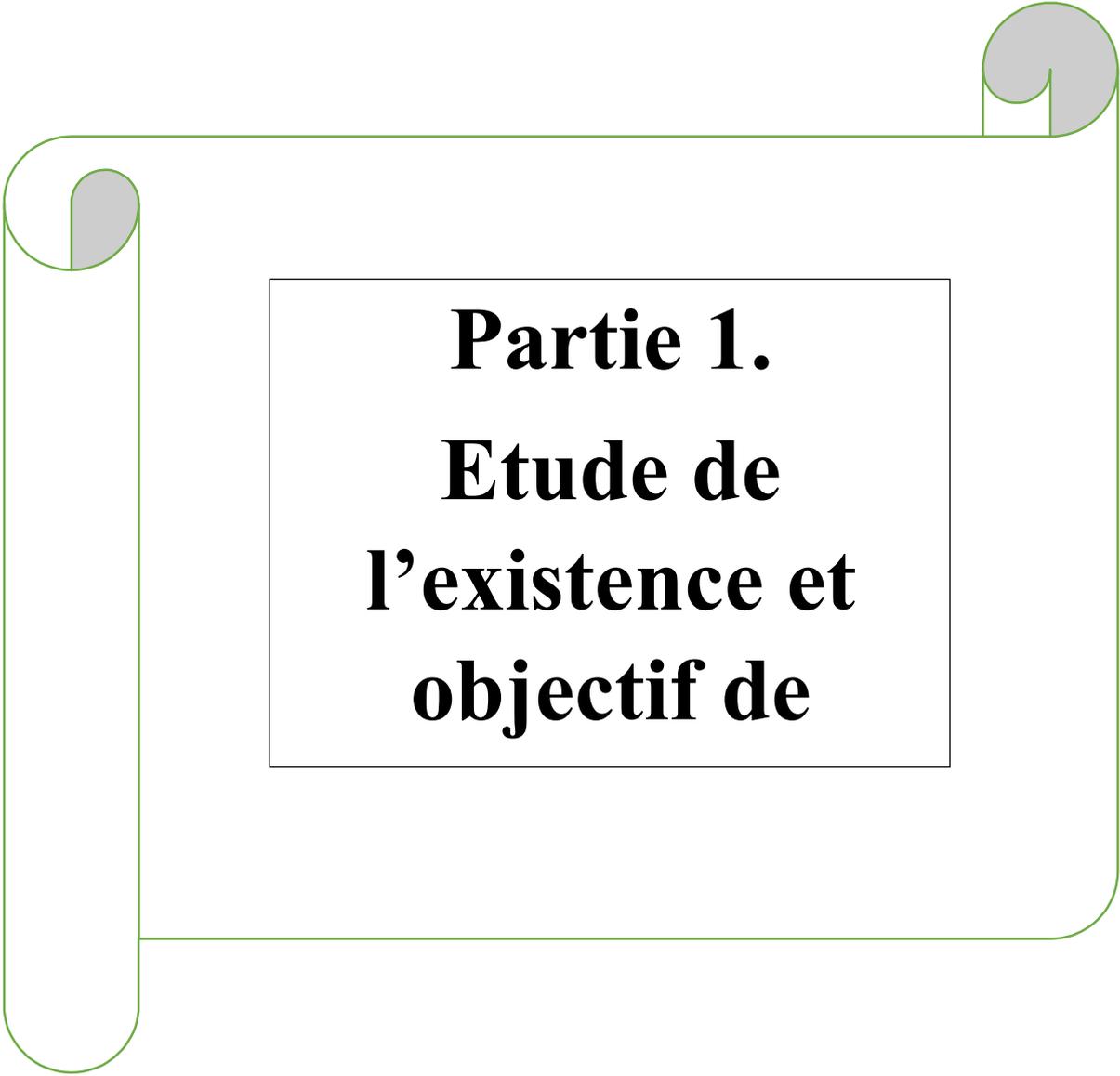
Mots clés : Services web, Orienté Objet, UML, JAVA, MySQL, Oracle, XML, UDDI, SOAP.

Abstract

The third generation of the Web is oriented service. This orientation promotes the interoperability of applications and systems. Web services are emerging and promising technologies for the development, deployment and integration of Internet applications. These technologies, based on XML, provide an infrastructure to describe (WSDL), find (UDDI), and invoke services (SOAP). This memo summarizes the work done in the research framework in the field of diets.

Despite the dramatic technological advances experienced by the computer but there is a lack in the automation of certain tasks such as diets. So we allow up an automated information system for the management of diets and also a web services-based system for the use of the results provided by the first application. And this regardless of the state of health of the person concerned. We adopted it for the 2TUP method as a method of design, UML as modeling language, Java as a programming language and the MySQL DBMS, Oracle for the implementation of the database.

Keywords: Web services, UML, Java, MySQL, Oracle, XML, UDDI, SOAP.

A decorative graphic of a scroll with a green outline and grey shading at the top and bottom edges. The text is centered within a white rectangular box in the middle of the scroll.

Partie 1.
Etude de
l'existence et
objectif de

Introduction

La vie est un perpétuel fonctionnement, il a pour siège les organes qui, travaillant et désassimilant sans cesse, tendent sans cesse à revenir à leur type primitif.

D'où la nécessité d'un continuel apport d'aliments. Suivant sa nature et sa quantité, l'alimentation conserve normale la composition du milieu vital et la texture des tissus, ou bien elle les transforme lentement et, avec eux, modifie les actes fonctionnels.

Rien ne saurait donc être plus important que de s'alimenter régulièrement, cependant rien de plus difficile ni de plus méconnu, et sur l'une des conditions essentielles dont dépend étroitement la santé de l'individu, la prospérité de la famille, l'amélioration des constitutions et des races, on vit de traditions et de sentiments dès qu'il s'agit de l'espèce humaine.

Depuis les temps les plus lointains, l'homme s'est alimenté des fruits de la terre et de la chair des animaux. Mais à certaines époques, sous l'influence de théories philosophiques ou d'idées religieuses, quelquefois en raison de considérations hygiéniques, ou simplement forcé par la nécessité, il s'est, volontairement ou non, soumis à des régimes spéciaux tantôt ne mangeant que des fruits et des herbages, tantôt ajoutant le lait à ces végétaux, tantôt, au contraire, se nourrissant presque uniquement déviant des, ou bien adoptant un régime mixte mais dont était retranchée la chair des animaux.

Ces modes exclusifs alimentation ont donné des résultats hygiéniques ou sociaux que nous allons faire connaître. Ils trouveront d'ailleurs plus loin leurs applications dans le traitement diététique des maladies.

Cependant l'alimentation fait son œuvre irrationnelle, elle laisse chaque jour un déficit, ou bien apporte un fâcheux excès de graisses, de chair, d'eau, de sels minéraux, et de ces régimes inconsidérés, les effets s'accumulant, les plasmas nutritifs se modifient peu à peu, les cellules et les organes subissent une lente déchéance, la santé décline, la sénilité s'établit, la constitution morbide s'accroît, la maladie survient.

Il importe donc beaucoup que l'homme apprenne à se nourrir normalement et garde le plus longtemps possible sa jeunesse et sa vigueur.

S'il vient à tomber malade il faut aussi que le médecin sache lui appliquer le régime alimentaire le plus propre à ramener la santé.

1.1. Discours alimentaire selon les époques et les cultures

Toute culture est multiple, fruit de diverses provenances et ouverte à toutes les évolutions.

Cependant, leur combinaison est unique et repose sur certains caractères dominants. Une culture religieuse dominante n'a pu le devenir qu'en adoptant les valeurs dominantes dans la société où elle s'est implantée (acculturation).

Par ailleurs, toute société étant issue d'une culture religieuse spécifique, les individus partagent sensiblement les mêmes représentations, c'est-à-dire qu'ils se trouvent au moins dans un même bassin de représentations.

Ce que nous voulons montrer en parcourant les cultures et les époques, c'est que l'on retrouve aujourd'hui des représentations en matière alimentaire dont la trace est perceptible jusque dans l'Antiquité et ce, à peu près dans les mêmes régions géographiques qu'alors.

Les « styles » religieux qui se sont développés ont eux-mêmes respecté cette disposition initiale.

On peut ainsi observer des constantes socio-historiques à travers le clivage entre un tempérament par exemple « germanique », celui des Celtes et des Germains, qui se prolonge dans un monde à dominante religieuse protestante et correspondant à la sphère culturelle anglo-saxonne, et un tempérament méditerranéen, celui des Grecs et des Romains, devenant un monde à dominante catholique, toujours lié à l'ère de civilisation méditerranéenne.

Le modèle culturel catholique, par exemple, apparaît donc autant comme un modèle lié à la civilisation méditerranéenne que comme un modèle religieux.

La commensalité s'y exprime comme un plaisir partagé qui engendre un certain équilibre alimentaire, c'est-à-dire une discipline collective du « manger ensemble ».

À l'inverse, dans le modèle culturel protestant, le plaisir est subordonné à la santé.

L'analyse du rapport à l'alimentation dans la période contemporaine, fait apparaître un clivage entre deux approches antagonistes qui recourent des imaginaires religieux spécifiques: l'alimentation comme outil (santé, futur) ou comme substance (convivialité, présent).

Le temps consacré à l'alimentation est ainsi une limite au temps consacré aux autres activités dans les pays tandis qu'il peut devenir un temps à utiliser dans un autre pays.

Se nourrir apparaît alors pour une majorité des personnes comme un acte technique, reposant sur une conception fonctionnelle de l'alimentation. L'acte alimentaire n'est donc pas valorisé en tant que tel.

Il n'y a évidemment pas un seul type de prescriptions par époque et/ou par région. Chacune d'entre elles est traversée par plusieurs agents prescripteurs (par exemple à l'église et dans sa famille) et plusieurs messages (par exemple religieux et médical) tantôt complémentaires, tantôt contradictoires.

Mais chacune d'entre elles est aussi marquée par des tendances dominantes, à la fois héritées de leur fond culturel propre et définissant une trajectoire culturelle en cours. [1]

1.1.1. L'Antiquité

Caractérisé par :

Les techniques de séchage, de fumage et de salage, de même que la fabrication du pain, du fromage et des boissons fermentées remontent au Néolithique.

La prescription culturelle dominante dit que l'être civilisé doit observer tel régime (et donc que ce régime distingue le « civilisé » du « barbare »).

La viande, pour le « civilisé », est d'abord un objet de sacrifice, un moyen d'offrande aux dieux (Grecs), et elle est liée aux repas de fêtes (Romains). Elle ne constitue pas un aliment de base du quotidien. De même, le cru appartient au régime de la sauvagerie (viande, lait) tandis que le cuit est un signe de civilisation (pain).

Cependant, ce qui différencie surtout le « civilisé » du « barbare » n'est pas la consommation de viande, car on en trouve dans les banquets romains, mais le fait que l'alimentation soit variée alors que la viande est l'aliment principal du « barbare » (consommée seule, elle est signe de barbarie - avec le lait, qui est l'aliment principal de l'enfant — donc de l'être encore non civilisé).

Ce qui différencie le « civilisé » du « barbare » est aussi que les aliments sont transformés (donc domestiqués, « civilisés » par lui) ou « inventés » par l'Homme (le pain, le vin, l'agriculture) ; et pour certains consommés rituellement lors de cérémonies (convivium et symposium en Grèce). Les viandes, comme le lait, sont des aliments sauvages, car non transformés. [1]

1.1.2. Le haut Moyen-Âge

Celui qui veut être fort et vigoureux doit observer tel régime. [1]

Manger, au Moyen Age comme aujourd'hui est un acte à la fois biologique, social et culturel. On ne mange pas pareil selon qu'on est paysan ou seigneur.

Caractérisé par :

Les viandes : La noblesse était très grande consommatrice de viandes, contrairement aux couches plus populaires.

Les poissons : Les poissons séchés, fumés ou salés étaient donc très présents dans la cuisine médiévale.

Les fruits et légumes Les paysans mangeaient moins de viande que les nobles, mais davantage de fruits et légumes. Les légumes étaient de trois sortes : les racines (carottes, navets, betteraves, salsifis), les féculents (pois, haricots, fèves), et les légumes verts (choux, salades, bettes, cardons, épinards).

Les céréales et le pain : Les céréales sont très consommées au Moyen Age, notamment sous forme de pain et de bouillie.

Les épices : De nombreuses épices étaient utilisées dans les plats médiévaux. La gastronomie médiévale témoigne d'une recherche harmonieuse des épices....

Grosso modo on trouve,

- **Les paysans** mangeaient surtout du pain.
- **Les Seigneurs** mangeaient plus d'aliments différents et beaucoup de viandes. On les cuit à la broche. La viande la plus répandue est le porc. Mais on mange aussi les produits de la chasse. On aime aussi beaucoup les épices.
- **Les moines** Pour compenser l'absence de viande, la cuisine des moines utilise des œufs et des poissons. [2]

1.1.3. Du bas Moyen-Âge au XVIIème siècle

C'est une période de retrait des famines qui autorise l'apparition de préoccupations diététiques dans les classes nobles.

Le XIII^{ème} siècle offre les premiers exemples de livres de cuisine en Europe depuis la fin de l'époque romaine.

En 1256, Maître Aldebrandin de Sienna rédige « Le régime du corps », premier ouvrage de diététique en langue française.

Au XV^{ème} siècle, on trouve de nombreux livres de recettes et de prescriptions de bonne digestion, comme par exemple l'assaisonnement pour acclimater la viande à son estomac, manifestation d'un intérêt renouvelé pour le plaisir de la nourriture. [1]

1.1.4. XVII^{ème} et XVIII^{ème} siècles

Celui qui veut être distingué doit observer tel régime.

Au Moyen-Âge et jusqu'au début du XVII^{ème} siècle, « l'alimentation des élites suivait d'assez près les prescriptions des médecins » en matière de diététique : dans le choix des aliments, dans les manières de les cuire, dans l'assaisonnement, et dans l'ordre des plats. À partir de là, le souci du goût prime sur celui de la diététique, jugée dépassée.

Au « manger beaucoup » (recherche de force, d'opulence) succède le « bienmanger » (recherche d'élégance, de distinction), « ritualité conviviale fondée sur l'élégance »

La viande, par exemple, n'est plus découpée sur la table mais dans les cuisines. Si « la qualité, c'est le pouvoir », il s'agit de manifester son statut social.

On voit ici affirmée une correspondance entre la qualité de la nourriture et la qualité de la personne (être alimenté selon sa naissance).

Les banquets se caractérisent par « la beauté de la table, des nappes et de la vaisselle ; la bonne compagnie et les conversations plaisantes ; la musique, les spectacles et le raffinement des manières ». [1]

1.1.5. Le XIX^{ème} siècle

La double révolution, agricole et industrielle, qui s'est produite en Angleterre au XVIII^{ème} siècle a jeté les bases de l'agro-industrie, et modifié radicalement les conditions de l'agriculture et de l'alimentation.

Les industries alimentaires vont d'abord se substituer à l'agriculture dans la transformation des produits agricoles (par exemple : substitution du beurre industriel au beurre fermier) puis remplacer le travail domestique alimentaire. [1]

1.1.6. XXème ET XXIème siècle**Évolutions des relations entre l'alimentation et la santé.**

Au sortir de la première guerre mondiale, les maladies infectieuses et l'alcoolisme provoquent beaucoup d'absentéisme au travail et représentent, pour un pays en plein essor industriel, un problème économique majeur.

C'est donc pour « assurer les conditions de travail optimales en termes de rendement que l'on s'est intéressé à la santé des classes laborieuses »

C'est à cette période que les sciences de l'alimentation et de la nutrition sont apparues et avec elles, l'évolution de la pensée quant au rôle de l'état dans ce domaine.

Avec l'arrivée de la 2^{de} guerre mondiale, des problèmes d'alimentation et de ravitaillement se posent. [1]

1.2. Nutrition et alimentation

La nutrition est un sujet traditionnel de l'éducation pour la santé. Depuis toujours on se préoccupe de la façon dont on s'alimente à tout âge de la vie, de façon naturelle ou artificielle dès la naissance jusqu'aux questions de dénutrition des personnes âgées.

1.2.1. Définitions**➤ Nutriment**

Également qualifiés d'éléments nutritifs, les nutriments sont des molécules issues de l'alimentation et majoritairement fabriquées par le processus de la digestion. [3]

➤ Nutrition

La nutrition est l'ensemble des processus d'absorption et d'utilisation des aliments, indispensable à l'organisme pour assurer son entretien et ses besoins en énergie. [4]

Remarque

Elle ne se limite pas à la prise alimentaire. C'est également le devenir des aliments ingérés jusqu'à l'utilisation qui en est faite dans la cellule et au rejet des déchets. [4]

➤ Aliment

Un aliment est une substance ingérée nécessaire au développement, au fonctionnement, à la constitution des réserves et à la réparation des cellules. [4]

Exemple, le pain, le poisson, le riz, les tomates, le fromage, ...

L'aliment est donc ce que l'on avale lorsque l'on mange.

➤ Alimentation

Action de fournir à un être vivant ou de se procurer à soi-même les éléments nécessaires à la croissance, à la conservation.

1.2.2. Nature et rôle des constituants alimentaires**1.2.2.1. Les aliments organiques**

Les aliments organiques contiennent plusieurs atomes de carbone. Ils sont synthétisés par les êtres vivants.

Les aliments organiques sont classés en quatre catégories : glucides, lipides, protides et vitamines. [4]

a. Les glucides ou hydrates de carbone

1. Les glucides sont composés de carbone (C), d'hydrogène (H) et d'oxygène(O).
2. Les glucides constituent une source d'énergie (1 g de glucose = 16,72 kJ ou 4 kcal).
3. Les glucides sont appelés des sucres. Ils ont un goût sucré ou non.
4. Les sucres rapides sont rapidement absorbés au niveau de l'intestin grêle car ils ne sont pas hydrolysés.
5. Les sucres lents (polyosides) sont de grosses molécules. Leur hydrolyse est longue. Ils sont absorbés plus tardivement. [4]

b. Les lipides

1. Les lipides sont composés de carbone (C), d'hydrogène (H) et d'oxygène (O).
2. Les acides gras (obtenus après digestion) se classent en trois catégories :
3. Les acides gras saturés : ils ne peuvent s'unir à une autre structure ;
4. Les acides gras mono-saturés ou polyinsaturés : ils peuvent s'unir à une autre structure.
5. Les lipides ont deux fonctions principales :

Ils constituent une source d'énergie (1 g de lipide = 37,62 kJ ou 9kcal) ;

Ils transportent tous les éléments liposolubles et se retrouvent dans la composition de la cellule. [4]

c. Les protides

1. Les sont protides composés de carbone (C), d'hydrogène (H), d'oxygène(O) et d'azote (N). Ce sont des chaînes plus ou moins longues d'acides aminés.
2. Les protides ont un rôle essentiel dans l'organisme :
3. Elles participent à l'édification des tissus ;
4. Elles sont une source d'énergie (1 g = 16,72 kJ ou 4 kcal). [Ref1.4page 2]

d. Les vitamines

Les vitamines se classent en deux catégories :

1. Les vitamines hydrosolubles : B1, B2, B5, B6, B12, acide folique, C, PP ;
2. Les vitamines liposolubles : A, D, E, K.

Les vitamines sont des régulateurs de réactions enzymatiques ; ce sont des catalyseurs. L'être humain ne peut les synthétiser (l'alimentation doit les lui apporter). [4]

1.2.2.2. Les aliments minéraux

Les aliments minéraux sont des substances non synthétisées par les êtres vivants mais dont ils ont besoin.

a. L'eau

1. Environ 70% du corps humain est constitué d'eau.
2. Les pertes sont variables en fonction de la température extérieure et de l'activité physique de l'individu. Elles atteignent 2,5 litres par jour (urine, sueur, vapeur d'eau expirée). L'apport se fait par les aliments et sur tout par les boissons. [4]

b. Les minéraux

1. Les différents minéraux du corps humain se présentent sous forme de sels (Na, Ca), sous forme d'anions (S, Cl) et de cations (Fe, K).
2. Ils sont indispensables à l'organisme car ils participent au métabolisme.
3. Ils participent entre autre à la formation du squelette (phosphate de calcium), à la molécule d'hémoglobine (fer). Ils interviennent, en tant que catalyseurs, dans les processus biologiques (iode indispensable au fonctionnement de la thyroïde).
4. Les besoins sont largement couverts par une alimentation variée. [4]

Exemple : Composition chimique d'un homme de 72 kg [4]

Constituants de l'organisme	Masse en kg	Pourcentage
Eau	46,2	66
Protides	11,1	15,9
Lipides	9	12,8
Glucides	0,2	0,3
Sels minéraux, vitamines	3,5	5

Tableau 1. Composition chimique d'un homme de 72 kg

1.2.2.3. Aliments simples, aliments composés

Les aliments simples sont des aliments constitués d'une des trois catégories d'aliments organiques définis chimiquement :

- Soit un glucide (exemple : le saccharose ou « sucre ») ;
- Soit un lipide (exemple : l'huile) ;
- Soit un protide (exemple : blanc d'œuf). [4]

Les aliments composés sont des aliments constitués de deux ou trois catégories d'aliments simples. La plupart des aliments sont composés (exemple : la farine, les œufs...).

Exemple Composition chimique de quelques aliments courants [4]

	Lait	Œufs	Pain	Viande maigre	Pommes de terre	Riz
Eau	87%	74%	42%	72%	75%	12,3%
Protides	4	13	6	21	4	8
Lipides	4	12	1	6	0	0,3
Glucides	4	traces	50	traces	21	79
Sels minéraux	1	1	1	1	0	0,4

Tableau 2. Composition chimique de quelques aliments courants

1.2.2.4. Aliments complets, aliments incomplets

Les aliments complets sont des aliments dans lesquels toutes les catégories sont présentes. Par exemple, le lait est composé de :

- Protides (le caillot ou caséine utilisé pour faire les fromages, la peau ou frangipane apparaissant à la chaleur) ;
- Lipides (la crème utilisée pour faire le beurre) ;
- Glucides (le lactose) ;
- Vitamines ;
- Eau et sels minéraux. [4]

1.2.2.5. Rôle des aliments ingérés chez l'homme [4]

Alliments	Rôle énergétique	Rôle plastique	Rôle fonctionnel
Glucides	Glucose. Glycogène (après simplification en oses).	Certains oses entrent dans la composition des acides nucléiques et dans la construction de divers tissus.	La cellulose (les « fibres ») permet le transit intestinal.
Protides		Les acides aminés entrent dans la composition des tissus. Huit acides aminés ne sont pas synthétisés par l'organisme humain mais lui sont essentiels, ce sont les « A.A.E. ».	Certaines protéines entrent dans la composition des hormones, des enzymes, ...
Lipides	Après conversion en oses par l'organisme.	Des lipides entrent dans la construction des membranes cellulaires. Trois acides gras sont essentiels pour l'homme, ce sont les « A.G.E. ».	Les lipides sont les supports de certaines vitamines.
Eau		Elle imbibe les cellules et les espaces intercellulaires.	Réactions enzymatiques, équilibre hydrique, ...
Sels minéraux		Certains entrent dans la composition des tissus.	Equilibre ionique. Certains entrent dans la composition des enzymes.
Vitamines			Elles permettent le fonctionnement de l'organisme.

Tableau 3. Rôle des aliments ingérés chez l'homme

1.2.2.6. Les aliments

Les aliments sont classés en 6 grands groupes. Les aliments d'un même groupe possèdent des caractéristiques communes, ce qui permet de les remplacer l'un par l'autre et donc de varier les menus. [4]

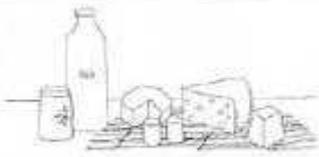
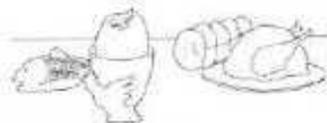
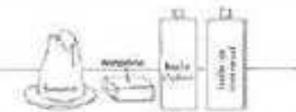
Différents groupes d'aliments	Principaux éléments énergétiques	Principaux éléments non énergétiques
 <p>Lait et produits laitiers</p>	<ul style="list-style-type: none"> - Protéines animales - Lipides animaux 	<ul style="list-style-type: none"> - Vitamines A, B₁₂, D - Minéraux (Calcium)
 <p>Viande, poisson et œuf</p>	<ul style="list-style-type: none"> - Protéines animales - Lipides animaux 	<ul style="list-style-type: none"> - Vitamines A, D, B₁₂ - Phosphore (poisson)
 <p>Féculents</p>	<ul style="list-style-type: none"> - Glucides complexes - Protéines végétales 	<ul style="list-style-type: none"> - Vitamine B - Fibres - Minéraux
 <p>Végétaux frais: fruits et légumes verts</p>	<ul style="list-style-type: none"> - Glucides complexes - Protéines végétales - Lipides végétaux 	<ul style="list-style-type: none"> - Vitamine C, A - Fibres - Minéraux - Eau
 <p>Corps gras (d'origine animale et végétale)</p>	<ul style="list-style-type: none"> - Lipides animaux - Lipides végétaux 	<ul style="list-style-type: none"> - Vitamines A, D, E
 <p>Produits sucrés</p>	<ul style="list-style-type: none"> - Glucides simples 	<ul style="list-style-type: none"> - Minéraux

Tableau 4. Groupes d'aliments

1.2.2.7. Besoins nutritionnels et apports recommandés***a. Les besoins nutritionnels***

Le fonctionnement de l'individu entraîne des dépenses ou des pertes :

- Pertes en eau et éléments minéraux dans la sueur et les urines ;
- Pertes en composés azotés par la desquamation de la peau, la chute des cheveux, les sécrétions digestives ;
- Pertes d'énergie liées à la chaleur et aux mouvements.

Pour vivre, les cellules de l'organisme ont deux types de besoins :

b. *Des besoins énergétiques* que l'on peut comparer à des besoins décarburants, qui vont être assurés par les nutriments énergétiques que sont les glucides, lipides, protides

c. *Des besoins structuraux et fonctionnels* que l'on peut comparer à des besoins de réparation, de fabrication, de remplacement, de fonctionnement des cellules, ces besoins vont être assurés :

Par des nutriments non énergétiques : eau, sels minéraux, vitamines ;

Par un nutriment énergétique, élément de construction par excellence : les protides. [4]

Les deux figures suivantes montrent les besoins en aliments énergétiques et non énergétiques

Les aliments énergétiques		
Aliments	Sources	Rôle
<u>Protéines</u>	<ul style="list-style-type: none"> - <i>Protéines animales</i> : <ul style="list-style-type: none"> • lait et produits laitiers • viandes, poissons, œufs - <i>Protéines végétales</i> : <ul style="list-style-type: none"> • céréales • féculents 	<ul style="list-style-type: none"> - Fabrication des tissus du corps - Croissance (apport d'acides aminés) - Entretien, réparation
<u>Glucides</u>	<ul style="list-style-type: none"> - <i>Glucides complexes</i> : <ul style="list-style-type: none"> • sucre des céréales (amidon) - <i>Glucides simples</i> : <ul style="list-style-type: none"> • sucre du lait : le lactose • sucre des fruits : le fructose • sucre ordinaire : le saccharose 	Source d'énergie pour le fonctionnement du cerveau, des muscles, ...
<u>Lipides</u>	<ul style="list-style-type: none"> - <i>Lipides animaux</i> : <ul style="list-style-type: none"> • lait et produits laitiers • viandes, poissons, œufs - <i>Lipides végétaux</i> : <ul style="list-style-type: none"> • huiles végétales 	<ul style="list-style-type: none"> - Energie - Croissance osseuse - Absorption des vitamines liposolubles - Croissance cérébrale (apport d'acides linoléique et linolénique)

Tableau 5. Besoin en aliments énergétiques

Les aliments non énergétiques		
Éléments	Sources	Rôle
<u>Seis minéraux et oligo-éléments</u> - calcium - fer	- lait, produits laitiers - foie, viandes, légumes verts	- anti-rachitique - anti-anémique
<u>Vitamines</u> - vitamines liposolubles <ul style="list-style-type: none"> • vitamine A • vitamine D • vitamines E, K - vitamines hydrosolubles <ul style="list-style-type: none"> • vitamine C • vitamine B (B1, B2, B6, B12) 	- lait, carottes, légumes verts - poisson - foie, œuf, huiles végétales - fruits acides, crudités - céréales, lait, poisson, foie, œuf	- vision, croissance - anti-rachitique - anti-oxydant, anti-hémorragique - anti-infectieux - croissance physique et cérébrale
<u>Eau</u> L'apport d'eau est très important	- eau de boisson - eau des aliments	- réaction d'hydrolyse (digestion) - transport des aliments et évacuation des déchets - apport de minéraux

Tableau 6. Besoin en aliments non énergétiques

1.2.3. Elaboration des rations alimentaires

La ration alimentaire est la quantité d'aliments indispensable, quantitativement et qualitativement, aux besoins de l'organisme pendant 24 Heures. Elle couvre la totalité des dépenses de l'organisme et assure son équilibre.

Une alimentation équilibrée pour couvrir les besoins d'un individu est établie sur une journée. Elle doit être adaptée à chaque individu en fonction de son âge, son activité, son état physiologique. [4]

La ration alimentaire fluctue selon l'âge, de l'activité physique, de la taille...

Elle est :

- ✓ Pour un enfant : à peu près 1600 kcal
- ✓ Pour un adolescent/une adolescente : de 2900 kcal/ de 2400 kcal

Pour un homme (avec un poids moyen de 70 kg et une activité physique modérée) : de 2800 kcal

✓ Pour une femme (avec un poids moyen de 55 kg et une activité physique modérée) : de 2200 kcal

- ✓ Pour une femme enceinte : de 2550 kcal

✓ Pour une personne avec une activité physique intense : de 300 à 700 kcal qui plus est que la ration normale.

1.2.4. L'alimentation équilibrée



La fonction première de l'alimentation est de maintenir l'organisme en bonne santé. Il est donc nécessaire d'apporter chaque jour la quantité et la qualité d'aliments dont le corps a besoin : aucun aliment n'a en lui la totalité des nutriments indispensables à notre organisme, à nous d'équilibrer notre alimentation tout en conservant le plaisir de manger.

[5]

1.2.4.1. Définition

Une alimentation équilibrée suppose un juste équilibre entre les proportions nécessaires de nutriments, de glucides, de protéines, de lipides, de minéraux, de vitamines, de fibres et d'eau. Ainsi, cela permet d'éviter la malnutrition et les problèmes liés aux insuffisances alimentaires ou, à l'inverse, aux excès alimentaires.

Une alimentation équilibrée nécessite la connaissance :

- Qualitative et quantitative des principaux constituants des aliments,

- Des besoins quantitatifs et qualitatifs des organismes.

1.2.4.2. Maîtriser quantité et qualité

La qualité de l'alimentation est donc aussi importante que la quantité. L'espérance de vie augmente d'ailleurs quand on consomme moins de calories (à condition de ne pas être en situation de carence). Mais pour lutter efficacement contre les maladies et le vieillissement, notre organisme a besoin de beaucoup de vitamines et d'antioxydants, apportés en particulier par les fruits et légumes.

Une alimentation bénéfique pour la santé est donc la conséquence d'une maîtrise (consciente ou non) à la fois de la quantité et de la qualité des aliments que nous consommons. Cette maîtrise est possible soit quand nous nous trouvons dans un environnement social qui favorise une bonne alimentation (par exemple les Crétois ou les habitants d'Okinawa), soit quand nous décidons consciemment de maîtriser notre alimentation plutôt que de la subir.

Heureusement, cette maîtrise n'est pas si complexe que l'on pourrait croire. On obtient des résultats considérables en appliquant certains principes simples. La difficulté est en premier lieu d'appliquer une certaine discipline alors que l'alimentation a un contenu social et affectif fort, qui peut entrer en conflit avec des choix raisonnés. [7]

1.2.4.3. Les cinq clés de l'équilibre alimentaire

- **La diversité** On appelle diversité alimentaire la consommation journalière d'aliments appartenant à chacune des différentes catégories d'aliments : produits céréaliers, fruits, légumes, produits laitiers, viande, poisson, œufs. Lorsque la diversité alimentaire est faible, ce sont fréquemment les fruits et les légumes qui sont oubliés.

- **La variété** On appelle variété alimentaire la consommation journalière d'aliments différents au sein d'une même catégorie.

En mangeant des aliments différents, on favorise la diversification alimentaire et donc une meilleure couverture des besoins nutritionnels.

- **La structure** La structure des repas est représentée par la succession de divers plats composant un repas traditionnel (entrée, plat de résistance composé de viande ou de poisson garnis, laitage et/ou dessert). Elle est nécessaire pour la couverture des besoins nutritionnels si cette structure assure diversité et variété alimentaires. On mange n'importe

quoi à n'importe quelle heure et en n'importe quelle quantité, ce qui aggrave les problèmes de surpoids. Il est difficile de concilier simplification et équilibre alimentaire...

➤ **La densité** On peut manger beaucoup et peu calorique, ou peu et très calorique, selon les aliments consommés. Dans le cadre des problèmes de surpoids, on peut fréquemment observer que les sujets ont une alimentation assez monotone, qui privilégie les aliments à forte densité énergétique. La mise en place d'une alimentation diversifiée et variée est donc déjà un grand pas en avant.

➤ **La fréquence** De même qu'aucun aliment n'est, en soi, mauvais pour la ligne, aucun aliment n'est mauvais, en soi, pour l'équilibre alimentaire ou la santé. Tout est question de fréquence ou de juste mesure : la consommation trop fréquente de certains aliments, ou à contrario, la consommation trop rare de certains aliments, ont toutes deux des conséquences défavorables sur la santé.

1.2.4.4. Equilibre et régime alimentaires

La notion d'équilibre réfère à la stabilité. Dans le domaine alimentaire, c'est d'abord l'équilibre énergétique qui est recherché, parallèlement à la régulation du métabolisme énergétique.

Les calories ingérées ne doivent pas dépasser l'énergie dépensée. En plus de l'équilibre énergétique, la couverture des besoins en nutriments essentiels (acides aminés, acides gras, vitamines, minéraux), est indispensable pour la survie et le maintien en bonne santé des individus.

Le terme équilibre alimentaire couvre les notions énergétique et nutritionnelle. Dans cette perspective, l'équilibre alimentaire est le résultat de la cohérence entre un régime alimentaire et le mode de vie (notamment pour les dépenses énergétiques et les défenses immunitaires) dans lequel il s'insère. [8]

1.2.4.5. Déséquilibre alimentaire et maladies nutritionnelles

Il existe des maladies par carence et des maladies par excès.

➤ **Maladies par carence** liées à une sous-alimentation, par exemple :

- le kwashiorkor, cette maladie apparaît chez les jeunes enfants après sevrage. Nourris exclusivement de féculents, les enfants présentent une carence protéique en quantité et en qualité (acides aminés essentiels absents) ;

- le marasme, cette maladie s'observe surtout pendant la première partie de la vie, elle est causée par une carence globale en aliments protéiques et énergétiques.

➤ **Maladies par excès** liées à une suralimentation : ce sont essentiellement les maladies cardiovasculaires et l'obésité. Les facteurs responsables sont multiples, le déséquilibre alimentaire joue un rôle important.

1.3. Le régime alimentaire

Une bonne alimentation repose sur des notions simples qui relèvent du bon sens.

Pour rester en bonne santé, le corps a besoin d'une certaine proportion de glucides, de lipides et de protéines mais aussi de vitamines et de minéraux.

Aucun aliment ne contient à lui seul tous les nutriments qui nous sont nécessaires. Pour l'équilibre alimentaire comme pour le plaisir de manger, il faut prendre chaque jour des aliments de chaque famille en fonction des apports conseillés. Aucun aliment n'est à proscrire complètement dans un régime alimentaire normal.

1.3.1. Ce que nous dit l'histoire des régimes

En 1825, Jean-Anthelme Brillat-Savarin publie *La physiologie du goût*. Il a interrogé plus de 500 contemporains obèses. Tous partagent une particularité qui a frappé l'auteur : ils mangent beaucoup de féculents : pain, pâtes, pommes de terre, riz. Brillat-Savarin en déduit que « les amidons et les farines que l'homme utilise comme base journalière de son alimentation » sont à l'origine de l'obésité et que le sucre aggrave les choses.

En 1844, le Dr Jean-François Dancel donne une conférence sur le surpoids et l'obésité à l'Académie des Sciences, à Paris. Cas à l'appui, Dancel montre que l'on guérit tous les obèses, « sans exception », par un régime riche en protéines comme la viande, et très pauvre en pain, pâtes, riz et sucre. « Tous les aliments riches en carbone et en hydrogène [les glucides], dit-il, ont tendance à produire de la graisse. »

En 1856, le Dr William Harvey, un médecin londonien, rapporte que « les sucres et les farineux sont utilisés pour engraisser les animaux et qu'un régime qui n'en comprend pas pourrait être utile pour arrêter la formation de graisse. » En 1862, Harvey prescrit un tel régime à William Banting, un de ses patients obèses. En 9 mois, celui-ci perd 25 kilos.

En 1862, Banting publie à son tour une « Lettre sur la corpulence » de 16 pages. Il décrit les régimes qu'il a essayés sans succès, et comment une alimentation pauvre en sucres et

féculeux lui a permis de perdre si facilement du poids. Le fascicule de Banting devient un best-seller européen.

Le régime Banting a un tel succès que même les têtes couronnées s'y mettent. En 1864, Napoléon III suit comme beaucoup de ses compatriotes le régime Banting, avec « un grand bénéfice. »

En 1869, Thomas Tanner, le célèbre médecin britannique, publie *The Practice of Medicine*, dans lequel il donne une longue liste de traitements ridicules contre l'obésité, dont la saignée, les sangsues et... les régimes sans graisses. Toutes ces méthodes, écrit-il, échouent lamentablement. Le seul traitement efficace, assure-t-il, c'est d'éviter les glucides, pain, farineux, pommes de terre, sucre.

En 1877, dans *Anna Karénine*, Léon Tolstoï se fait l'écho des régimes que suivent ses contemporains lorsqu'ils veulent maigrir. L'amant d'Anna, le comte Vronsky doit participer à une course équestre qui exige de ne pas dépasser le poids de 160 livres. Pour y parvenir, écrit Tolstoï, « il évitait les aliments farineux et les desserts. »

En 1901, le Dr William Osler, considéré comme le père de la médecine américaine moderne écrit dans *Principles and Practice of Medicine*, ouvrage de référence, que les obèses doivent pour maigrir et rester en bonne santé « réduire les farineux et les sucres. »

En 1925, le Dr Gardiner Hill (Hôpital St Thomas de Londres) décrit dans le *Lancet* les principes du régime amaigrissant dont l'efficacité est avérée : « Toutes les formes de pain contiennent une forte proportion de glucides, qui vas de 45 à 65 pour cent. Il doit donc être condamné. »

Entre 1943 et 1952, des chercheurs de Stanford, Harvard, Chicago, Cornell, publient indépendamment une série d'articles sur l'obésité et les moyens de la traiter. Tous ces régimes se ressemblent. Ils interdisent invariablement sodas, sucre, miel, sucreries, fruits au sirop, gâteaux, tartes, biscuits, pain, pommes de terre, pâtes, céréales du petit déjeuner.

En 1963, le livre de référence en Grande-Bretagne, *Human Nutrition and Dietetics*, écrit par Sir Stanley Davidson et Reginald Passmore rappelle que « la consommation d'aliments riches en glucides devrait être réduite drastiquement dans la mesure où ces aliments sont la cause la plus commune d'obésité. » À l'époque, c'est tellement évident que Passmore écrit : « N'importe quelle femme sait que les glucides font grossir : d'ailleurs tous les nutritionnistes sont d'accord avec cela. »

En 1972, parution du Régime du Dr Atkins. [9]

1.3.2. Définitions

➤ **Régime** : Un régime n'est donc qu'une ligne de conduite et ne veut rien dire en soi. Un régime politique est une ligne de conduite politique ; un régime alimentaire est une ligne de conduite alimentaire.

➤ **Le régime alimentaire** peut être défini à partir des moyens par lesquels un individu va acquérir une certaine quantité d'éléments vitaux, propres à assurer son existence et son développement psycho-physiologique de manière équilibrée.

En d'autres termes :

➤ Un régime est une modification des habitudes alimentaires. Il doit être suivi avec vigilance pour ne pas causer de carences alimentaires.

➤ Un régime alimentaire se définit d'une manière quantitative et qualitative, mais pour mieux en saisir les nuances, il ne sera pas inutile de rappeler quelques précisions sur les trois éléments de base de notre alimentation : glucides, protides et lipides. [11]

✓ **Définition quantitative**

Un régime alimentaire se définit quantitativement par le nombre total de calories et par le pourcentage de calories provenant des trois éléments fondamentaux : glucides, lipides et protides. [11]

Exemples

Aux Etats-Unis, le régime alimentaire moyen s'élève à 3 250 calories assurées pour 55,4 % par les glucides, pour 19,9 % par les lipides et 24,6 % par les protides,

En Corée, le régime alimentaire moyen atteint 1 900 calories dont 90,4 % sont des glucides, 3,7 % des lipides et 5,8 % des protides.

De ces deux exemples "extrêmes" résulte une double constatation quantitative. Le nombre total de calories d'un régime alimentaire moyen varie entre 1 800 et 3 300 calories suivant les différents pays. [11]

Mais cette définition quantitative est bien incomplète. Elle ne tient pas compte de l'extrême variété de la qualité des aliments à l'intérieur même des trois grands groupes, glucides, lipides et protides, de la proportion des vitamines, des calories initiales et des calories définitives, autant de notions qualitatives indispensables pour élaborer une

définition complète d'un régime alimentaire et pour tenter une classification suivant des critères rationnels.

✓ Définition qualitative

La définition quantitative d'un régime alimentaire doit être en effet précisée par d'autres données qualitatives : la variété des aliments à l'intérieur de chaque groupe, les calories initiales et les calories définitives, les vitamines et les sels minéraux.

Variété de la qualité des aliments à l'intérieur de chaque groupe.

A l'intérieur de chaque groupe, glucides, lipides et protéides, il peut exister des différences considérables entre les aliments absorbés. [11]

1.3.3. Les critères de classifications des régimes alimentaires

Nous pouvons distinguer plusieurs critères de classifications des régimes alimentaires selon les régions (les pays), l'état de santé de l'être humain, et nous n'oublions pas les philosophies existantes qui respectent les besoins de l'être humain (Les croyances religieuses, l'habitude de l'alimentation des gens par exemple les végétariens).

1.3.3.1. La diversité des régimes alimentaires

✓ Les régimes alimentaires sont fonction :

a. De l'âge (le nourrisson et l'adulte ne s'alimentent pas de la même manière)

b. Des ressources alimentaires de la région et du moment (on ne trouve pas les mêmes animaux ni les mêmes végétaux aux quatre coins du globe, de plus les aliments changent en fonction de l'année) **de la région pour des ressources données** (spécialités régionales, locales, familiales)

c. du climat (un Inuit mangera plus de graisse et de viande qu'un Pygmée)

d. du niveau social, des ressources économiques d'un pays, des contraintes économiques et démographiques (les serfs ne mangeaient pas les mêmes aliments que les seigneurs).

e. de l'époque (australopithèque cueilleur-charognard, sédentarisation et agriculture qui entraîna une baisse de la taille moyenne de la population, de la poule au pot d'Henri IV au fast-food américain...)

f. de l'avancée des connaissances et de la technologie (conservation des aliments dans le froid [neige, glace, réfrigérateur, congélateur ; lyophilisation, cryogénéisation ; repas artificiel, sang de reconstitution] et par le fumage et le salage) (thé dans les pays chauds car cela permet de stériliser l'eau de consommation et de la libérer des germes qu'elle contient généralement, germes source de maladies)

g. de la religion (on ne mange pas de porc (mauvaise conservation) dans les pays musulmans, de vache en Inde) et de la société (on ne mange pas de cheval dans les pays anglo-saxons (rarement de lapin), on mange du chien au Viêt-nam)

h. des goûts (selon l'odeur, le goût, l'aspect, l'appétence des aliments est différente) [12]

- **Les voyages** ont permis le commerce des épices et autres denrées exotiques et ont contribué aux modifications des habitudes alimentaires et à l'uniformisation des régimes alimentaires.

1.3.4. Le suivre d'un régime

Un régime peut-être prescrit par un médecin comme traitement thérapeutique ou préventif (ex. : diabète, risques cardio-vasculaires) :

1. Le régime est alors étroitement encadré par des professionnels de la santé comme des nutritionnistes.

2. Le médecin étudie l'alimentation du patient pour ensuite établir un nouveau régime alimentaire avec des indications sur les aliments à consommer et à éviter.

De nombreuses personnes entament un régime sans consulter un médecin. Dans ce cas, il est important de s'informer consciencieusement sur le régime et d'écouter son corps.

En effet, la vigilance reste de mise, car un régime n'est pas anodin et peut causer une dénutrition engendrant des problèmes de santé. [10]

1.3.5. Un régime n'est pas sans risque

Les régimes alimentaires sont souvent présentés comme des remèdes miracles contre la prise de poids. Une promesse qui cache de nombreuses difficultés, voire de vrais dangers pour la santé.

Il indiquait notamment que ces régimes alimentaires « peuvent induire des déséquilibres nutritionnels et des inadéquations d'apports (insuffisance et excès), notamment en vitamines et en minéraux »

Il soulignait que « la pratique de ces régimes alimentaires peut avoir pour conséquences des perturbations somatiques, d'ordre osseux et musculaires, ainsi que des perturbations psychologiques (notamment troubles du comportement alimentaire), ou encore des modifications profondes du métabolisme énergétique et de la régulation physiologique du comportement alimentaire » [13]

1.3.6. Le bon régime

Il devrait y en avoir : ce serait, rêvons un peu, un régime idéal :

- un régime qui respecterait les besoins individuels,
- ceux liés aux besoins professionnels
- et aux besoins physiques (sport ou non, jardinage, bricolage...),
- un régime tout doux dans un monde de brutes : doucement, dans le bon rythme,
- sans épuiser la monture qui serait ménagée par l'attention qu'on lui porterait.
- On n'y perdrait pas du poids « à toute vitesse », « avant l'été », mais on aurait l'impression d'y manger, de continuer à partager avec d'autres quelque chose qui dépasserait de loin les « bêtes calories ».

1.4. La structure utilisée pour le régime alimentaire

Pour aider l'être humain à trouver son régime alimentaire d'une façon rapide et orientée vers ses besoins, et après la recherche effectuée dans ce chapitre, nous proposons une structure de régime basée sur quelques critères. Ces critères sont choisis en fonction de leurs importance dans l'état de santé de l'être humain, et qui sont : l'état de santé, l'âge, l'effort physique ainsi que la religion et la région d'habitation.

En effet, la structure qu'on va adopter dans notre application pour un régime alimentaire est la suivante :

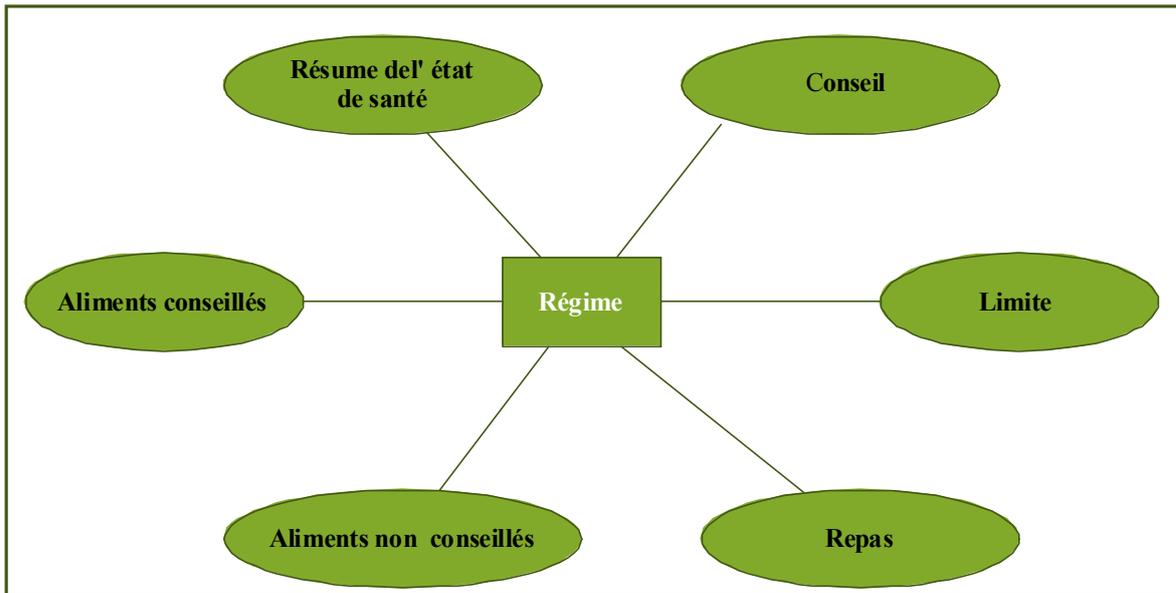


Figure 1. La structure de notre régime alimentaire

Conclusion

Nous pouvons donc conclure que les régimes alimentaires ont une place critique dans notre vie quotidienne contre dérapage de l'état de santé non seulement dans la contribution de traitement spécial des maladies, mais dans la prévention contre ces anomalies.

Introduction

La diversité des applications d'un même système d'information, et la nécessité de manipuler des quantités de plus en plus importantes de données, posent un grand problème d'intégration d'applications. En effet, la communication entre les applications et les différents systèmes d'information devient de plus en plus difficile à cause des technologies hétérogènes utilisées qui amènent les utilisateurs à travailler dans un environnement incohérent, mal adapté et incompatible. Le manque d'interopérabilité est alors relevé comme principal problème de ces systèmes.

Les architectures à base de services SOA ont introduit une nouvelle organisation des applications, basée sur le concept de services, composants logiciels auto-descriptifs, faiblement couplés et interagissant. L'une des implémentations des SOA, les services Web (Web Services) sont devenus incontournables pour la réalisation d'applications distribuées sur Internet.

Les services Web prennent leurs racines dans l'informatique distribuée et dans l'avènement du Web. La technologie des services Web a pour objectif d'uniformiser la présentation des services offerts par une entreprise et d'en rendre l'accès transparent pour tout type de plate-forme, au travers d'un certain nombre de standards d'interopérabilité. Cette notion de services Web désigne essentiellement une application mise à disposition sur Internet par un fournisseur de service, et accessible par les clients au travers de protocoles Internet standards.

2.1. L'évolution des réseaux informatiques

Avec une évolution exponentielle dans le temps, les Systèmes d'Informations et réseaux informatiques sont devenus plus complexes et plus hétérogènes en raison de la diversité des besoins, des exigences des clients, et de la grande masse d'information stockées et manipulées par les internautes. Les infrastructures publiques et privées du secteur informatique sont de plus en plus multiplateformes, multifournisseurs et distribuées à grande échelle.

Dans la suite on cite les principales étapes de l'évolution réseaux informatiques :

2.1.1. Le système central

Dans un système central, la logique de traitement se situe sur la machine serveur ou hôte, l'utilisateur interagissant avec celui-ci au travers d'un terminal PC ou passif en mode caractère généralement. [1]

Les avantages

- ✓ Assure la haute disponibilité. [1]
- ✓ Assure l'intégrité des données et offre à l'entreprise un système cohérent et fiable.

[Ref2.2 page 5]

Les inconvénients

- ✓ Coût exorbitant (Couts d'acquisition et d'exploitation sont élevés. [2])
- ✓ Système propriétaire. [2]



Figure 1. Le système central.

2.1.2 Le modèle client/serveur

Le client-serveur est une évolution du Mainframe et permet par l'utilisation de nouvelles méthodes et techniques de passer outre les limites que l'on connaissait avec l'environnement Mainframe et les systèmes propriétaires et ainsi d'améliorer l'interopérabilité, la flexibilité des systèmes. Dans une architecture client-serveur, le client est défini comme un demandeur de services auprès du serveur qui en est le fournisseur. [1]

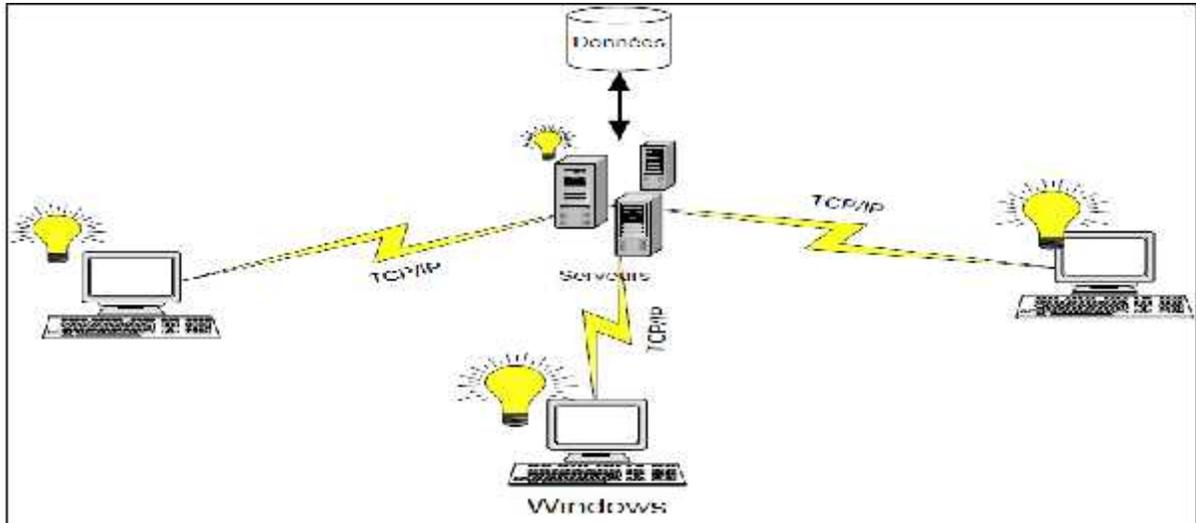


Figure 2. Le modèle client-serveur bipartite.

L'architecture client-serveur généralement utilisée aujourd'hui repose sur un modèle d'architecture distribuée bipartite. L'interface graphique se situe sur le poste client et la base de données est localisée sur le serveur.

La logique de traitement pouvant se situer sur l'une ou l'autre des parties. L'utilisateur final contrôle le poste client qui réalise une grande partie des traitements de l'application et sollicite des informations ou des traitements de la part d'un ou plusieurs serveurs. Ce type d'architecture est une bonne solution d'informatique distribuée lorsque le nombre d'utilisateurs ne dépasse pas une centaine d'utilisateurs, cependant il existe d'une part une limite tenant au fait que la connexion est maintenue en permanence entre le client et le serveur, même si aucun travail n'est effectué.

D'autre part les procédures d'accès aux données étant spécifiques aux moteurs de base de données, la flexibilité et le choix d'une base de données sont réduites. L'architecture tripartite permet de dépasser ces limites, et d'apporter une meilleure réactivité de l'entreprise en cas de changements. [1]

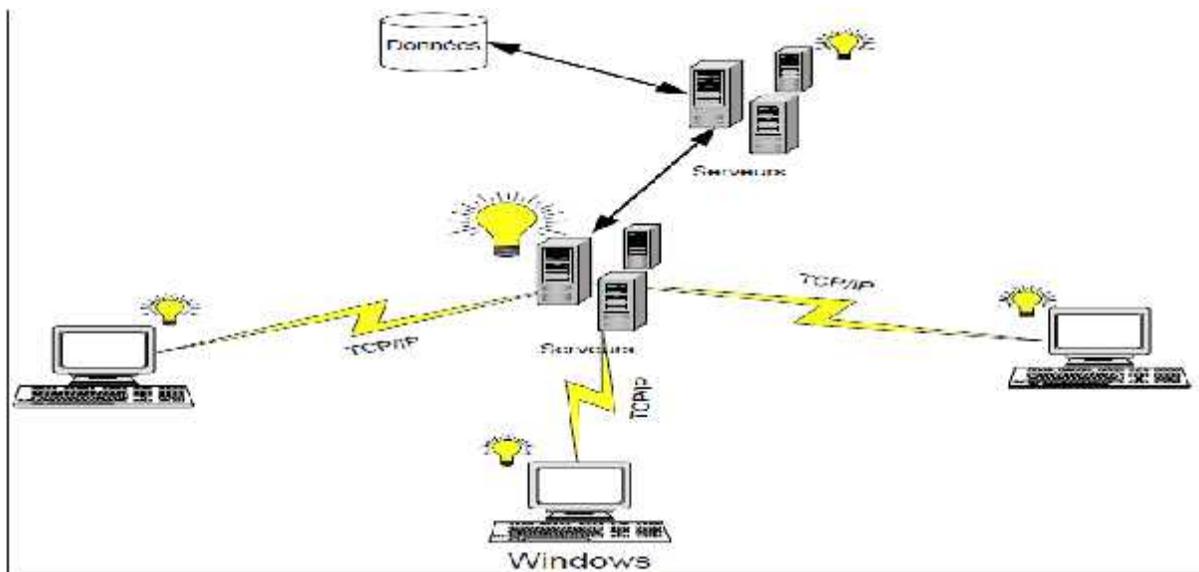


Figure 2. 1. Le modèle client-serveur tri-partite.

Dans l'architecture trois tiers ou multi-tiers, un niveau supplémentaire est ajouté entre les deux niveaux précédents, permettant de séparer les traitements de l'interface graphique et du serveur de base de données. Ce niveau intermédiaire peut être implémenté de différentes manières entre moniteur transactionnel, serveur de messages, ou serveur d'applications. Le dialogue peut se faire en mode synchrone ou en mode asynchrone, dans ce cas l'utilisateur est informé lors d'une nouvelle connexion du résultat de sa requête précédente. L'architecture tripartite supporte de la centaine d'utilisateurs à plusieurs milliers accédant à plusieurs serveurs répartis géographiquement.

La division de l'application en couches distinctes, consacrées à l'interface utilisateur graphique, à la logique de gestion (partitionnée entre plusieurs processeurs) et aux traitements sur la base de données permet de faciliter l'extension et la maintenance des applications tout en offrant un moyen d'intégration des nouvelles applications aux systèmes existants. Ce gain engendre toutefois des tâches plus complexes d'administration des composants de l'architecture (clients, serveurs et équipement réseau) ou du déploiement de l'application vers les serveurs. [1]

Les avantages

- ✓ Rapidité. [1]
- ✓ Optimisation des traitements. [1]

- ✓ Faible coût des nouvelles applications plus légères. [Ref2.2page 6]

Les inconvénients

- ✓ Choix de plateforme. [1]
- ✓ Evolutivité difficile. [1]
- ✓ Gourmand en ressources systèmes. [1]
- ✓ Duplications d'informations. [2]
- ✓ Le poste de travail deviens charge de plusieurs exécutables. [2]

2.1.3. Le modèle internet

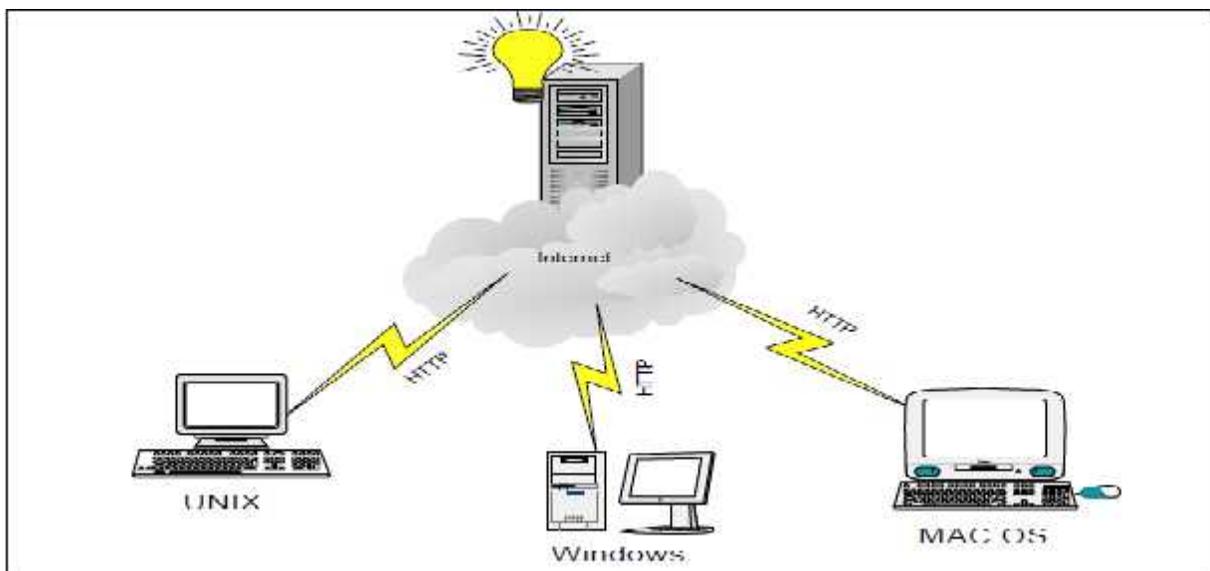


Figure 2. 2. Le modèle Internet.

L'Internet est actuellement le plus grand réseau informatique sur la planète. Il peut être considéré comme le réseau des réseaux. Il est né en 1969, lorsque le département Américain de la défense (DoD) décida d'investir des fonds pour développer un réseau expérimental permettant l'échange d'informations entre des sites éloignés de recherche et de développement et pouvant fonctionner sans interruption, même en cas de destruction partielle du réseau.

Cette technologie permet à partir d'un logiciel client appelé navigateur (ou browser) d'accéder facilement à des documents stockés sur un serveur distant connecté à l'Internet. Avec le Web, l'Internet s'ouvre au grand public et ne nécessite plus de connaissances spécifiques en informatique. Le modèle Internet est celui du client-serveur, où un programme

client permet à un utilisateur de soumettre des requêtes à un serveur Web et de visualiser le résultat, le serveur Web étant un programme qui tourne sur un ordinateur dans le but de répondre à des requêtes de logiciel client qui tournent sur d'autres ordinateurs.

Les avantages

- ✓ Multiplateforme.
- ✓ Idéale pour de la consultation.
- ✓ Facilité de mise en œuvre.
- ✓ Maintenance aisée.

Les inconvénients

- ✓ Petits fichiers.
- ✓ Petits traitements.
- ✓ Traitement local quasi inexistant.

2.1.4. Le modèle web services

C'est la synthèse actuelle entre le modèle client-serveur classique et le modèle Internet.

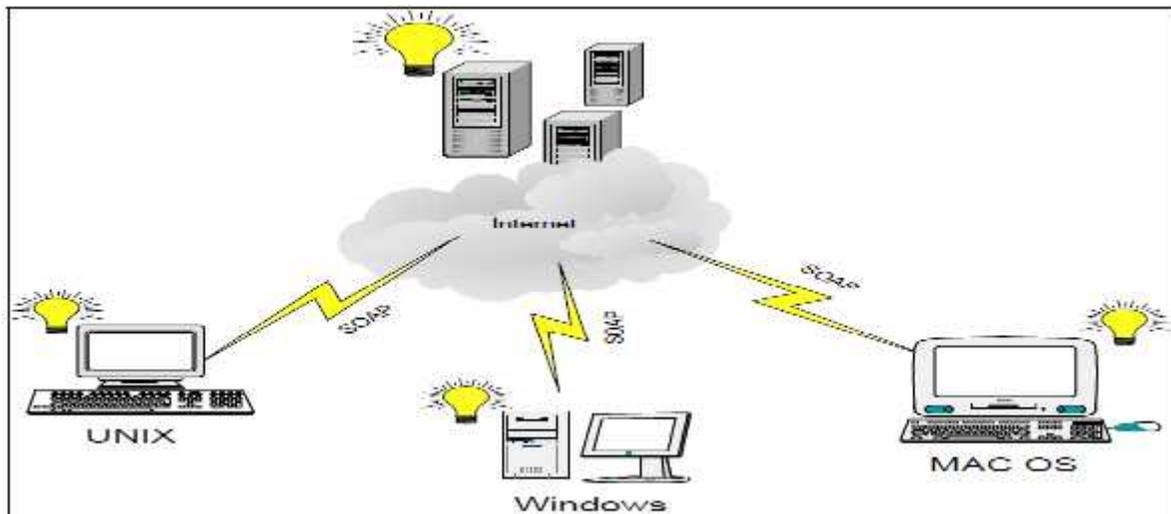


Figure 2. 3. Le modèle Web Services.

Remarque :

On peut distinguer trois phases de développement dans l'histoire du Web :

✓ **Le Web du Document**, le phénomène Internet originel, utilisé principalement par des organisations et des particuliers pour publier des informations sur leur travail, leurs produits, etc.

✓ **Le Web Applicatif**, le progrès grâce auquel les entreprises ont commencé à utiliser le Web à des fins commerciales. Les sites Internet sont devenus plus interactifs, et plus complexes, gérés par des serveurs d'applications, capable de distribuer leur charge avec d'autres serveurs, en fonction des besoins.

✓ **Le Web des Services est la phase émergente**, dans laquelle les serveurs d'application précédents communiquent désormais entre eux. [3]

2.2. L'architecture orientée service

Actuellement, sous le vocable de SOA, se développe un style d'architecture orientée service permettant de construire des systèmes informatiques évolutifs et adaptables, en améliorant leur qualité et en simplifiant leur intégration dans l'infrastructure informatique de l'entreprise, par recours à des composants réutilisables appelés services.

Dans la suite, nous présentons les concepts sur lesquels repose l'architecture SOA.

2.2.1. Service

Le terme de service est inspiré par analogie avec des services simples et réutilisables du monde réel tel que le service de distribution de billets automatique.

Au sens SOA, un service est un programme autonome, réutilisable, indépendant des langages de programmation et qui peut s'exécuter sur n'importe quelle plateforme. [4]

2.2.2. Histoire de la SOA

Le concept d'architecture orientée service, décrit la première fois par le groupe américain de recherche en technologie Gartner Group en 1996, se présente comme étant la solution adéquate à la demande actuelle des entreprises. Pour la première fois, l'architecture orientée service fut définie comme « *client/server software design approach in which an*

application consists of software services and software service consumers (also known as clients or service requesters). SOA differs from the more general client/server model in its definitive emphasis on loose coupling between software components, and in its use of separately standing interfaces » [6]

2.2.3. Définition

SOA est un style architectural qui permet de construire des solutions d'entreprises basées sur les services. Ces derniers rassemblent les grandes applications de l'entreprise (dites applications composites) en **services interopérables et réutilisables**.

Le service est un composant logiciel exécuté par un **fournisseur** (Provider) à l'attention d'un **client** (Requester). Cependant, l'interaction entre un client et un fournisseur est matérialisée grâce à un protocole responsable de l'échange des messages entre les deux entités.

L'objectif d'une architecture orientée services est donc de décomposer les fonctionnalités d'un système ou d'une application en un ensemble de fonctions basiques, appelées **services**, implémentés par des composants, et de décrire les différentes interactions possibles entre ces services.

L'objectif de cette opération est de créer une architecture logicielle globale décomposée en services correspondant aux processus métiers de l'entreprise. De ce fait, les applications d'un système d'information seront vues comme une collection de services qui interagissent et communiquent entre eux. [5]

On d'autre terme on peut dire que L'architecture SOA vise trois objectifs importants :

- ✓ Identification des composants fonctionnels.
- ✓ Définition des relations entre ces composants.
- ✓ Etablissement d'un ensemble de contraintes sur chaque composant de manière à garantir les propriétés globales de l'architecture.

Cette communication peut consister en un simple retour de données ou en une activité plus complexe (coordination de plusieurs services).

Cependant, l'aspect le plus important de l'architecture SOA est qu'elle permet de séparer l'implémentation du service de son interface. C'est cette caractéristique qui assure le haut degré d'interopérabilité visé par cette architecture. [3]

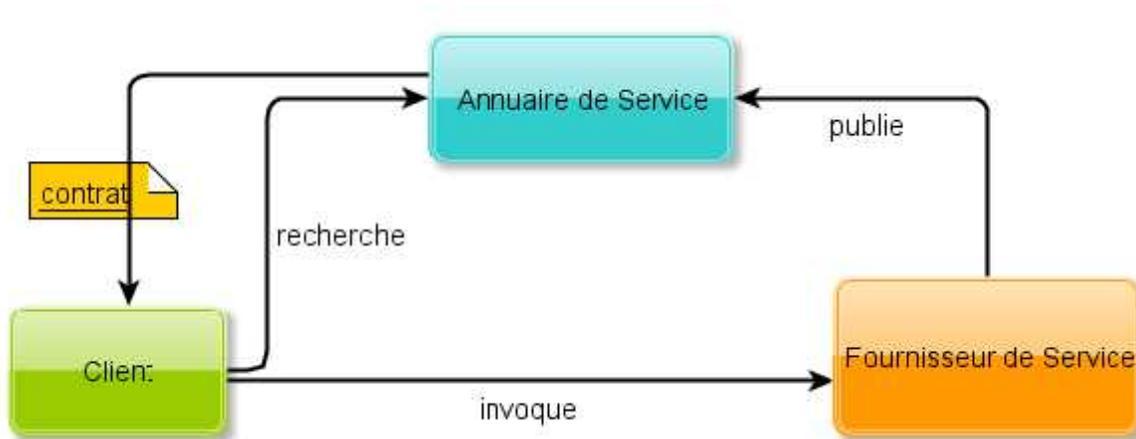


Figure 2. 4. Modèle fonctionnel de l'architecture SOA.

2.2.4. Concepts de SOA

SOA est plus qu'un ensemble de technologies. Elle n'est directement liée à aucune technologie, bien qu'elle soit le plus souvent mise en application avec des Web Services qui sont considérés comme la technologie la plus appropriée pour la réalisation de SOA.

Cependant, l'utilisation des Web Services n'est pas proportionnée pour construire SOA. Nous devons employer les Web Services selon les concepts que SOA définit. Les concepts de SOA les plus importants sont :

- ✓ Services.
- ✓ Interfaces Self-describing (description d'individu).
- ✓ Échange des messages.
- ✓ Soutien de liaison synchrone et asynchrone.
- ✓ Accouplement faible.
- ✓ Enregistrement de service.
- ✓ Qualité du service.
- ✓ Composition des services dans des processus d'affaires. [5]

2.2.5. Composants de la SOA

Afin de déduire les composants de bases de l'architecture SOA, il est nécessaire de présenter en premier lieu son paradigme de fonctionnement.

Le paradigme "découvrir, interagir et exécuter" comme montré dans la figure suivante permet au consommateur du service (client) d'interroger un annuaire pour le service qui répond à ses critères.

Si l'annuaire possède un tel service, alors il renvoie au client le contrat du service voulu ainsi que son adresse.

SOA consiste en quatre entités configurées ensemble pour supporter le paradigme découvrir, interagir et exécuter. [5]

Suivant ce protocole de fonctionnement, nous pouvons présenter les composants de l'architecture SOA comme suit :

✓ **Le consommateur de service**

Le consommateur de service est une application qui requière un service. C'est l'entité qui initie la localisation du service dans l'annuaire, interagit avec le service à travers un protocole et exécute la fonction exposée par le service. [5]

✓ **Le fournisseur de service**

Le fournisseur de service est une entité adressable via un réseau, il accepte et exécute les requêtes venant d'un client.

Le fournisseur de service publie le contrat de service dans l'annuaire pour qu'il puisse être accédé par les clients. [5]

✓ **L'annuaire de service**

L'annuaire de service est un annuaire qui contient les services disponibles. C'est une entité qui accepte et sauvegarde les contrats du fournisseur de service et présente ces contrats aux éventuels clients. [5]

✓ Le contrat de service

Le contrat spécifie la manière dont le client de service va interagir avec le fournisseur de service. Il spécifie le format de la requête et la réponse du service. [5]

Donc Le contrat de service est spécifié par le fournisseur de service. Il s'agit d'un document qui définit :

- ✓ **Les fonctions du service** : des descriptions abstraites de l'offre de service.
- ✓ **L'interface du service** : est une description des mécanismes et des protocoles de communication avec le prestataire de service.
- ✓ **La qualité du service** : les détails de fiabilité, de disponibilité, de robustesse, etc.

2.2.6. L'accès aux services

Dans l'architecture SOA, l'accès aux services peut être décomposé en trois étapes :

1. La recherche et la localisation des fournisseurs proposant le service souhaité, en utilisant l'outil de recherche et de localisation mis à disposition par l'infrastructure sur laquelle le client se trouve.
2. Le choix de l'un des fournisseurs trouvés lors de la première étape.
3. La réalisation du service par l'invocation du fournisseur choisi en utilisant l'une des opérations de l'interface du service. [3]

2.2.7. Avantages de l'architecture orientée service

Les principaux avantages d'une architecture orientée service :

- ✓ **Au niveau organisationnel : la flexibilité du système** :
 - La flexibilité technologique se définit comme « la capacité de la technologie à se transformer et à s'adapter au rythme des activités d'une façon aisée et au moindre coût possible » Dans cette optique, l'architecture orientée service permet la possibilité de créer un environnement technologique flexible grâce à la standardisation des infrastructures et la décomposition des applications en services. [6]

✓ Au niveau technique

- La réutilisabilité des ressources : Réduction de la complexité du système grâce à ses fonctionnalités réutilisables et ses services indépendants.
 - Réaliser des économies en dépenses et en temps nécessaire à la réalisation de nouveaux projets.
 - Diminue les redondances et les erreurs pouvant apparaître dans le système tout en accélérant le temps d'automatisation des processus.
 - Simplifie également les problèmes de maintenance grâce à sa structure organisée.
 - L'interopérabilité : permet d'une part la standardisation des protocoles pour un meilleur échange entre les services, et d'autre part la transformation des données entre les services consommateurs et les services producteurs.
 - L'indépendance totale entre les composantes crée la neutralité technologique sans contrainte de technologies ou de localisation des services. Ainsi l'échange d'informations entre les divers services est exécuté de manière adéquate pour une meilleure communication dans l'ensemble des composants du système.

✓ Au niveau financier

- La baisse des coûts : La plus importante valeur ajoutée du concept d'architecture orientée service est sa capacité de réduction des coûts sur le plan financier.

2.3. Les Web Services

D'après la définition, SOA est une approche architecturale qui ne fait aucune hypothèse sur la technologie de mise en œuvre. En particulier, l'amalgame souvent faite entre SOA et les Web services est une erreur.

Cependant, la conception des spécifications Web services a été menée dans l'objectif de répondre au mieux aux enjeux de l'architecture SOA Les Web services fournissent les bases technologiques nécessaires à la réalisation de l'interopérabilité entre les applications en utilisant différentes plateformes, différents systèmes d'exploitation et différents langages de programmation. [Ref2.5 page 43]

Nous peuvent dire que ce modèle est la synthèse actuelle entre le modèle client-serveur classique et le modèle Internet. Cette architecture consiste à mutualiser les traitements les plus

communs (services) sur un serveur accessible depuis l'Internet (WEB). Les clients qui effectuent des traitements spécifiques peuvent à tout moment déléguer les tâches générales à un ou plusieurs Web Services et réintégrer le résultat à leur traitement de façon transparente.

2.3.1. L'apparition des Web Services

Les Web services sont nés de l'effort de plusieurs organisations qui ont partagé un intérêt commun en développant et en maintenant "un marché électronique". Celles-ci souhaitent pouvoir communiquer plus simplement et sans avoir à se concerter sur chacune de leur transaction pour pouvoir interpréter leurs différentes données. Elles souhaitent supprimer l'isolement de leur système informatique avec les autres.

Services Web ont évolué à travers trois phases distinctes :

PHASE	PÉRIODE	DESCRIPTION
1	1999 – 2001	En Juin 2001 Gartner Group documenté un calendrier pour l'adoption des services Web à partir de 2001 - 2005. Ils ont suggéré que l'année 2001 verrait de nombreux services Web des outils de développement livrés. Avec des outils de la version finale de Microsoft, IBM, Oracle et bien d'autres, cette prolifération d'outils de services Web est en bonne voie.
2	2002 – 2004	Gartner a suggéré que 2002 verra Web Services commerciaux commencent à apparaître en grand nombre, aussi avec (B2C) l'accès Business-to-Consumer de masse des services Web axés sur le consommateur. Un exemple de ces Services Web B2C est Mes Services de Microsoft (de Hailstorm nom de code). Mes services sont prévus pour être libéré en 2002 en association avec Microsoft .NET. De 2003 à l'adoption de l'UDDI Registres (Universal Description, Discovery and Integration). Ce devrait croître, avec des registres privés émergents pour soutenir les échanges privés. Registres publics apparaîtront à soutenir les échanges publics, avec l'utilisation

		de services du gouvernement du Web accélérer aussi fortement. Certains de ces registres peuvent offrir un accès gratuit à des services Web, mais la plupart devraient être libérés sur une base payante.
3	A partir de 2005	L'année 2005 verra publics UDDI Registres attirer l'attention tant publics échanges B2B commencent à réapparaître après un hiatus par rapport à 2001-2002 durant la récession dot-com. Services Web dynamiques seront également gagner plus d'attention. Pour les services Web pour livrer une intégration rapide et transparente des partenaires commerciaux à l'échelle de l'entreprise pendant la phase 2, un certain nombre de questions devront être abordées. Ceux-ci comprennent: la Qualité de Service (QoS); la fiabilité du réseau; récupération de transaction; messagerie en temps réel; la sécurité; et des mécanismes de facturation.

2.3.2. Définitions

Nous trouvons plusieurs façons qui définissent ce modèle.

- **Citation Wikipédia**

Un service Web est un programme informatique permettant la communication et l'échange de données entre les applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et en temps réel.

- **La Citation W3C**

Un service Web est un composant logiciel identifié par une URI, dont les interfaces publiques sont définies et appelées en XML. Sa définition peut être découverte par d'autres systèmes logiciels. Les services Web peuvent interagir entre eux d'une manière prescrite par leurs définitions, en utilisant des messages XML portés par les protocoles Internet.

- **Citation Dico du Net**

Une technologie permettant à des applications de dialoguer à distance via Internet indépendamment des plates-formes et des langages sur lesquels elles reposent.

- **En d'autres termes**

- Un service Web est tout simplement un programme accessible au moyen d'Internet, qui utilise un système de messagerie standard XML, et n'est lié à aucun système d'exploitation ou langage de programmation. [8]

- Les services Web sont des technologies émergentes permettant une interopérabilité entre les différents acteurs (fournisseurs et demandeurs de services) du fait de leur architecture reposant sur des technologies standard. Ils ont pour vocation de favoriser une architecture orientée services, intégrant des systèmes hétérogènes complexes, fortement distribués et pouvant coopérer sans recourir à une intégration spécifique et coûteuse. [3]

- Un Web service est un composant logiciel identifié par une URI, qui possède une interface publiable. Cette dernière peut être découverte par d'autres systèmes, qu'ils peuvent interagir avec le Web service selon les règles prescrites par sa description, en utilisant des messages basés sur XML et portés par des protocoles standards d'Internet. [5]

- Les Web Services fournissent une couche d'abstraction entre le client et le fournisseur d'un service. Cette couche est indépendante de la plateforme et du langage d'implémentation, grâce à un ensemble de protocoles standardisés comme **SOAP** (Simple Object Access Protocol), **WSDL** (Web Service Description Language) et **UDDI** (Universal Description, Discovery and Integration). [5]

2.3.3. Les caractéristiques et l'intérêt des services Web

D'après les définitions précédentes nous pouvons dire que :

- Ils sont utilisables à distance via n'importe quel type de plate-forme.
- Ils peuvent servir au développement d'applications distribuées et sont accessibles depuis n'importe quel type de clients.

- Ils appartiennent à des applications capables de collaborer entre elles de manière transparente pour l'utilisateur. [9]
- Ils sont des composants d'application.
- Ils communiquent en utilisant des protocoles standards.
- Ils sont autonomes et auto- descriptifs.
- Ils peuvent être découverts.
- Ils peuvent être utilisés par d'autres applications.
- Ils s'appuient sur XML (Extensible Markup Language).
- Ils sont extensibles : chacun peut adjoindre ses propres données, protocoles ou mécanismes propriétaires. [5]
- Son interface décrite d'une manière interprétable par les machines, qui permet aux applications clientes d'accéder aux services de manière automatique.
 - Son utilisation de langages et protocoles indépendants des plates-formes d'implantation, qui renforcent l'interopérabilité entre services.
 - Son utilisation des normes actuelles du Web, qui permettent la réalisation des interactions faiblement couplées et favorisent aussi l'interopérabilité. [8]

2.3.4. Les applications des services Web

Les technologies des services Web peuvent être appliquées à toutes sortes d'applications auxquelles elles offrent des avantages considérables en comparaison aux anciennes API propriétaires, aux implémentations spécifiques à une plate-forme et à quelques autres restrictions classiques que l'on peut rencontrer (multi-plateforme, multi-langage, disponible sur Internet avec une information actualisée disponible en temps réel, ...).

L'application des services Web est multiple, autant dans les domaines du **B2C**, **B2B** que pour des domaines de gestion, par exemple gestion de stock, gestion commerciale, etc...

- **B2C (Business to Consumer)** : Qualifie une application, un site Internet destiné au grand public.

- **B2B (Business to Business)** : Qualifie une application, un site Internet destiné au commerce de professionnel à professionnel.

Les services Web peuvent être utiles dans la plupart des scénarios applicatifs lorsque la communication peut être établie sur un modèle bidirectionnel (requête/réponse). C'est néanmoins loin d'être aussi limitatif, beaucoup d'autres modèles peuvent avoir recours aux services Web, sans même que vous vous en rendiez compte. Les entreprises qui mettent à disposition leurs services Web permettent aux développeurs intéressés par ses fonctionnalités de les réutiliser sans avoir à les recoder.

Le principe des services Web permet d'avoir un partage des fonctionnalités et facilite grandement le développement. [9]

2.3.5. Les types de services web

- Un service web est une technologie client-serveur fondée sur les protocoles du web.
- Un annuaire web est un site web répertoriant des sites web.
- Un moteur de recherche est un site permettant d'effectuer des recherches sur l'ensemble des sites web selon différents critères.
- Un portail web est un site web tentant de regrouper la plus large palette d'informations et de services possibles dans un site web. Certains portails sont thématiques.
- Un agrégateur web est un site web qui sélectionne, organise et, éventuellement, valide des pages (URL) concernant un sujet précis et les met en forme de façon ergonomique et/ou attractive.
 - Un blog est un journal web, fournissant des informations dépendantes du temps. Un blog BD est un type de blog utilisant le medium de la bande-dessinée comme support de communication.
 - Un forum est un site permettant de faire des échanges entre différents utilisateurs, utilisé notamment dans le support ou la communication.
 - Un site d'e-commerce est une boutique en ligne permettant de commander des produits ou services à distance.
 - Un comparateur est un site chargé de comparer différents produits, services ou prix.
 - Un gestionnaire de projet, est un outil de projet et de support pour différents types d'activité.

- Un Logiciel de suivi de problèmes est un site spécialisé dans le support, il permet de trier les problèmes de retrouver des traces de problèmes et les solutions apportées et d'échanger des données permettant à la résolution.

- Un web mail est un lecteur de courrier électronique utilisant une interface web.
- Un wiki est un site web éditable par les utilisateurs. [10]

2.3.7. Architecture des services des web

2.3.7.1. Architecture de base

Les services Web possèdent trois acteurs principaux :

a. Le fournisseur de service (service provider) : Il définit le service à publier et la description du service dans l'annuaire.

b. L'annuaire (service broker) : Il reçoit et enregistre les descriptions des services publiés par les fournisseurs, d'autre part il reçoit et répond aux recherches de services lancées par les clients.

c. Le demandeur (service requestor) : obtient la description du service grâce à l'annuaire utilisé par les Services Web et invoque les services demandés. [11]

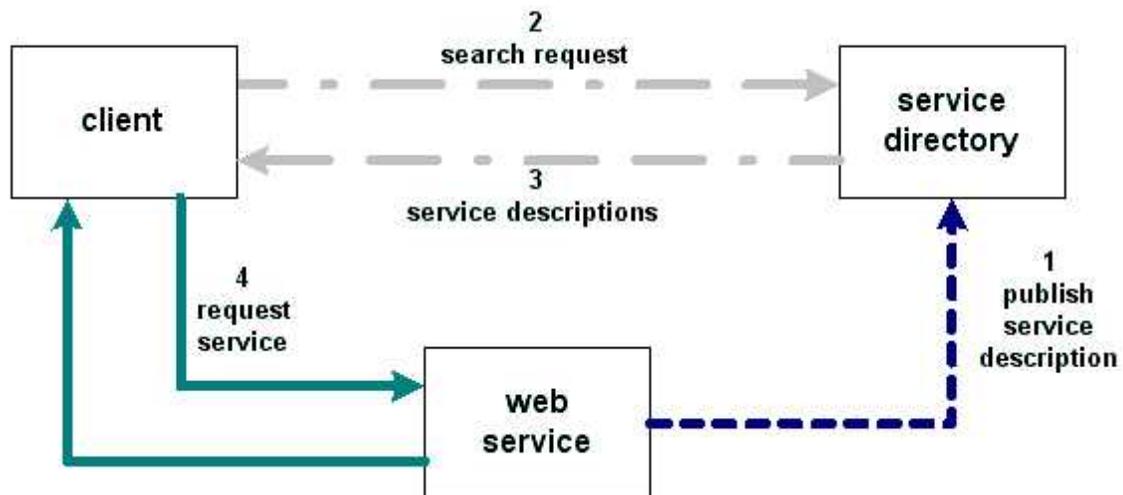


Figure 2. 5. L'architecture générale du Web Service.

2.3.7.2. Architecture en couche

L'architecture standard d'un service Web est implémentée à l'aide des diverses technologies et standards (UDDI, WSDL, SOAP et HTTP), organisés en quatre couches. Chacune d'elles répond à des préoccupations fonctionnelles différentes telles que la publication, la description, la messagerie et le transport, comme illustré sur la Figure suivante : [12]

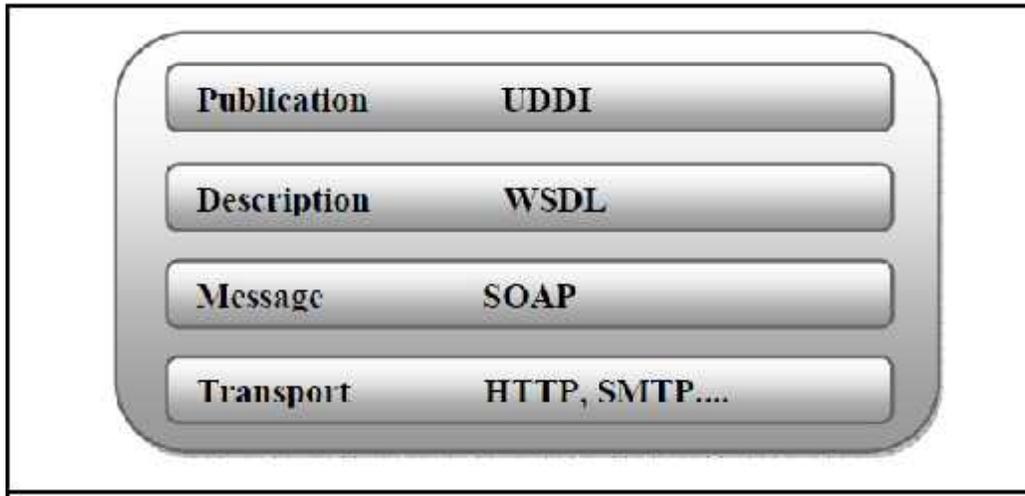


Figure 2. 6. Le cadre architectural standard des services Web.

Dans un souci d'interopérabilité, les différentes couches de l'architecture d'un service Web s'interfaçent avec des standards, comme suit :

- ✓ **La couche de publication** : repose sur le protocole UDDI (*Universal Description, Discovery and Integration*), qui assure le regroupement, le stockage et la diffusion des descriptions des services Web.
- ✓ **La couche description** : est prise en charge par le langage WSDL (*Web Service Description Language*), qui décrit les fonctionnalités fournies par le service Web, les messages reçus et envoyés pour chaque fonctionnalité, ainsi que le protocole utilisé pour la communication.
- ✓ **La couche message** : la couche message utilise des protocoles reposants sur le langage XML, car sa syntaxe unique résout les conflits syntaxiques lors de l'encodage des données.

Actuellement, SOAP (*Simple Object Access Protocol*) est le protocole le plus utilisé pour cette couche.

✓ **La couche transport** : le protocole le plus utilisé dans cette couche est l'HTTP (*Hyper Text Transfert Protocol*). Cependant, d'autres protocoles peuvent être utilisés, tels que le SMTP (*Simple Mail Transfer Protocol*) ou le FTP (*File Transfer Protocol*), permettant ainsi aux services Web de rester indépendants du mode de transport utilisé. [12]

Remarque

L'architecture étendue (avancée) est constituée de plusieurs couches se superposant les unes sur les autres, d'où le nom de pile des Web services. La figure suivante les montre décrite un exemple d'une telle pile.

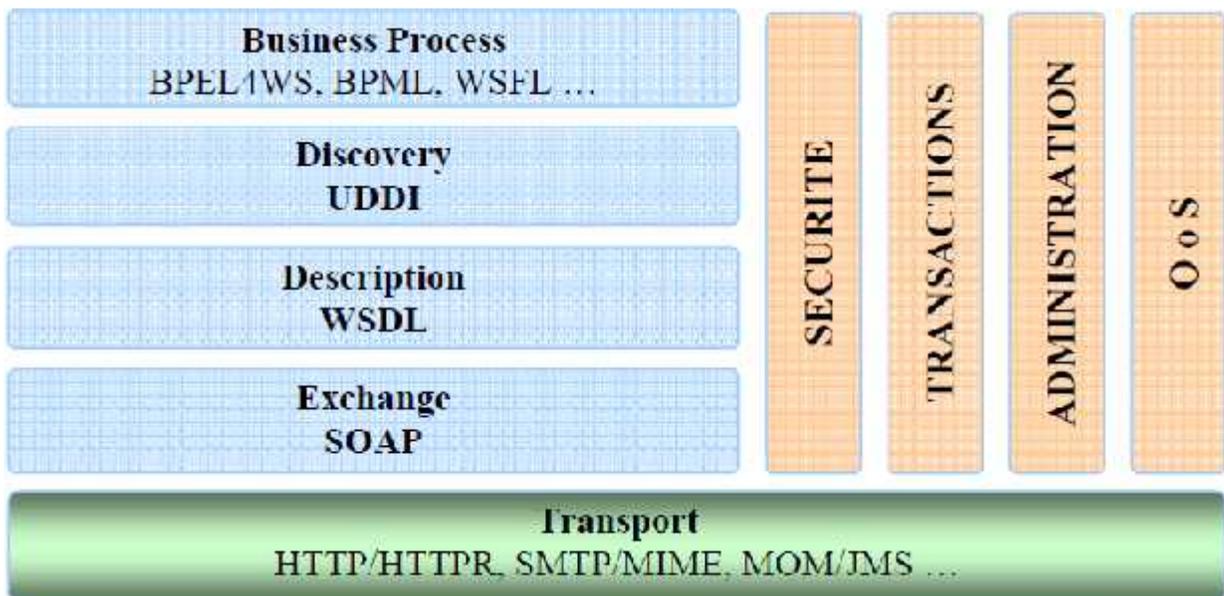


Figure 2. 7. L'architecture en pile (étendue).

La pile est constituée de plusieurs couches, chaque couche s'appuyant sur un standard particulier. On retrouve, au-dessus de la couche de transport, les trois couches formant l'infrastructure de base décrite précédemment. Ces couches s'appuient sur les standards émergents SOAP, WSDL et UDDI. Comme mentionné précédemment, l'infrastructure de base définit les fondements techniques permettant de rendre les business processus accessibles à l'intérieur d'une entreprise et au-delà même des frontières d'une entreprise.

Dans ce contexte deux types de couches permettent de la compléter :

(i) les couches dites transversales (ex : sécurité, administration, transactions et qualité de services (QoS)) rendent viable l'utilisation effective des services Web dans le monde industriel.

(ii) une couche Business processus permet l'utilisation effective des services Web dans le domaine du e-business. [3]

2.3.8. Le cycle de vie d'un service web et le fonctionnement d'un service web

2.3.8.1. Le cycle de vie d'un service web

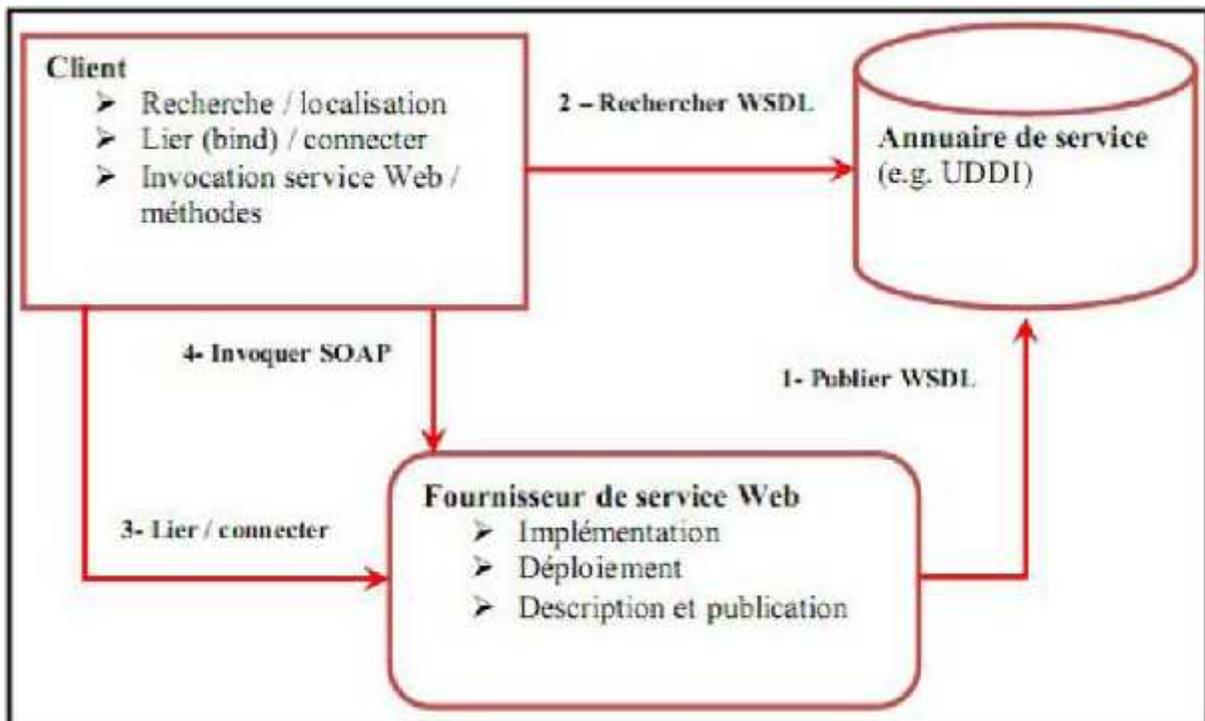


Figure 2. 8. Le cycle de vie d'un service web.

L'emploi d'un service web connaît le cycle de vie suivant :

D'abord, on effectue le déploiement du service web, en fonction de la plateforme.

Ensuite, on enregistre le service web à l'aide de WSDL (Web Services Description Language), dans l'annuaire UDDI.

L'étape suivante est la découverte du service web par le client, par l'intermédiaire d'UDDI qui lui donne accès à tous les services web inscrits. Pour ce, on utilise SOAP.

Enfin, Le client invoque le service web voulu, ce qui termine le cycle de vie de ce service web.

Remarque

Les interactions entre les services Web impliquent trois participants : le fournisseur de services, l'annuaire de services et le client du service

- **Le fournisseur** : correspond au propriétaire du service du point de vue métier, et du point de vue architecture technique ; il est constitué par la plate-forme d'accueil du service.
- **Le client** : d'un point de vue métier correspond au demandeur du service, et d'un point de vue architecture technique, il est constitué par l'application de recherche et d'invocation d'un service. L'application cliente peut être elle-même un service Web.
- **L'annuaire** : correspond à un registre de descriptions des services offrant des facilités de publication de services pour les fournisseurs ainsi que des facilités de recherche pour les clients. [12]

2.3.8.2. Le fonctionnement d'un service web

a. Services Web non publics

- Services web non publiés dans un annuaire UDDI.
- Services web dont le point d'accès est connu des utilisateurs du service.
- Généralement des SW intranet, des SW de type B2B (Business to Business). [16]



Figure 3. Fonctionnement d'un service web à accès non public

b. Services Web publics

- ✓ Services web publiés dans un annuaire UDDI.

- ✓ Généralement des SW de type B2C (Business to Consumer), ex : agence de voyage, ...etc.
- ✓ Le SW et l'annuaire UDDI qui le publie peuvent ne pas résider sur la même machine. [16]

2.3.9. Les standards du service Web

Le concept des services web s'articule actuellement autour de plusieurs standards, nous présentons dans la suite les trois standards de base, à savoir, SOAP (*Simple Object Access Protocol*), WSDL (*Service web Description Language*) et UDDI (*UniversalDescription, Discovery and Integration*).

Sans oublions que ces standards basés sur le langage XML (*eXtensible Markup Language*) dont nous allons présenter son rôle dans les services web, en suite nous nous intéresserons aux trois standards cités ci-dessus.

2.3.9.1. XML – EXtensible Markup Language

a. Définition

XML est un standard promulgué par le W3C en 1998, l'organisme chargé de standardiser les évolutions du Web. On retrouve dans XML une généralisation des idées contenues dans HTML, on n'utilise les balises que pour décrire l'aspect graphique que doit revêtir la page dans le navigateur Web. Dans XML, les balises permettent d'associer toute sorte d'informations au fil du texte. [9]

XML a été conçu pour des documents arbitrairement complexes, tout en s'appuyant sur cinq grands principes simples et clairs :

- ✓ Lisibilité à la fois par les machines et par les utilisateurs.
- ✓ Définition sans ambiguïté du contenu d'un document.
- ✓ Définition sans ambiguïté de la structure d'un document.
- ✓ Séparation entre documents et relation entre document.
- ✓ Séparation entre structure du document et présentation du document. [9]

Le XML permet de structurer l'information dans des fichiers textes. On peut l'utiliser typiquement comme fichier de configuration pour des programmes mais aussi pour enregistrer des résultats (mesures, carnet d'adresse, liste de pièces,...)

Le XML s'impose de plus en plus car il permet de structurer l'information sous une forme plus robuste que les fichiers binaires ou tabulaires. Cette forme permet à la fois de faciliter le traitement informatique (internet, intranet, tableur, base de données,...) tout en conservant un support texte lisible et éditable sans outil particulier par l'être humain. [14]

b. XML Namespaces « espace de nommage »

XML Namespaces est une recommandation du W3C qui a été rapidement adoptée après XML0.1, visant à résoudre le problème de l'ambiguïté éventuelle des balises dans un document XML.

Les namespaces permettent ainsi de résoudre le problème des différences éventuelles d'interprétation du même document XML par des applications différentes. En s'appuyant sur le dispositif des URI, qui en assure l'unicité, et au prix d'une écriture un peu plus «bavarde», les balises et les attributs XML sont alors dotés d'une interprétation spécifique, non ambiguë. [9]

c. XML schéma

La recommandation XML schéma, adoptée après de longues discussions dans les comités techniques du W3C, représente un réel tour de force et une innovation dans l'utilisation de XML, rompant tout net avec son usage original de la publication des documents. XML schéma précise comment représenter en XML la structure des données en générale – ce qu'on a l'habitude d'appeler les métadonnées dans le monde des bases de données relationnelles (description des tables, des colonnes, de leurs types etc. ...). [9]

d. Rôle de XML dans l'infrastructure des services Web

Dans HTML, on n'utilise les balises que pour décrire l'aspect graphique que doit revêtir la page dans le navigateur Web. Dans XML, les balises permettent d'associer toutes sortes d'informations au fil du texte.

On le retrouve par exemple aussi bien comme élément de sauvegarde de documents au sein de fichiers ou de bases de données ou encore comme format d'échange de données.

XML est aujourd'hui un standard qui permet de décrire des documents structurés transportables sur les protocoles communs d'Internet. Il constitue la technologie de base des architectures services web, il est un facteur important pour contourner les barrières techniques. En effet, il apporte à l'architecture l'extensibilité et la neutralité vis à vis des plates-formes et des langages de développement. De plus, grâce à la structuration, XML permet la distinction entre les données des applications et les données des protocoles permettant ainsi une correspondance facile entre les différents protocoles.

L'interopérabilité entre les systèmes hétérogènes demande des mécanismes puissants de correspondance et de gestion des types de données des messages entre les fournisseurs et les clients. Pour les services Web, on utilise systématiquement XML avec les Namespaces et la spécification XML Schema, tous deux indispensables pour exprimer les structures des données habituellement complexes figurant dans les messages échangés.

```
<?xml version='1.0' encoding='utf-8' ?>
<!-- Simple module configuration for tracing with uranine -->

<CONFIGURATION>
  <ACQUISITION_SEQUENCE value="0">  <!-- ++++++ SLOW SEQUENCE ++++++++ -->
    <PROBES_POWER state="3"/>
    <ACQ_VBAT channel="0" threshold="11.9" />  <!-- SMS EOL on low battery -->
    <PROBES_POWER state="C"/>
    <CONDITIONAL_SMS>
      alarme, dseacute;passement niveau d'eau
    </CONDITIONAL_SMS>
  </ACQUISITION_SEQUENCE>
</CONFIGURATION>
```

Figure 2. 9. Un fichier XML.

e. Avantages / Inconvénients [14]

AVANTAGES	
Fichier texte	C'est un fichier texte, donc il sera toujours lisible dans des décennies. On garantit ainsi une meilleure pérennité de l'information
Le XML est standard	Cela signifie qu'il existe de nombreux outils informatiques qui permettent de lire ou d'écrire du XML. On trouve des librairies C, C++, Java, PHP, perl, ... De plus en plus d'outils sont capables de lire des fichiers XML (Internet Explorer, Excel, Mathematica, ...)
Le XML est strict	On ne peut pas écrire le XML n'importe comment; Vous êtes obligés de suivre une certaine syntaxe. Ça permet de garantir que le fichier soit toujours lisible. Pour vérifier la syntaxe d'un fichier XML, vous pouvez l'ouvrir dans Internet Explorer. Si le fichier est incorrect, Internet Explorer indiquera l'endroit de l'erreur.
Le XML est structuré et hiérarchique	Le fichier contient des <BALISES> qui peuvent contenir d'autres balises et ainsi de suite (hiérarchie). L'ordre d'apparition des balises est conservé.
On peut ajouter des commentaires	Les commentaires sont des éléments prévus par la spécification. On peut en rajouter dans le fichier sans casser la structure. Ceci permet de commenter des fichiers afin de garantir une meilleure pérennité de l'information
INCONVENIENTS	
Le XML est verbeux	C'est vrai. Les fichiers XML sont plus gros que des fichiers binaires ou tabulaires. Mais on peut facilement les compresser pour le stockage (avec des outils OpenSource par exemple). On assure ainsi aussi une pérennité de l'information à très long terme
Le tabulaire est mieux compris par Excel	C'est vrai, mais les choses ne peuvent pas forcément toujours être décrites par des tableaux 2-dimensions. De plus on peut difficilement ajouter des commentaires dans les fichiers tabulaires

2.3.9.2. Le protocole SOAP (simple Object Access Protocol)

a. Présentation

SOAP est un protocole adopté par le Consortium W3C. Le Consortium W3C crée des standards pour le Web : son but est donc de créer des standards pour favoriser l'échange d'information.

SOAP veut dire Simple Object Access Protocol. Si l'on voulait traduire cette définition en français cela donnerait Protocole Simple d'Accès aux Objets. En effet, le protocole SOAP consiste à faire circuler du XML généralement, via du http sur le port 80 (ou en utilisant un autre protocole). Cela facilite grandement les communications, car le XML est un langage standard et le port utilisé est le port 80, qui ne pose donc pas de problème pour les firewalls (pare-feu) de l'entreprise, contrairement à d'autres protocoles.

Tout comme la technologie des services Web, le protocole SOAP est très jeune. Le protocole SOAP a été créé en septembre 98, avec la version 0.9, par trois grandes entreprises :

Microsoft, UserLand et DevelopMentor. IBM n'a participé au protocole SOAP qu'à partir de la version 1.1 en Avril 2000. C'est cette même année que SOAP a été soumis au W3C.

Depuis septembre 2000, SOAP 1.1 est en refonte complète pour donner jour à la version 1.2 avec un groupe de travail de plus de 40 entreprises, on trouve bien sur Microsoft, IBM mais aussi HP, Sun, INTEL... . [9]

b. Définition

SOAP est un protocole d'invocation de méthodes sur des services distants. Basé sur XML, SOAP a pour principal objectif d'assurer la communication entre machines. Le protocole permet d'appeler une méthode RPC et d'envoyer des messages aux machines distantes via HTTP. Ce protocole est très bien adapté à l'utilisation des services Web, car il permet de fournir au client une grande quantité d'informations récupérées sur un réseau de serveurs tiers, [8]

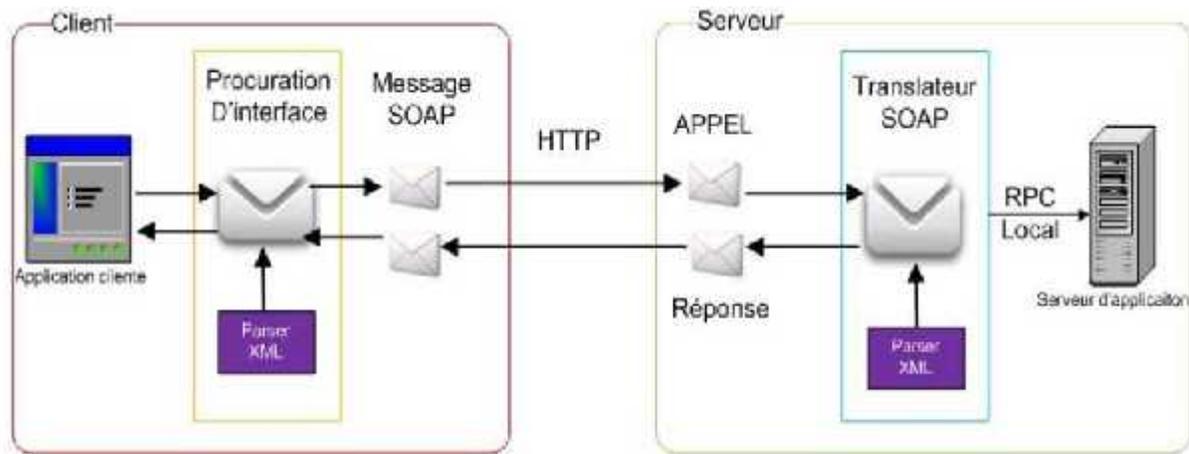


Figure 2. 10. Le Message SOAP.

SOAP est bien plus populaire et utilisé que XML-RPC. C'est une recommandation du W3C. D'après cette recommandation, SOAP est destiné à être un protocole léger dont le but est d'échanger des informations structurées dans un environnement décentralisé et distribué.

Une des volontés du W3C vis-à-vis de SOAP est de ne pas réinventer une nouvelle technologie. SOAP a été construit pour pouvoir être aisément porté sur toutes les plateformes et les technologies existantes. [8]

c. Ses caractéristiques

SOAP permet une normalisation des échanges de données. Les données sont encodées en XML et échangées par des appels de procédures à distance en utilisant HTTP/SMTP/POP comme protocole de communication.

- ✓ Simple, extensible et permet le diagnostic des erreurs.
- ✓ Message unidirectionnel (Requête_ Réponse).
- ✓ Fonctionne de manière synchrone et asynchrone.
- ✓ Indépendant de la plate-forme et du langage.
- ✓ N'est pas perturbé par un pare-feu. [9]

d. Fonctionnement

La requête SOAP transmise par le client à l'aide des protocoles de transmissions (exemple HTTP) passe par Listener () que le permet d'adresser (cas de java) aux bibliothèques donnée au client l'accède aux méthodes des services. Si l'exécution des méthodes a besoin des informations de la base des données alors il doit connecter avec la base des données. [13]

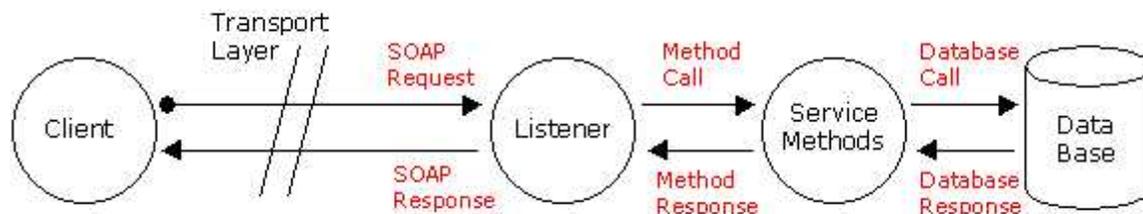
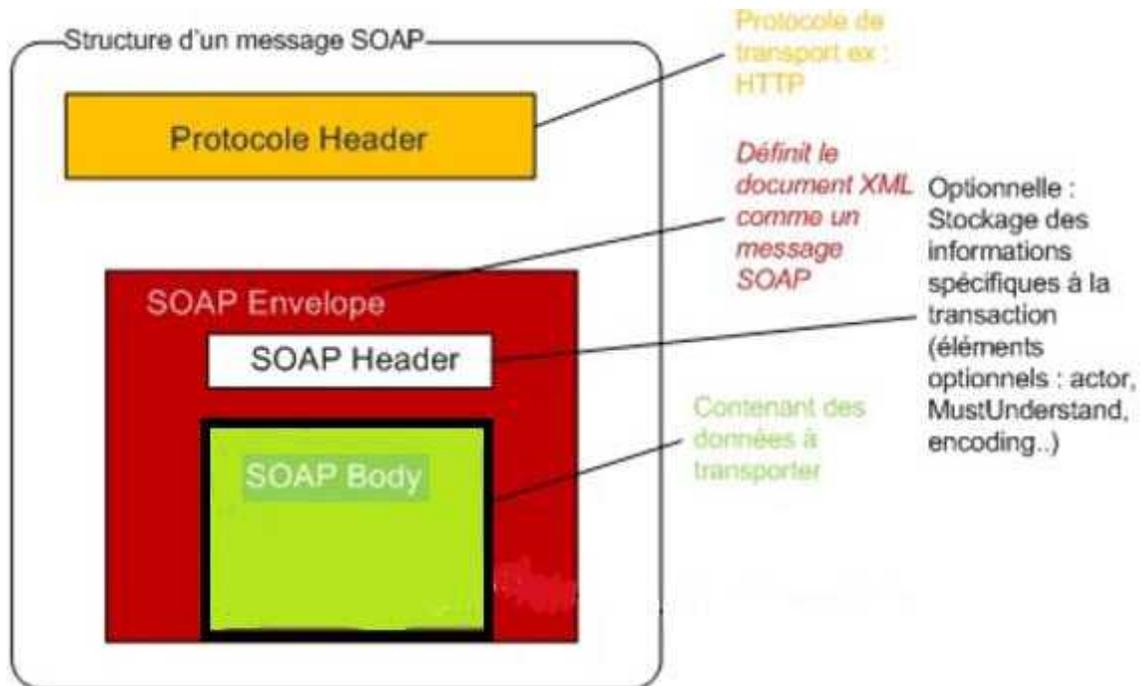


Figure 2.13. Traitement d'un message SOAP.

e. Structure d'un message SOAP

La grammaire de SOAP est assez simple à comprendre. Elle procure un moyen d'accès aux objets par appel de méthodes à distance. Les deux plus fortes fonctionnalités de SOAP sont sa simplicité et le fait que tout le monde a accepté de l'utiliser. Un message SOAP est

composé de deux parties obligatoires : l'enveloppe SOAP et le corps SOAP ; et une partie optionnelle : l'entête SOAP. [8]



```
<soap:Envelope
xmlns:"http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
  </soap:Header>
  <soap:Body>
  </soap:Body>
</soap:Envelope>
```

Figure 2.14. La structure d'un message SOAP.

e1. SOAP envelope (*enveloppe*) est l'élément de base du message SOAP. L'enveloppe contient la spécification des espaces de désignation (**namespace**) et du codage de données.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding">

  <soap:Header>
    <!-- en-tête -->
  </soap:Header>

  <soap:Body>
    <!-- corps -->
  </soap:Body>
</soap:Envelope>
```

Figure 2.15. SOAP envelope.

Toutes les balises XML associées à SOAP ont le préfixe soap (on trouve des développeurs utilisant "soap-env"). L'entête est <soap:Header> et le corps <soap:Body>.

SOAP repose entièrement sur les espaces de noms XML. Dans cet exemple, les espaces de noms sont introduits à l'aide d'un mot-clé « xmlns » *XML namespace* qui signifie espace de noms XML.

L'espace de noms est utilisé pour identifier toutes les balises afin d'éviter les conflits. La spécification impose que tous les attributs contenus dans l'enveloppe SOAP soient explicitement associés à un *namespace*, de manière à supprimer toute ambiguïté. Par convention, la spécification SOAP définit deux *namespaces* fréquemment utilisés :

- **soap-env** ou soap associé à l'URI « [...]schemas.xmlsoap.org/soap/envelope » pour définir le *namespace* de l'enveloppe dans la version 1.1, et à « [...]www.w3.org/2001/06/soap-envelope » dans la version 1.2 reprise par le W3C.
- **soap-enc**: encoding Style associé à l'URI « [...]schemas.xmlsoap.org/soap/encoding » pour la définition des formats de types de données dans la version 1.1, et à « [...]www.w3.org/2001/06/soap-encoding » dans la version 1.2.

Deux autres espaces de noms fortement utilisés dans SOAP sont « **xsd** » et « **xsi** ».

- **xsd namespace** précise que les balises proviennent de la définition de schéma XML.
- **xsi namespace** indique que les balises viennent d'une instance d'un schéma XML. [8]

e2. *SOAP header (entête)* est une partie facultative qui permet d'ajouter des fonctionnalités à un message SOAP de manière décentralisée sans agrément entre les parties qui

communiquent. C'est ici qu'il est indiqué si le message est mandataire ou optionnel. L'entête est utile surtout, quand le message doit être traité par plusieurs intermédiaires. [Ref2.8 page 8]

- L'en-tête d'un message SOAP commence avec la balise `<soap:Header>` et se termine avec la balise `</soap:Header>`, on peut aussi faire `<sopa-env:Header></sopa-env:Header>`.

Trois attributs associés à l'en-tête SOAP peuvent être utilisés :

- **soap:mustUnderstand** : cet attribut prend la valeur 1 ou 0. La valeur 1 signale que le récepteur doit reconnaître l'information présente dans l'en-tête et que son traitement est obligatoire. La valeur 0 indique que l'en-tête peut être ignoré par le récepteur.
- **soap:role** : sert à indiquer le destinataire SOAP auquel un bloc d'en-tête SOAP particulier est destiné.
- **soap:relay** : est utilisé pour indiquer si un bloc d'en-tête SOAP ciblé sur un récepteur SOAP doit être réacheminé (relayé) s'il n'est pas traité.

e3. *SOAP body (corps)* est un **container** pour les informations mandataires à l'intention du récepteur du message, il contient les méthodes et les paramètres qui seront exécutés par le destinataire final. [8]

```
<soap:Body>
  <checkAccountBalance>
    <accountNumber xsi:
type="xsd:int">1234567890</accountNumber>
  </checkAccountBalance>
</soap:Body>
```

Figure 2.16. SOAP body.

- Le corps du message SOAP commence avec la balise `<soap:Body>` et se termine avec la balise `</soap:Body>`.
- L'élément `<checkAccountBalance>` fournit le nom de la méthode à appeler : `checkAccountBalance`. L'élément `accountNumber` est un paramètre qui est passé dans la méthode `checkAccountBalance`.

• Les attributs xsi et xsd définissent les espaces de noms qui vont être utilisés dans le corps du message. La définition de xsi permet d'utiliser xsi:type dans le corps du message, le xsd:int signifie que cette valeur est de type entier. 1234567890 est la valeur donnée au paramètre. [8]

f. Les types de message SOAP

SOAP définit trois types de message :

- appel (call) obligatoire.
- Réponse(Reponce) optionnel.
- Erreur(Fault) optionnel. [13]

2.3.9.3. WSDL (Web Service Description Language)

WSDL est un standard du W3C, qui permet de définir une syntaxe XML pour décrire les méthodes et les paramètres des services Web invocables par le biais de messages au format SOAP. Il permet de définir qu'est-ce qu'un service Web est capable de faire, son emplacement et comment l'invoquer. De même que SOAP, WSDL est un langage indépendant des protocoles de transport de messages. [12]

a. Eléments de document WSDL

D'une manière générale, Les éléments qui composent un document WSDL sont les suivants :

- ✓ Les opérations proposées par le service Web.
- ✓ Les données et messages échangés lors de l'appel d'une opération.
- ✓ Le protocole de communication.
- ✓ Les ports d'accès au service. [12]

Dans un document WSDL, il existe une séparation entre deux niveaux indépendants (le niveau abstrait et le niveau concret). Le niveau abstrait regroupe les informations qui peuvent être réutilisées (c'est à dire, ils ne sont pas spécifiques à un service). Tandis que, le niveau concret est constitué de la description des protocoles d'accès au service Web (informations particulières à un service). [12]

a1. Le niveau abstrait

Ce niveau décrit les informations propres aux méthodes proposées par le service, ainsi que les informations traitant des messages et des données échangés lors de l'invocation du service. Si deux services proposent les mêmes méthodes, le niveau abstrait de description WSDL peut être réutilisé. Ce niveau est composé des informations suivantes :

✓ Les types de données

Le document WSDL permet de décrire les types de données échangées. WSDL supporte les types élémentaires (tels que les entiers, les chaînes de caractères, etc.) et les types complexes. Dans le cas où les données échangées possèdent une structure particulière (c'est à dire un type complexe), il est possible de les décrire par l'intermédiaire d'un XSD (XML *schema*).

✓ Les messages (les données)

Un message correspond aux données qui sont Véhiculées selon les méthodes invoquées. Chaque opération du service possède deux définitions de message : la première correspond à la requête (*Input*) et la seconde correspond à la réponse (*Output*). La description d'un message contient le nom de l'élément en paramètre d'entrée ou de sortie selon le type du message.

✓ Les opérations

Une opération représente une unité de travail, c'est-à-dire, une Méthode proposée par le service Web décrit. Chaque opération est identifiée par son nom. [12]

a2. Le niveau concret

Ce niveau décrit la manière dont le client accède à un service Web en particulier, il est non réutilisable (propre à un service unique). Les informations décrites dans le niveau concret sont les suivantes :

✓ Le protocole de communication

Cette description permet de définir le protocole à utiliser pour l'appel des méthodes du service.

✓ Les ports d'accès au service

Dans un document WSDL, l'accès au service est défini par une collection de ports d'accès. Chaque port représente la localisation du service par son URI. Un même service peut être accessible sur plusieurs ports différents. [12]

b. Structure d'un document WSDL

La structure de document WSDL se distingue selon les deux versions de WSDL, WSDL1.1 et WSDL 2.0. Une description WSDL est un document XML composé d'un élément racine (*definitions*) et de cinq sous-éléments obligatoires (*types*, *message*, *portType*, *binding* et *service*). La Figure suivante illustre la structure générale d'une description WSDL.



Figure 2.17. La structure générale d'un document WSDL 1.1.

✓ **L'élément « définitions »**

Est la racine du document WSDL, il contient les autres sous éléments (*types*, *message*, *portType*, *binding*, *service*) et les espaces de noms qui permettent de connaître la version de SOAP, les définitions de schéma XML utilisées dans le document et d'autres espaces de noms utiles, comme l'espace de nom pour le fournisseur du service Web.

✓ **L'élément « types »**

Il contient les définitions des types de données appliqués aux messages échangés. Afin de garantir une interopérabilité maximale, et une grande indépendance au niveau des plateformes, WSDL utilise XSD en tant que système de types de données.

✓ **L'élément « message »**

Il introduit les types de messages supportés par le service et décrit les données transmises lors des appels des méthodes du service Web.

✓ **L'élément « portType »**

Cet élément décrit l'ensemble des opérations proposées par le service Web.

✓ **L'élément « binding »**

Il définit les protocoles de communication et les spécifications des formats de données pour les ensembles d'opérations (portType).

✓ **L'élément « service »**

Il décrit la collection des ports d'accès au service. Cet élément permet de localiser le service Web et de faire des appels aux méthodes disponibles. [12]

2.3.9.4. UDDI (Universal Description, Discovery and Integration)

a. Définition

L'UDDI est un standard résultant d'une initiative d'un regroupement d'entreprises (Ariba, IBM, Microsoft), il constitue un référentiel commun d'entreprises visant à établir un format d'annuaire des services Web. UDDI permet de décrire et de localiser une entreprise (ou toute unité organisationnelle) ou un service, à partir des interfaces qui sont personnalisables. [12]

b. La recherche d'un service web

L'annuaire UDDI est consultable sous trois facettes (Comme illustré sur la figure suivante) de classification pour les services web pour trois types de recherche différents

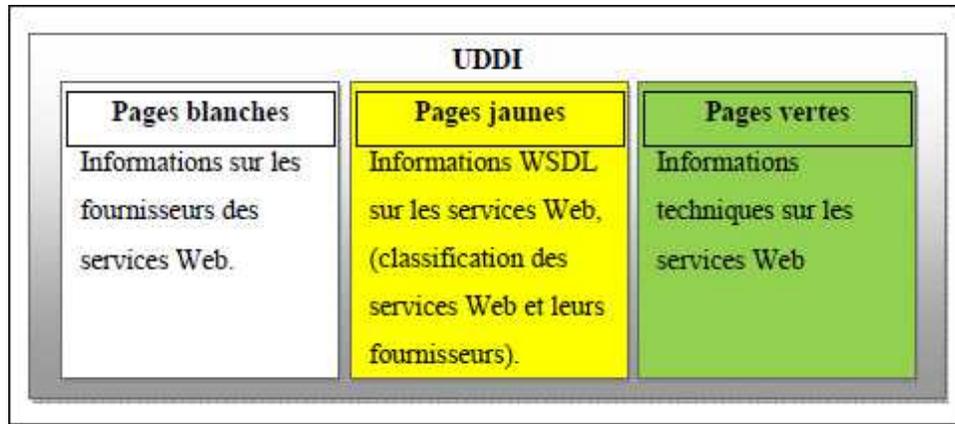


Figure 2.18. Les trois facettes de l'annuaire UDDI.

✓ **Pages blanches**

Elles comportent des informations sur les fournisseurs des services telles que le nom et les coordonnées du fournisseur.

✓ **Pages jaunes**

Elles comportent la description WSDL des services Web déployés par les fournisseurs ainsi que les informations permettant de les classer par catégorie.

✓ **Pages vertes**

Elles comportent les informations techniques détaillées sur les services fournis, telles que les descriptions des services Web et les informations de liaison sur les services. [12]

c. La publication d'un service web

L'enregistrement des services web dans un annuaire UDDI s'effectue auprès d'un opérateur en accédant au site web de ce dernier à partir d'un navigateur ou d'un outil intégré à un environnement de développement.

Grace à un jeu d'API XML basé sur SOAP, on peut interagir avec UDDI au moment de la conception d'exécution des S.W pour découvrir des données techniques et administratives sur les services et les entreprises qui publient, de manière que ces services puissent être invoquées et utilisés. UDDI envoyés sous forme des messages SOAP.

L'API se décompose en 3 groupes :

✓ La manipulation (save et delete).

- ✓ L'authentification des commandes par jeton (get_authToken et discard_authToken).
- ✓ L'ajout de relations inter-entreprises (joint_ventures). [13]

d. Les Caractéristiques d'UDDI :

- ✓ Neutre en terme de protocole _comme tout registre, il peut y avoir des pointeurs sur n'importe quoi (email, web page . . .).
- ✓ Possibilité de faire des recherches par domaine d'activité, service, Web Service, binding.
- ✓ Pas de QoS.
- ✓ Nécessite un modérateur Risques d'entrées erronées, de doublons, de fraude.
- ✓ Nœuds privés et publics autorisés. [16]

2.3.10. Problématique de recherche sur les services Web

Des recherches faites sur les services Web par rapport à SOA couvrent plusieurs problèmes intéressants : la Sélection des services Web, la Découverte et Découverte dynamique des services Web, la Composition des services Web... etc.

2.3.10.1. Sélection des services Web

Avec la sélection des services web, on cherche à choisir le meilleur fournisseur d'un service web, étant donné un ensemble de fournisseurs de ce service.

Un nouveau modèle de registre et découverte de services web a été défini basé sur la Qualité de Service (i.e. QoS en anglais Quality of Service - QoS).

La plate-forme de cette proposition est constituée de quatre rôles : Fournisseur de services web, Consommateur de services Web, Certificateur de la Qualité de Service et le nouvel registre.

Le fournisseur du service offre le service Web en publiant ce dernier dans le nouveau registre, le consommateur a besoin du service web offert par le fournisseur, le nouvel registre UDDI est un lieu de stockage de services web enregistrés avec facilités de recherche.

Le certificateur QoS sert à vérifier les revendications de qualité de service pour un service Web.

Le nouveau registre est différent de l'actuel modèle UDDI en ayant information sur la description fonctionnelle du service Web, et sur la qualité de service enregistré dans le stockage.

La consultation pourrait être faite selon la description fonctionnelle du service Web désirable, avec la qualité exigée de service attribuée comme des contraintes de consultation. [15]

2.3.10.2. La découverte de services Web

Dans le contexte d'une application qui a besoin d'exécuter une fonctionnalité implémentée comme un service web par plusieurs fournisseurs, la découverte fait référence au processus de recherche des services web implémentant la fonctionnalité souhaitée.

Les registres UDDI sont des entités qui servent d'appui à la découverte de services web pour les applications client. De cette façon une application interroge un registre UDDI pour les fournisseurs d'un service web. [15]

2.3.10.3. La découverte dynamique d'un service Web

S'effectue d'une façon différente. Au lieu de s'enregistrer dans un registre connu, les services Web découverts dynamiquement annoncent explicitement leur arrivée sur le réseau et leur départ. WS-Discovery définit des protocoles pour annoncer et découvrir des services Web via des messages de multi diffusion.

Lorsqu'un service Web se connecte à un réseau, il annonce son arrivée en émettant un message "Hello". Lorsqu'un service quitte un réseau, WS-Discovery précise qu'un message "Bye" doit être envoyé au réseau ou au proxy de découverte. Ce message informe les autres services du réseau que le service Web concerne n'est plus disponible.

La découverte dynamique étend aussi l'architecture de services Web à des équipements, comme les imprimantes et les systèmes de stockage qui peuvent être intégrés dans un système sous la forme de services Web, sans avoir besoin d'outils ou de protocoles spécialisés. [15]

2.3.10.4. Composition des services Web

L'objectif principal de l'approche service web est de transformer le web en un dispositif distribué de calcul où les programmes (services) peuvent interagir d'une manière intelligente

en étant capables de se découvrir automatiquement, de négocier entre eux et de se composer en des services plus complexes.

La composition de services reste un problème très complexe. Cette complexité tient au fait que les solutions de composition de services doivent tenir compte du nombre croissant de services déployés sur le web, de leur mise à jour continue et de leur hétérogénéité.

Les solutions proposées peuvent être classifiées selon deux axes :

a. Selon que la sélection des services et la gestion du flot soient faites ou non a priori, une approche sera dite statique ou dynamique.

a1. Composition statique

On dit qu'une composition est statique lorsqu'elle prend place à l'étape de conception, au moment où l'architecture et la conception du système logiciel sont planifiées. Les composants (ou services) qui seront utilisés sont préalablement choisis et reliés, et la gestion du flot est effectuée a priori, l'agrégation des différents services Web se fait de manière statique avant exécution. Les approches statiques de composition de services sont celles adoptées par l'industrie, elles s'inspirent de la gestion de processus métiers quant à la description des données et des flots de contrôle pour le processus de composition. [15]

a2. Composition dynamique

Par opposition, une composition de services est dite dynamique si les services sont sélectionnés et composés à la volée en fonction des besoins formulés par l'utilisateur.

Une approche dynamique pour la composition de services offre le potentiel de réaliser des applications flexibles et adaptables en sélectionnant et en combinant les services de manière appropriée sur la base de la requête et du contexte de l'utilisateur. [15]

b. En fonction du degré de participation de l'utilisateur dans la définition du schéma de composition, ces propositions sont manuelles, semi-automatiques ou automatiques.

b1. La composition manuelle

Les techniques de composition manuelles reposent sur un expert qui se charge de définir et générer, graphiquement ou moyennant un éditeur de texte, des scripts de workflow qui sont ensuite soumis à un moteur d'exécution de workflows. Elle est la plus adaptable aux besoins

de l'utilisateur, car il va tout définir à son goût depuis le début, en générale les utilisateurs sont soutenus avec des interfaces graphiques pour faciliter la composition, le service composite est vulnérable au changement, la correction est manuelle si le service composite est non valable ou lorsque un nouveau service apparaît. La plupart d'entre elles sont entièrement statiques puisque les services à composer sont préalablement sélectionnés, et le flot de composition défini a priori. Elle est inappropriée dans le cas où le problème de composition de services web est un problème avec une échelle large (Large-scale WSC problem). [15]

b2. La composition semi-automatique

Dans les approches semi-automatiques, des outils sont développées afin de proposer à l'utilisateur des suggestions sémantiques quant à la sélection des services à composer (ex. en fonction de leurs entrées/sorties, annotations, etc.) durant le processus de composition.

Les techniques de composition semi-automatiques sont un pas en avant en comparaison avec la composition manuelle, dans le sens qu'ils font des suggestions sémantiques pour aider à la sélection des services Web dans le processus de composition.

Durant le processus de composition, le framework propose à l'utilisateur de sélectionner des buts, des médiateurs et des opérateurs de flot de contrôle. Une fois que le service composé est défini, l'outil instancie les opérateurs de contrôle ainsi que le workflow. [15]

b3. La composition automatique

Les techniques de la composition automatique automatisent entièrement le processus de composition en le traitant généralement comme un problème de planification. Dans ce cas, le moteur de composition y est toujours dynamique.

La composition totalement automatisée prend en charge tout le processus de composition et le réalise automatiquement, sans qu'aucune intervention de l'utilisateur ne soit requise.

L'approche de la composition automatique peut résister au changement dynamique des services web car elle utilise des agents logiciels qui sont chargés de mettre la connexion entre les différents services web impliqués dans le problème de composition, sachant que ces agents ont la capacité de savoir si deux services peuvent se connecter ou non via l'utilisation des

correspondances syntaxiques ou sémantiques entre les paramètres des services web (Syntactic matching or semantic matching). [15]

2.3.11. Les avantages et les inconvénients des services web

2.3.11.1. Les avantages

Les services web engendrent plusieurs avantages dont on peut citer :

- **L'interopérabilité**

Les services web fournissent l'interopérabilité entre divers logiciels fonctionnant sur diverses plates-formes.

- **La simplicité**

Les services web réduisent la complexité des branchements entre les participants. Cela se fait en ne créant la fonctionnalité qu'une seule fois plutôt qu'en obligeant tous les fournisseurs à reproduire la même fonctionnalité à chacun des clients selon le protocole de communication supporté.

- **Une composante logicielle légèrement couplée**

L'architecture modulaire des services Web, combinée au faible couplage des interfaces associées, permet l'utilisation et la réutilisation de services qui peuvent facilement être recombinaisonnés à différents autres applications.

- **L'hétérogénéité**

Les services web permettent d'ignorer l'hétérogénéité entre les différentes applications. En effet, ils décrivent comment transmettre un message (standardisé) entre deux applications, sans imposer comment construire ce message.

- **Auto-descriptivité**

Les services web ont la particularité d'être auto-descriptifs, c'est-à-dire capables de fournir des informations permettant de comprendre comment les manipuler. La capacité des services à se décrire par eux-mêmes permet d'envisager l'automatisation de l'intégration de services.

- **Les utilitaires de développement**

S'appuyant sur ces standards, permettent la création automatique de programmes utilisant les services Web existants. [13]

2.3.11.2. Les inconvénients

1. Les services Web ont de faibles performances par rapport aux autres approches de l'informatique répartie telle que RMI, CORBA, DCOM.

2. En l'utilisation de protocole HTTP, les services Web peuvent contourner les mesures de sécurité mis en place à travers les firewalls.

3. La sémantique n'est pas prise en charge de façon efficace car le WSDL décrit les services de manière syntaxique.

4. Les transferts reposent sur le XML, ce qui pose un problème sur la taille des fichiers échangés. Les fichiers XML sont le plus souvent de très gros fichiers et ceci entrainera une lourdeur considérable.

5. Ils ne sont pas sécurisés à 100%.

Conclusion

Les services web sont des applications accessibles par l'échange de documents XML entre deux URL. Ils permettent une souplesse d'utilisation, et une accélération du développement d'applications. En adoptant cette technologie, nous faciliterons l'interopérabilité et la réutilisation du code.

A decorative graphic of a scroll with a green outline and grey rollers at the top and bottom. The text is centered within a white rectangular box in the middle of the scroll.

Partie 2.
Méthodologie
2TUPprojet

Introduction

La complexité croissante des systèmes informatiques a conduit les concepteurs à s'intéresser aux méthodes, aux processus à suivre durant la réalisation des projets ainsi la démarche et une méthode précise pour la réalisation.

3.1. Processus de développement logiciel

Un processus définit une séquence d'étapes, en partie ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution 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. En conséquence, le processus peut se décomposer suivant deux axes de contrôle sur le développement :

- L'axe de développement technique, qui se concentre principalement sur la qualité de la production ;
- L'axe de gestion du développement, qui permet la mesure et la prévision des coûts et des délais. [1]

3.1.1. Processus unifié (unified process)

Le Processus Unifié ou UP (*Unified Process*) est une méthode générique de développement de logiciel développée par les concepteurs d'UML

➤ **Générique** signifie qu'il est nécessaire d'adapter UP au contexte du projet, de l'équipe, du domaine et/ou de l'organisation. Il existe donc un certain nombre de méthodes issues d'UP comme par exemple RUP (Rational Unified Process), 2TUP (Two Track Unified Process).

Il existe d'autres approches (qui ne sont en général pas complètement antinomique), comme par ex. les méthodes « agile », beaucoup moins orientées modèle, comme XP (eXtreme Programming) [1]

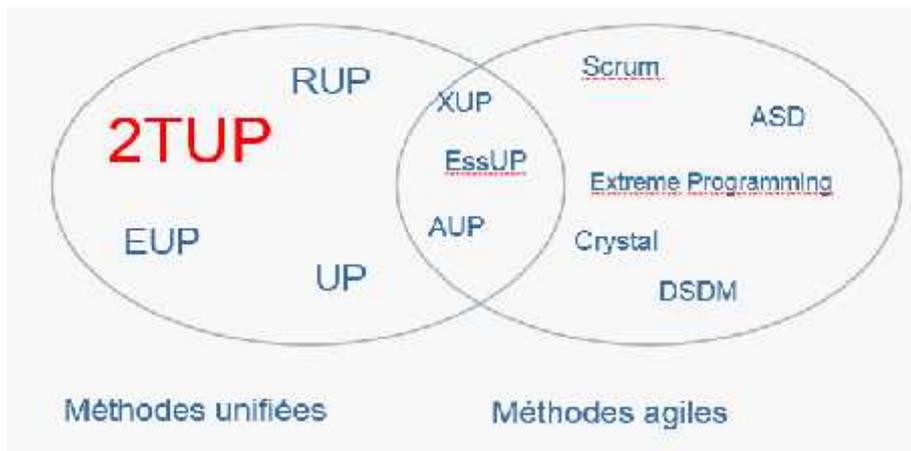


Figure 1. Les différentes méthodes de conception.

➤ **Il est itératif et incrémental** La définition d'itérations de réalisation est en effet la meilleure pratique de gestion des risques d'ordre à la fois technique et fonctionnel.

➤ **Itératif** : Le logiciel nécessite une compréhension progressive du problème à travers des raffinements successifs et développer une solution effective de façon incrémentale par des itérations multiples.



Figure 2. Un processus itératif et incrémental

➤ **Il est piloté par les risques** Dans ce cadre, les causes majeures d'échec d'un projet logiciel doivent être écartées en priorité.

➤ **Centré sur l'architecture** Dès le démarrage du processus, on aura une vue sur l'architecture à mettre en place. L'architecture d'un système logiciel peut être décrite comme les différentes vues du système qui doit être construit. [1]

➤ **Conduit par les cas d'utilisation** : le processus est orienté par les besoins utilisateurs présentés par des cas d'utilisation.

3.2. Avantages d'un processus itératif contrôlé

➤ Permet de limiter les coûts, en termes de risques, aux strictes dépenses liées à une itération.

➤ Permet de limiter les risques de retard de mise sur le marché du produit développé (identification des problèmes dès les premiers stades de développement et non en phase de test comme avec l'approche « classique »).

➤ Permet d'accélérer le rythme de développement grâce à des objectifs clairs et à court terme.

➤ Permet de prendre en compte le fait que les besoins des utilisateurs et les exigences correspondantes ne peuvent être intégralement définis à l'avance et se dégagent peu à peu des itérations successives.

La gestion d'un tel processus est organisée d'après les 4 phases suivantes :

a. Pré-étude (Inception) : c'est ici qu'on évalue la valeur ajoutée du développement et la capacité technique à le réaliser (étude de faisabilité).

b. Elaboration : sert à confirmer l'adéquation du système aux besoins des utilisateurs et à livrer l'architecture de base.

c. Construction : sert à livrer progressivement toutes les fonctions du système.

d. Transition : déployer le système sur des sites opérationnels.

3.3. Le processus 2TUP

2TUP signifie « 2 Track Unified Process ». C'est un processus UP qui répond aux caractéristiques que nous venons de citer. Le processus 2TUP apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes. « 2 Track » signifie littéralement que le processus suit deux chemins. Il s'agit des chemins « fonctionnels » et « d'architecture technique », qui correspondent aux deux axes de changement imposés au système informatique.



Figure 3. Le système d'information soumis à deux natures de contraintes.

3.3.1 Les Diverses phases de 2TUP

3.3.1.1. Etude préliminaire

Elle constitue généralement le cahier de charge.

➤ Cahier de charge

Le cahier des charges se définit comme un acte, un document de référence qui permet à un dirigeant d'entreprise de préciser les conditions, les règles et les exigences d'un travail à accomplir, en vue de résoudre un problème spécifique ou d'améliorer une situation donnée tout en déterminant les résultats attendus.

Les étapes de l'élaboration du cahier de charge

✓ **Identifier des acteurs** : ce sont des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

✓ **Identifier les messages** : un message représente la spécification d'une communication unidirectionnelle dont le but est de déclencher une activité chez le récepteur.

✓ **Modéliser le contexte** : tous les messages identifiés dans l'étape précédemment peuvent être représentés sur un diagramme, que l'on qualifie de diagramme de contexte dynamique.

3.3.1.2. Branche gauche (fonctionnels)

Cette branche comporte les étapes suivantes :

➤ **Capture des besoins fonctionnels :**

Produit un modèle des besoins focalisé sur le métier des utilisateurs.

➤ **Analyse :**

Qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en terme de métier.

3.3.1. 3. La branche droite (architecture technique)

Capitalise un savoir-faire technique. Elle constitue un investissement pour le court et moyen terme. Cette branche comporte les étapes suivantes :

➤ **Capture des besoins techniques :**

C'est Les outils, le matériel sélectionné ainsi que les contraintes d'intégration avec l'existant.

➤ **Conception générique :**

Définit les composants nécessaires à la construction de l'architecture technique.

3.3.1.4. La branche du milieu

➤ **Conception préliminaire :**

Intègre le modèle d'analyse dans l'architecture technique de manière à tracer la cartographie des composants du système à développer.

➤ **Conception détaillée :**

Etudie comment réaliser chaque composant.

➤ **Codage et tests :**

Produit au fur et à mesure les composants et les unités de code du système.

➤ **Recette :**

Consiste à valider les fonctions du système développé.

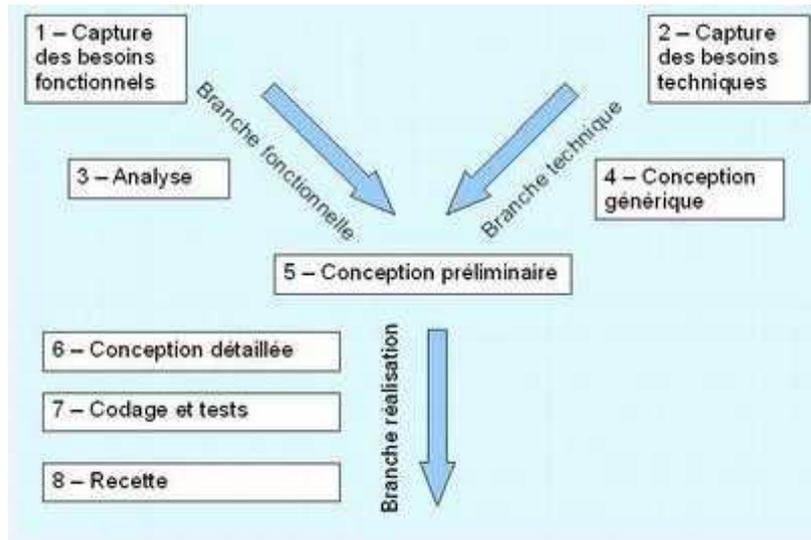


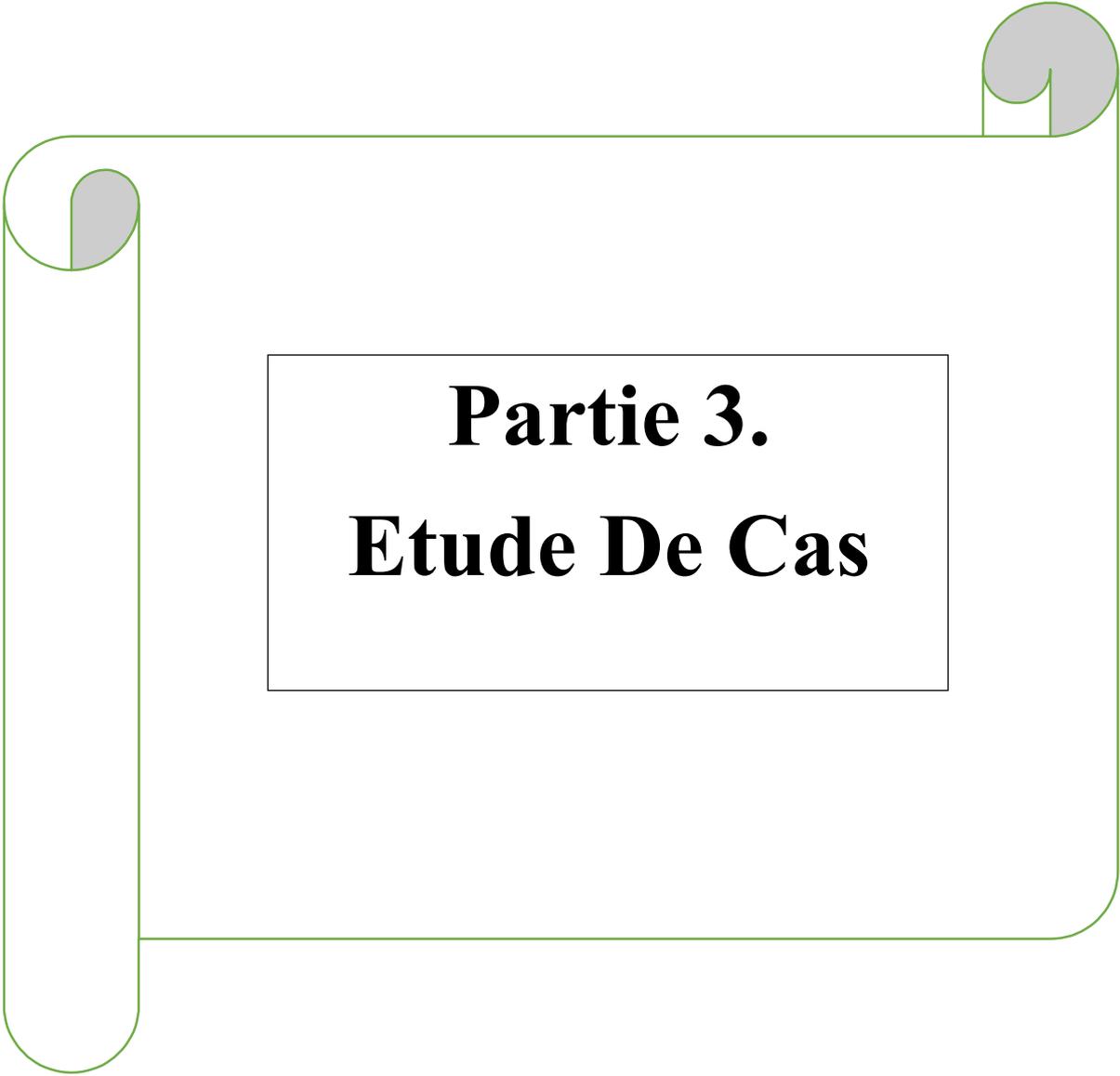
Figure 4. processus de développement en Y.

- **Les points forts de 2TUP**
 - ✓ Itératif
 - ✓ Fait une large place à la technologie et à la gestion du risque
 - ✓ Définit les profils des intervenants, les livrables, les plannings, les prototypes
- **Les points faibles de 2TUP**
 - ✓ Plutôt superficiel sur les phases situées en amont et en aval du développement : capture des besoins, support, maintenance, gestion du changement...
 - ✓ Ne propose pas de documents types.

Conclusion

Le processus de développement 2TUP constitue une meilleure pratique de développement en insistant sur le non corrélation initial des aspects fonctionnels et techniques. Il est construit autour de la construction et de maintien d'un modèle, qui permet la conception d'un système, en tenant compte les étapes de développement et les points de vue de modélisation préconisée.

Dans le prochain chapitre, on va faire la conception de notre application, on utilisant UML comme langage de modélisation et 2TUP comme processus de développement.

A decorative border resembling a scroll, with a green outline and grey circular accents at the corners and along the left edge.

Partie 3.
Etude De Cas

Introduction

L'étude préliminaire ou (pré étude) est la première étape de notre processus de développement. Elle consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels, en utilisant principalement le texte ou des diagrammes très simples. Elle prépare les étapes plus formelles de capture des besoins fonctionnels et capture des besoins techniques.

Dans un premier temps, nous allons introduire l'étude de cas qui servira de fil conducteur tout au long du développement de notre application en donnant une version textuelle préliminaire du cahier des charges. Dans un second temps, nous commencerons par détermination des besoins fonctionnels en considérant le système comme une boîte noire, afin d'étudier sa place dans le système. Après avoir identifié les acteurs qui interagissent avec le système, nous développerons un premier modèle UML de niveau contacte, pour pouvoir établir précisément les fonctionnelles du système.

4.1. L'élaboration du cahier de charges :

- ✓ Les choix techniques.
- ✓ La présentation du projet.
- ✓ Le recueil des besoins fonctionnels.
- ✓ Identification des messages.
- ✓ Identifications des acteurs.
- ✓ Modélisation du contexte.

4.1.1. Présentation du projet

Le but de ce projet est la conception et la réalisation d'une application à base de web services pour les régimes alimentaires.

4.1.2. Choix techniques

Nous souhaitons utiliser une approche itérative et incrémentale, fondée sur le processus en Y (méthode 2TUP) qui utilise le langage standard de modélisation UML, nous avons affranchi le choix d'un certain nombre de techniques clés pour ce projet :

- ✓ JAVA pour la programmation de la logique applicative de ce projet dans une plateforme Eclipse, Dream weaver, adobe photo shop.
- ✓ Oracle12, MySQL pour le stockage et la gestion des données.

4.1.3. Recueil des besoins fonctionnels

Nous avons effectué une recherche pour identifier les besoins de l'application, et ceci afin de répondre à la fois aux besoins des fournisseurs des régimes alimentaires et des utilisateurs finaux des régimes alimentaires.

4.1.3.1. Les besoins fonctionnels au niveau des fournisseurs des régimes alimentaires

Les fournisseurs des régimes alimentaires sont des nutritionnistes (ou par fois des médecins), ils font les actions suivantes :

a. Ajouter un régime alimentaire

Selon les critères de l'état de santé, l'âge, ...

b. Modifier un régime alimentaire

S'il est nécessaire après une évolution introduite dans le domaine.

c. Supprimer un régime alimentaire

S'il est nécessaire après une erreur qui ne peut pas corriger, ou bien par l'introduction des évolutions dans le domaine qui change leurs connaissances radicalement.

d. Consulter un régime alimentaire

Dont le but est de consulter un ou plusieurs régimes.

e. Envoyer un régime alimentaire

Vers l'utilisateur des régimes alimentaire.

4.1.3.2. Les besoins fonctionnels au niveau des utilisateurs des régimes alimentaires

a. Demander un régime alimentaire

A travers un site web.

4.1.4. Recueil des besoins opérationnels

Une application à base de web services doit respecter l'architecture des services web. Ainsi le système global est divisé en trois sous-systèmes, selon cette architecture nous avons : Un système pour les fournisseurs des régimes alimentaires, le système de l'UDDI (Annuaire de service), et un système pour l'utilisateur des régimes alimentaires.

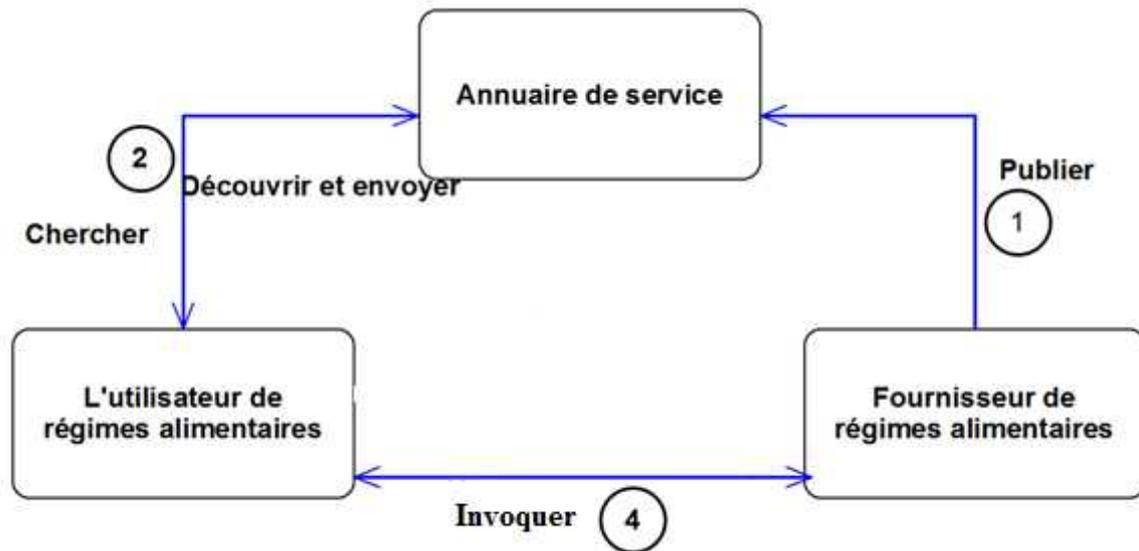


Figure 1. L'architecture d'un service web

Ainsi l'application doit assurer certaines qualités :

- ✓ **Sécurité** : chaque utilisateur de l'application doit être authentifié.
- ✓ **Temps de réponse** : doit être acceptable.
- ✓ **Interface graphique** : doit être simple à utiliser et convivial.

4.2. Description du contexte

4.2.1. Identifications des acteurs

Dans la suite nous présenterons les différents acteurs qui interagissent avec le système.

4.2.1.1. Au niveau des fournisseurs des régimes alimentaires

A ce niveau nous avons une application avec deux acteurs, l'administrateur des fournisseurs des régimes alimentaires et le fournisseur des régimes alimentaires.

a. Administrateur des fournisseurs des régimes alimentaires

Cet acteur accède de manière sécurisée au ce système. Il peut :

- Ajouter, Consulter, Modifier, et Supprimer un fournisseur des régimes alimentaires.

b. Fournisseur des régimes alimentaires

Cet acteur accède de manière sécurisée au ce système. Il peut :

- Ajouter, Consulter, Modifier et Supprimer un régime alimentaire, un aliment, une région, une religion.

4.2.1.2. Au niveau de l'utilisateur des régimes alimentaires

On suppose que l'utilisateur des régimes alimentaires ou dans le terme des services web « le client de service » est un site web des régimes alimentaires.

A ce niveau nous avons deux acteurs : l'administrateur de site web des régimes alimentaires et l'utilisateur final du site web des régimes alimentaires (qui peut accéder au système soit comme un client ou un visiteur de site web).

a. L'administrateur de site web des régimes alimentaires

Cet acteur accède de manière sécurisée à ce système. Il peut

- Modifier, et Supprimer le compte du client.
- Ajouter, Modifier, et Supprimer un fichier.
- Ajouter les annonces.
- Consulter les fichiers, les comptes clients, les annonces.

b. L'utilisateur final du site web des régimes alimentaires

L'interaction avec le site web ce fait par l'intervention de deux acteurs :

b1. Le client de site web des régimes alimentaires

Cet acteur accède de manière sécurisée au ce système. Il peut :

- Rechercher un régime alimentaire.
- Invoquer un régime alimentaire.
- Consulter la page d'accueil, les annonces.

b2. Le visiteur de site web des régimes alimentaires

Il peut :

- Consulter la page d'accueil, les annonces.
- Inscrire et deviendra ensuite un client du site web.

4.2.1.3. Au niveau de l'UDDI

A ce niveau après la publication de régime l'UDDI peut :

- Enregistrer les informations concernant un certain régime alimentaire.

Et après une recherche lancée par le client de site web, il peut :

- Découvrir puis envoyer les informations de régime demandé.

4.2.2. Identifications des messages

Notre système est une collection de trois systèmes : le premier au niveau de fournisseur des régimes alimentaires, le deuxième au niveau de site web des régimes alimentaires et le dernier au niveau de l'UDDI. Le système interagit avec les différents acteurs à l'aide des messages échangés entre eux.

4. 2.2. 1. Au niveau de fournisseur des régimes alimentaires

➤ Le système émis les messages suivants

- Notification de l'ajout, la modification, la suppression des fournisseurs des régimes alimentaires.
- Donner la liste des fournisseurs des régimes alimentaires, des aliments, des religions, des régions.
- Notification de l'ajout, la modification, la suppression des régimes alimentaires, des aliments, des religions, des régions.
- Donner la liste des régimes alimentaires.

➤ Le système reçoit les messages suivants

- Ajouter (), Consulter (), Modifier (), Supprimer (), les fournisseurs des régimes alimentaires.
- Ajouter (), Consulter (), Modifier (), Supprimer (), les régimes alimentaires, des aliments, des religions, des régions.
- La validation des différentes demandes.
- Invoquer () un régime alimentaire [ce message est reçu à partir de client de site web des régimes alimentaires].

4. 2.2. 2. Au niveau de l'utilisateur des régimes alimentaires

➤ Le système émis les messages suivants

- Notification de la modification, la suppression des clients.
- La liste des clients.
- Notification de l'ajout des annonces.

- La liste des annonces, fichiers.
- Notification de l’ajout, la modification, la suppression des fichiers.

➤ **Le système reçoit les messages suivants**

- Inscrire ()
- consulter (), modifier (), supprimer () les comptes des clients de site web.
- Ajouter (), consulter (), les annonces.
- Ajouter (), modifier (), supprimer (), consulter (), les fichiers.
- Envoyer () un régime alimentaire.

4. 2.2. 3. Au niveau de l’UDDI

➤ **Le système émis les messages suivants**

- Envoyer les informations concernant un régime demandé.

•**Le système reçoit les messages suivants**

- Rechercher (). [Le client de site web fait une recherche sur les informations d’un régime alimentaire].

Tous les messages émetts et reçus sont classés dans la table suivante

	Les messages émis	Les messages reçus
Au niveau de fournisseur des régimes alimentaires	E1.La notification de l’ajout de fournisseur. E2. La liste des fournisseurs. E3.La notification de la modification de fournisseur. E4.La notification de la suppression de fournisseur. E5. notification de l’ajout de régime. E6. La liste des régimes. E7.La notification de la modification de régime. E8. La notification de la suppression de régime.	R1. Ajouter_fournisseur () R2. Consulter_fournisseur () R3.Modifier_fournisseur () R4.Supprimer_fournisseur () R5. Ajouter_régime () R6. Consulter_régimes () R7.Modifier_régime () R8.Supprimer_régime () R9. Invoquer_régime () R10. Ajouter_aliment () R11. Modifier_aliment () R12. Supprimer_aliment () R13. Ajouter_région () R14. Modifier_région ()

	<p>E9. Le régime demandé</p> <p>E10. notification de l'ajout de l'aliment.</p> <p>E11. La notification de la modification de l'aliment.</p> <p>E12. La notification de la suppression de l'aliment.</p> <p>E13. notification de l'ajout de la région.</p> <p>E14. La notification de la modification de la région.</p> <p>E15. La notification de la suppression de la région.</p> <p>E16. notification de l'ajout de la religion.</p> <p>E17. La notification de la modification l'ajout de la religion.</p> <p>E18. La notification de la suppression l'ajout de la religion.</p> <p>E19. La liste des de aliments.</p> <p>E20. La liste des régions.</p> <p>E21. La liste des religions.</p>	<p>R15. Supprimer_région ()</p> <p>R16. Ajouter_religion ()</p> <p>R17. Modifier_religion ()</p> <p>R18. Supprimer_religion ()</p> <p>R19. Consulter_aliment ()</p> <p>R20. Consulter_région ()</p> <p>R21. Consulter_religion ()</p>
<p>Au niveau de l'utilisateur des régimes alimentaires</p>	<p>E22. La notification l'inscription.</p> <p>E23. La liste des clients.</p> <p>E24. La notification de la modification d'un compte client.</p> <p>E25. La notification de la suppression d'un compte client.</p>	<p>R22.inscrire ().</p> <p>R23. Consulter_compte_clients ().</p> <p>R24.Modifier_compte_client ().</p> <p>R25. Supprimer_compte_client ().</p> <p>R26. Ajouter_fichier ().</p> <p>R27. Consulter_les_fichiers ().</p> <p>R28.Modifier_fichier ().</p>

	<p>E26. La notification de l'ajout d'un fichier.</p> <p>E27. La liste des fichiers.</p> <p>E28. La notification de la modification d'un fichier.</p> <p>E29. La notification de la suppression d'un fichier.</p> <p>E30. La notification de l'ajout d'une annonce.</p> <p>E31. La liste des annonces.</p> <p>E32. Les informations existent dans la page d'accueil.</p>	<p>R29. Supprimer_fichier ().</p> <p>R30. Ajouter_annonce ().</p> <p>R31. Consulter_annonces ().</p> <p>R32. Consulter_la_page_d'accueil ().</p>
<p>Au niveau de l'UDDI</p>	<p>E33. Les informations de régime.</p>	<p>R33. Rechercher_régime ()</p>

Tableau 1. Modélisation de contexte de notre système

4.2.3. Diagramme de contexte dynamique

Un diagramme de contexte dynamique représente les messages échangés entre l'application et les acteurs identifiés (représentés de façon synthétique sur un diagramme) chaque numéro correspond à un message dans la table des messages précédente.

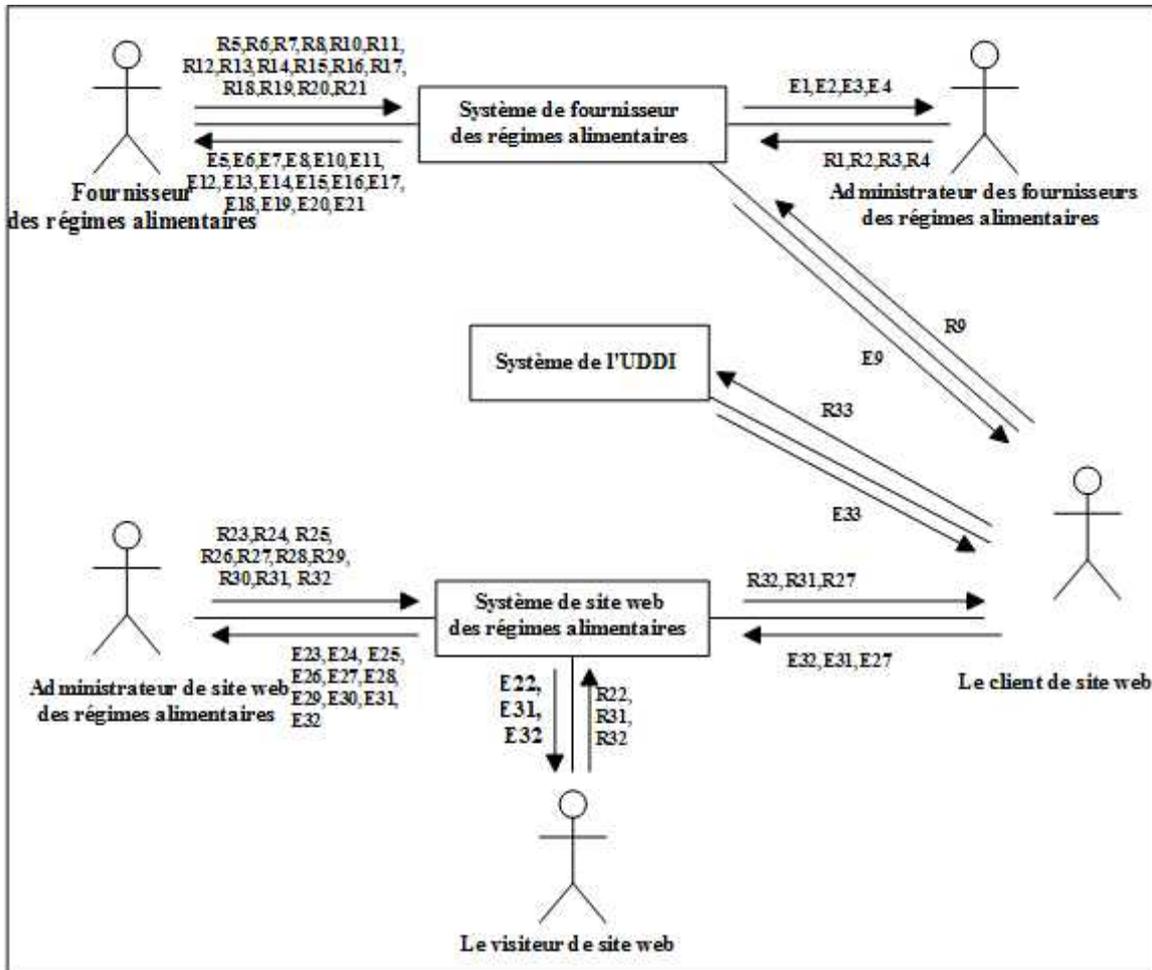


Figure 2. Diagramme de contexte dynamique.

4.3. Modélisation du contexte

A partir des informations obtenues lors des deux précédentes étapes, nous allons modéliser le contexte de notre application. Ceci va nous permettre dans un premier temps, de définir le rôle de chaque acteur dans le système :

	Utilisateur finaux	Description des besoins fonctionnels
Au niveau de fournisseur des régimes alimentaires	L'administrateur des fournisseurs des régimes alimentaires	L'application doit permettre à l'administrateur des fournisseurs des régimes alimentaires de : <ul style="list-style-type: none"> • S'authentifier. • Ajouter, Consulter, Modifier, et Supprimer un fournisseur des régimes alimentaires.
	Le fournisseur des régimes alimentaires	L'application doit permettre au fournisseur des régimes alimentaires de : <ul style="list-style-type: none"> • S'authentifier. • Ajouter, Consulter, Modifier, et Supprimer un régime alimentaire, un aliment, une région, une religion.
Au niveau de l'utilisateur des régimes alimentaires	L'administrateur de site web des régimes alimentaires	L'application doit permettre à l'administrateur de site web des régimes alimentaires de : <ul style="list-style-type: none"> • S'authentifier. • Ajouter, Modifier, et Supprimer un fichier. • Modifier, et Supprimer un compte client. • Ajouter les annonces. • Consulter les fichiers, les clients et les

		annonces.
	Le client de site web	<p>L'application doit permettre au client du site web des régimes alimentaires de :</p> <ul style="list-style-type: none"> • S'authentifier. • Invoquer un régime alimentaire. • Rechercher un régime alimentaire. • Consulter la page d'accueil, les annonces et les fichiers.
	Le visiteur de site web	<ul style="list-style-type: none"> • Consulter la page d'accueil et les annonces. • Inscrire et deviendra ensuite un client du site web.

Tableau 2. Gérer le rôle des acteurs

Conclusion

Après avoir dégagé les besoins fonctionnels et opérationnels et tous les critères qu'on doit prendre en considération, dans le prochain chapitre nous allons poursuivre la formalisation de ces besoins.

Introduction

Cette phase est la première étape de la branche gauche du cycle en Y. Elle représente un point de vue « fonctionnelle » de l'architecture système. Et pour ce faire nous utiliserons la notion d'Use Case. Chaque Use Case sera identifié, décrit, et organisé, classé en fonction de son importance dans le projet.

5.1. Identification des cas d'utilisation

Cas d'utilisation	Les acteurs	Les messages
Ajouter, Modifier, Supprimer.	L'administrateur des fournisseurs des régimes alimentaires	Emet : Ajouter (), Modifier (), Supprimer (), un fournisseur des régimes alimentaires. Reçoit : notification de l'ajout, la modification, la suppression des fournisseurs des régimes alimentaires.
Consulter		Emet : Consulter (), les fournisseurs des régimes alimentaires. Reçoit : La liste des fournisseurs des régimes alimentaires.
Ajouter, Modifier, Supprimer.	Le fournisseur des régimes alimentaires	Emet : Ajouter (), Modifier (), Supprimer (), un régime alimentaire, un aliment, une région, une religion. Reçoit : notification de l'ajout, la modification, la suppression d'un régime alimentaire, un aliment, une région, une religion.
Consulter		Emet : Consulter () les régimes alimentaires, les aliments, les régions, les religions.

		<p>Reçoit : La liste des régimes alimentaires, des aliments, des régions, des religions.</p>
Ajouter	Administrateur de site web des régimes alimentaires	<p>Emet : Ajouter () une annonce, un fichier.</p> <p>Reçoit : notification de l'ajout d'une annonce, d'un fichier.</p>
Modifier, Supprimer		<p>Emet : Modifier (), Supprimer () un compte de client de site web, un fichier.</p> <p>Reçoit : notification de la modification, la suppression d'un compte client de site web, un fichier, un client.</p>
Consulter		<p>Emet : Consulter (), les clients de site web, les annonces, les fichiers.</p> <p>Reçoit : La liste des clients de site web, des annonces, des fichiers.</p>
Consulter	Le client de site web	<p>Emet : Consulter (), les annonces, la page d'accueil.</p> <p>Reçoit : La liste des annonces, les informations existent dans la page d'accueil.</p>
Rechercher		<p>Emet : Rechercher (), un régime alimentaire. [S'effectue au niveau de l'UDDI].</p> <p>Reçoit : Les informations de régime</p>

		recherché. [Envoyées par l'UDDI]
Invoquer		<p>Emet : Invoquer (), un régime alimentaire. [S'effectue au niveau de fournisseur]</p> <p>Reçoit : Le régime demandé. [Envoyé par le fournisseur].</p>
Inscrire	Le visiteur de site web.	<p>Emet : Inscrire (), un visiteur de site web.</p> <p>Reçoit : Notification de l'inscription.</p>
Consulter		<p>Emet : Consulter (), la page d'accueil, les annonces.</p> <p>Reçoit : Les informations dans la page d'accueil, les annonces.</p>

Tableau 1. Liste des acteurs et des messages par cas d'utilisation de notre système

5.2. Le diagramme de cas d'utilisation pour les besoins fonctionnels

Après identification des cas d'utilisation et leurs acteurs, nous allons les représenter graphiquement sur un diagramme de cas d'utilisation qui représente le fonctionnement du système vis-à-vis de l'utilisateur :

5.2.1. Au niveau de fournisseur des régimes alimentaires

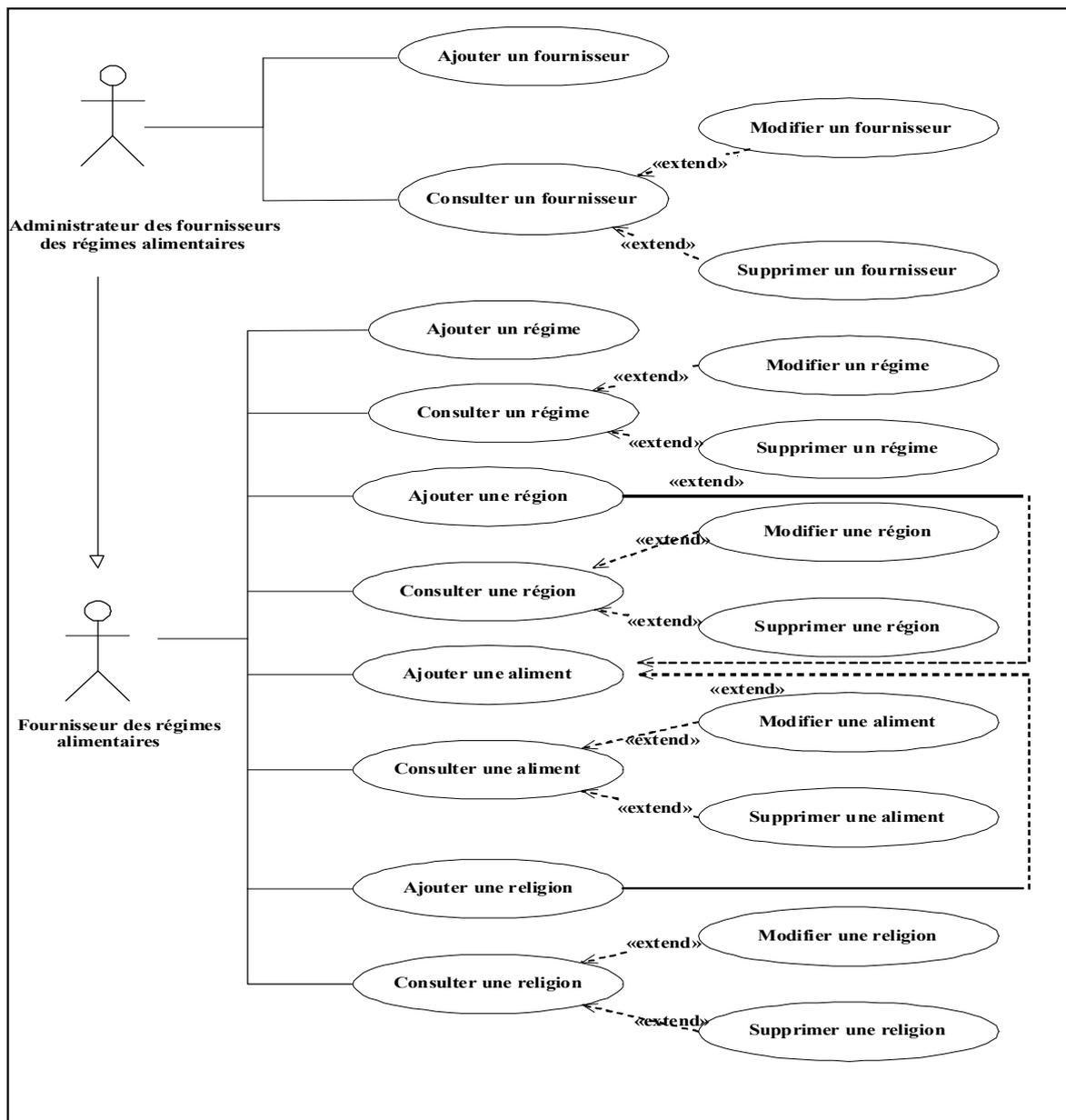


Figure 1. Diagramme de cas d'utilisation Au niveau de fournisseur des régimes alimentaire.

5.2.2. Au niveau de l'utilisateur des régimes alimentaires

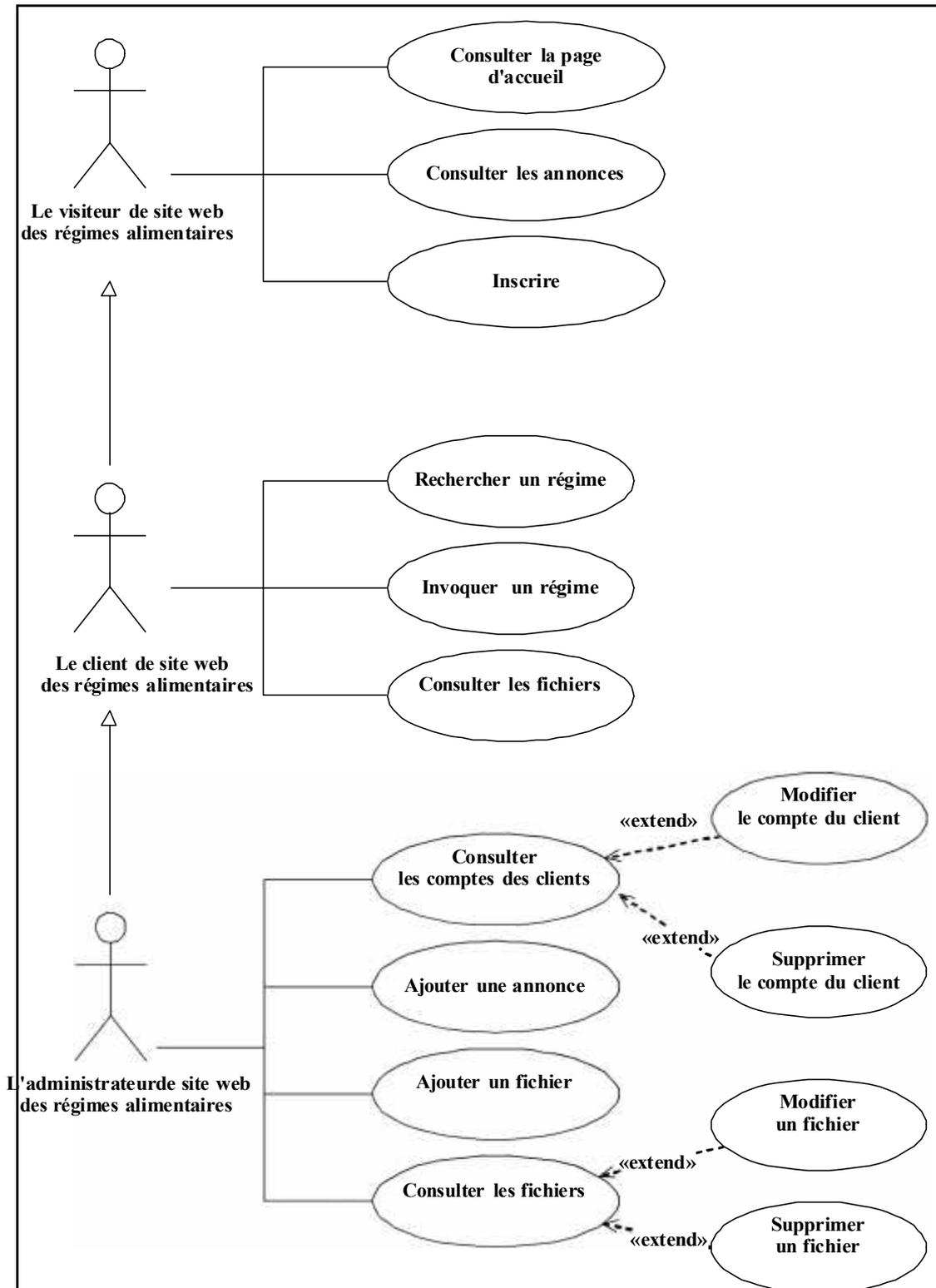


Figure 2. Diagramme de cas d'utilisation Au niveau de l'utilisateur des régimes alimentaires

5.3. Description détaillée des différents cas d'utilisation

5.3.1. Au niveau de fournisseur des régimes alimentaires

5.3.1.1. Cas d'utilisation « Ajouter un fournisseur des régimes alimentaires »

a. Fiche description

<p>Sommaire d'identification</p>
<p>Titre : Ajouter_fournisseur.</p> <p>But : Ajouter un fournisseur des régimes alimentaires.</p> <p>Résumé : Ajouter un fournisseur des régimes alimentaires dans la liste des utilisateurs de système.</p> <p>Acteur : L'administrateur des fournisseurs des régimes alimentaires.</p>
<p>Description des Enchaînements</p>
<p>Pré conditions :</p> <ul style="list-style-type: none"> • L'administrateur des fournisseurs des régimes alimentaires doit être authentifié. <p>Post-condition :</p> <ul style="list-style-type: none"> • L'ajout est effectué. <p>Scénario nominal :</p> <p>Ce cas d'utilisation commence lorsque l'administrateur des fournisseurs des régimes alimentaires demande au système de fournisseur des régimes alimentaires d'ajouter un fournisseur.</p> <ol style="list-style-type: none"> 1. le système affiche un formulaire d'ajout. 2. l'administrateur des fournisseurs des régimes alimentaires remplit toutes les informations. 3. Le système vérifie les informations saisies par l'administrateur des fournisseurs des régimes alimentaires puis demande une confirmation si les informations sont complètes et correctes. 4. l'administrateur des fournisseurs des régimes alimentaires confirme l'ajout.

5. Le système emmagasine le fournisseur et ces informations dans la base données.

6. Le système affiche une notification «Le fournisseur est ajouté».

Scénario alternatif

- Lorsque l'administrateur des fournisseurs des régimes alimentaires ne confirme pas l'ajout.
- Le système laisse la base des données avec son état primaire.
- Lorsque l'administrateur des fournisseurs introduit des informations incorrectes ou incomplètes.
- Le système affiche une notification «SVP vérifiez vos informations» et retourne à l'étape 1.

Scénario exceptionnel

- Lorsque l'administrateur des fournisseurs des régimes alimentaires choisit l'option annuler.
- Le système quitte la page d'ajout.

b. Diagramme d'activité

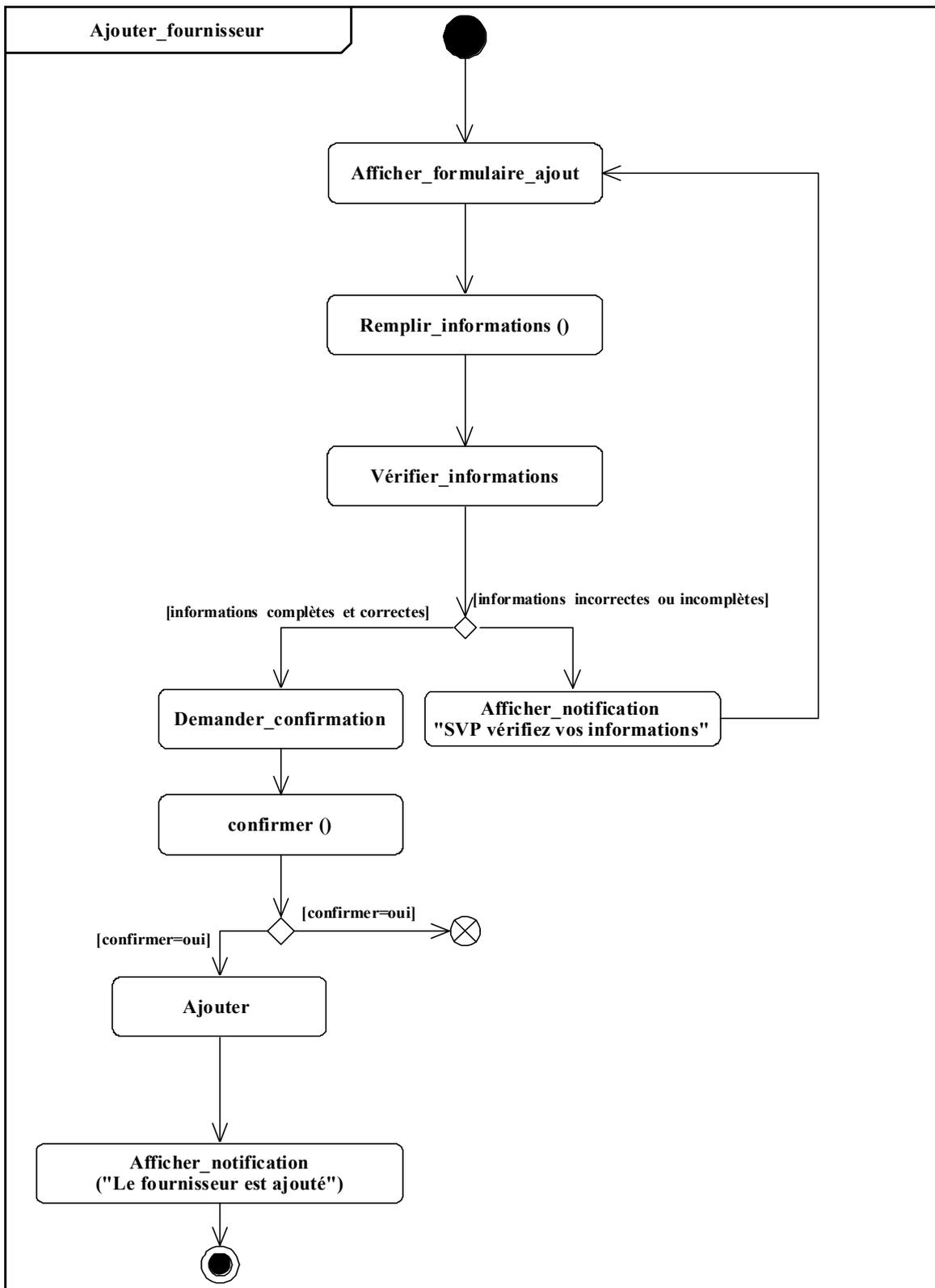


Figure 3. Diagramme d'activité cas «Ajouter fournisseur».

c. Diagramme de séquence

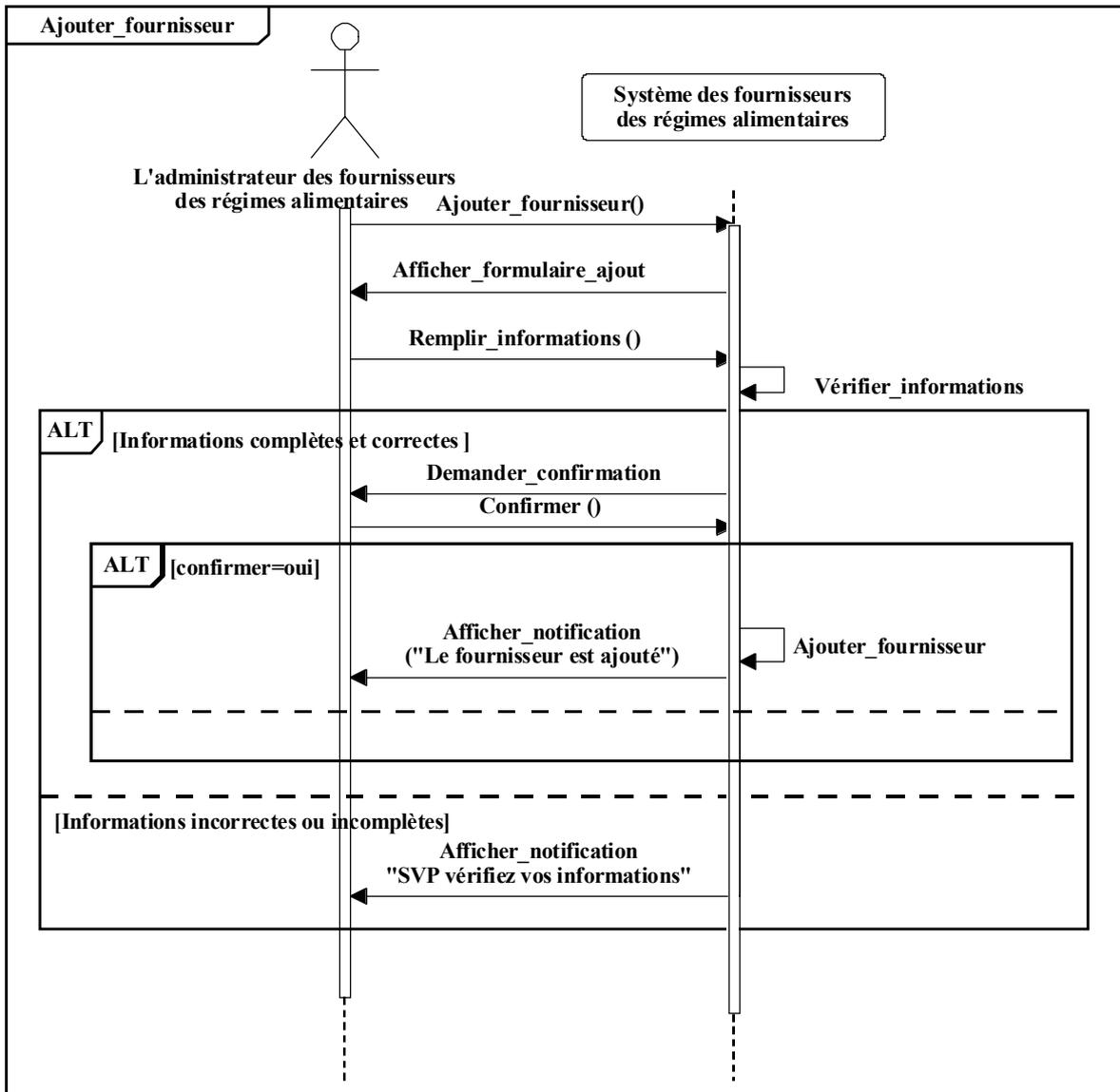


Figure 4. Diagramme de séquence système cas «Ajouter fournisseur».

5.3.1.2. Cas d'utilisation « Consulter les fournisseurs des régimes alimentaires »

a. Fiche description

Sommaire d'identification
Titre : Consulter_fournisseurs. But : Afficher les fournisseurs des régimes alimentaires existe dans la base de données. Résumé : Afficher la liste les fournisseurs des régimes alimentaires existe dans la base des données par conséquence leurs informations. Acteur : Administrateur de fournisseurs des régimes alimentaires.
Description des Enchaînements
Pré condition <ul style="list-style-type: none">• L'administrateur des fournisseurs des régimes alimentaires doit être s'authentifier. Post condition <ul style="list-style-type: none">• La liste des fournisseurs des régimes alimentaires est affichée. Scénario nominal <p>Ce cas d'utilisation commence lorsque l'administrateur des fournisseurs des régimes alimentaires demande au système de fournisseur des régimes alimentaires une consultation des fournisseurs des régimes alimentaires.</p> <ul style="list-style-type: none">• Le système affiche la liste des fournisseurs des régimes alimentaires. Scénario alternatif <ul style="list-style-type: none">➤ Lorsque le système ne parvient pas à accéder à la base des données.• Le système affiche notification « Impossible d'afficher le contenu de la base ».

b. Diagramme d'activité

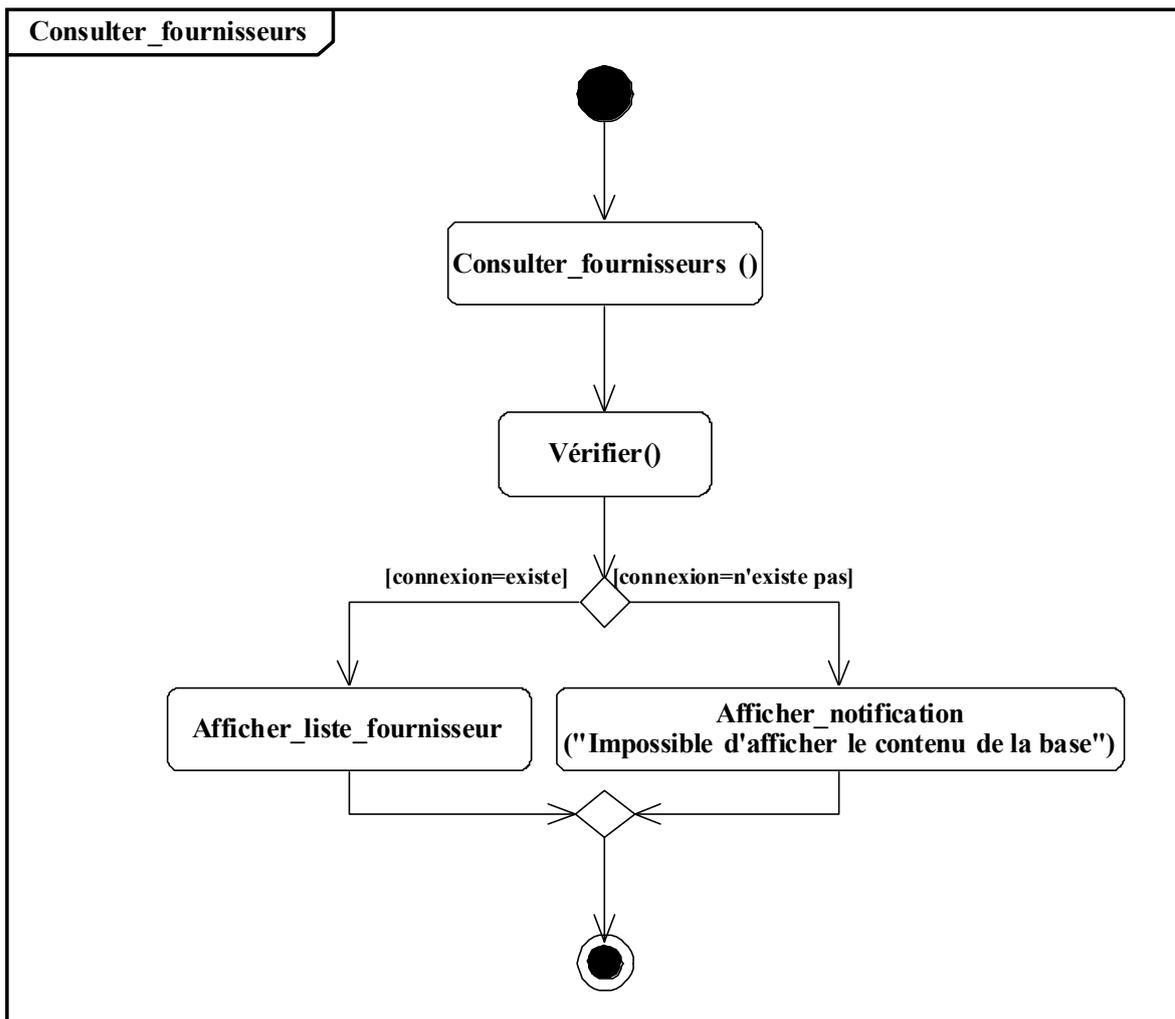


Figure 5. Diagramme d'activité cas «Consulter fournisseurs».

c. Diagramme de séquence

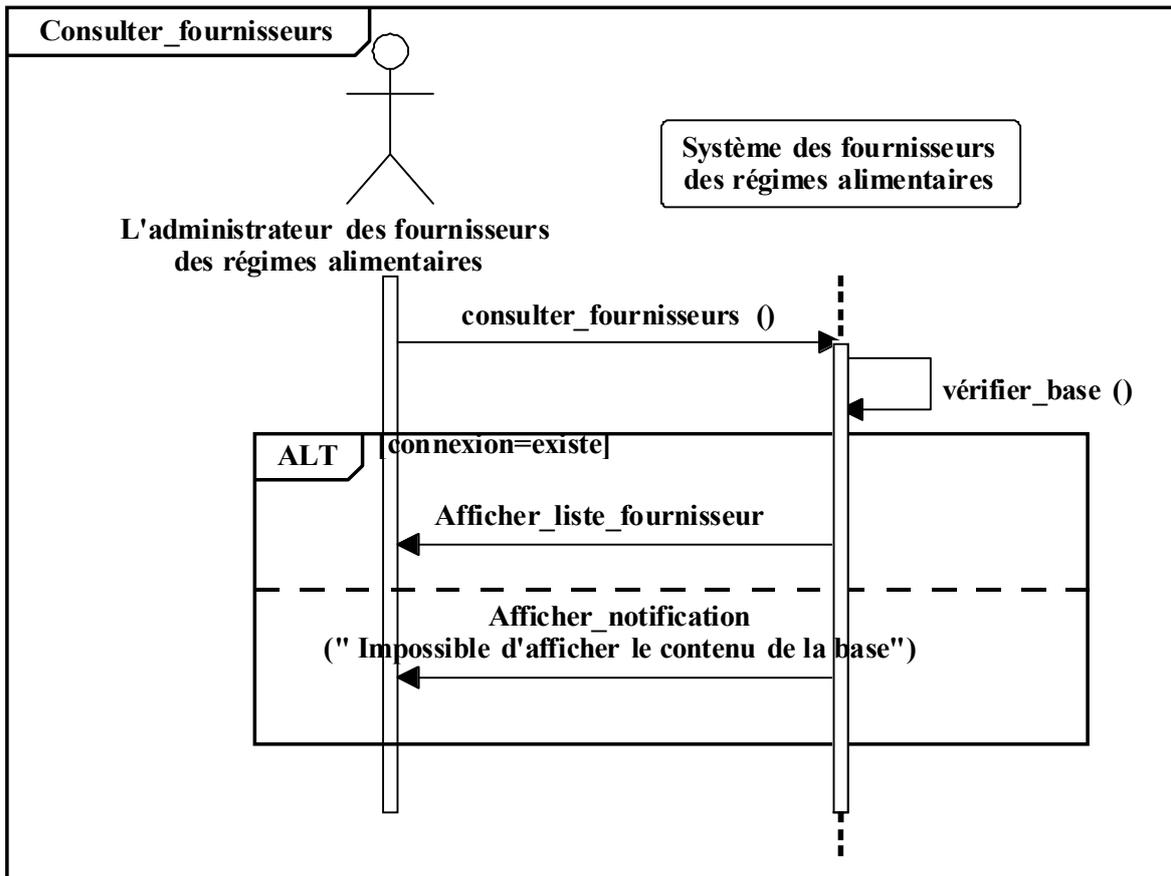


Figure 6. Diagramme de séquence système cas «Consulter fournisseurs».

5.3.1. 3. Cas d'utilisation « Modifier un fournisseur des régimes alimentaires »

a. Fiche description

Sommaire d'identification
Titre : Modifier_fournisseur. But : Modifier les informations d'un fournisseur des régimes alimentaires. Résumé : changer le profil d'un fournisseur des régimes alimentaires. Acteur : L'administrateur des fournisseurs des régimes alimentaires.
Des descriptions des enchainements
Pré conditions <ul style="list-style-type: none">• L'administrateur des fournisseurs des régimes alimentaires doit être authentifié.• Le fournisseur des régimes alimentaires existe déjà. Post conditions <ul style="list-style-type: none">• Le profil de fournisseur des régimes alimentaires est modifié. Scénario nominal <p>Ce cas d'utilisation commence lorsque l'administrateur des fournisseurs des régimes alimentaires demande au système de fournisseur des régimes alimentaires une modification d'un fournisseur des régimes alimentaires.</p> <ol style="list-style-type: none">1. L'administrateur des régimes alimentaires consulte la liste des fournisseurs des régimes alimentaires.2. Le système affiche la liste des fournisseurs des régimes alimentaires.3. L'administrateur des fournisseurs des régimes alimentaires choisit l'un des fournisseurs des régimes alimentaires.4. Le système affiche un formulaire pour changer les informations de ce fournisseur.5. L'administrateur des fournisseurs des régimes alimentaires modifie les champs qu'il veut dans le formulaire.6. Le système vérifie les informations saisis par l'administrateur des fournisseurs des

régimes alimentaires puis demande une confirmation de la modification.

7. L'administrateur des fournisseurs des régimes alimentaires confirme la modification.

8. Le système fait une mise à jour concernant ce fournisseur dans la base des données, puis il affiche notification «Le fournisseur est modifié».

Scénario alternatif

- Lorsque l'administrateur des fournisseurs des régimes alimentaires ne confirme pas la modification.
 - Le système laisse le fournisseur avec son état primaire.
- Lorsque l'administrateur des fournisseurs des régimes alimentaires remplit des informations incorrectes ou incomplètes.
 - Le système affiche une «SVP vérifiez vos informations» et retourne à l'étape 4.

Scénario exceptionnel :

- Lorsque l'administrateur des fournisseurs des régimes alimentaires choisit l'option annuler.
 - Le système quitte la page.

b. Diagramme d'activité

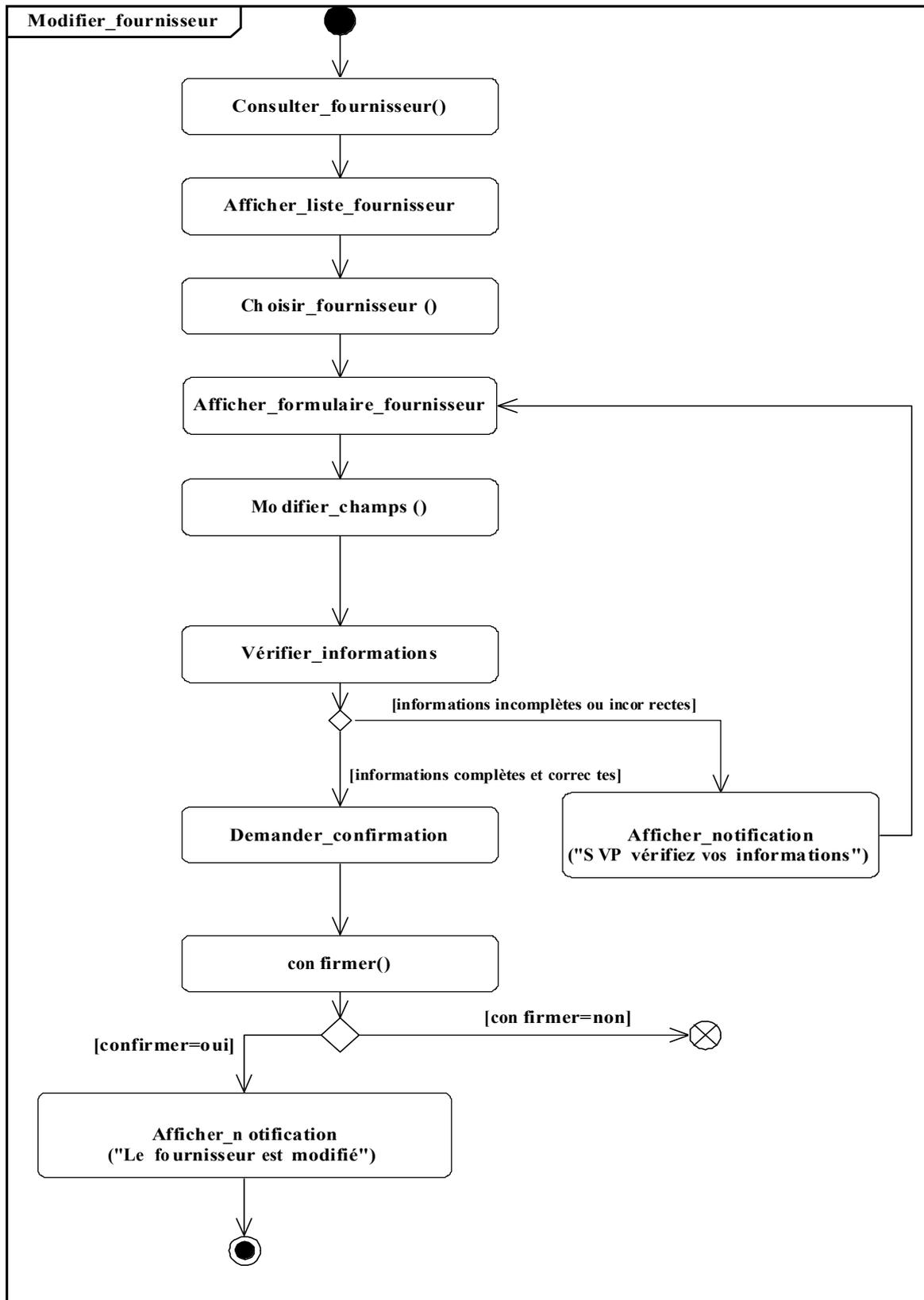


Figure 7. Diagramme d'activité cas «Modifier fournisseur».

c. Diagramme de séquence

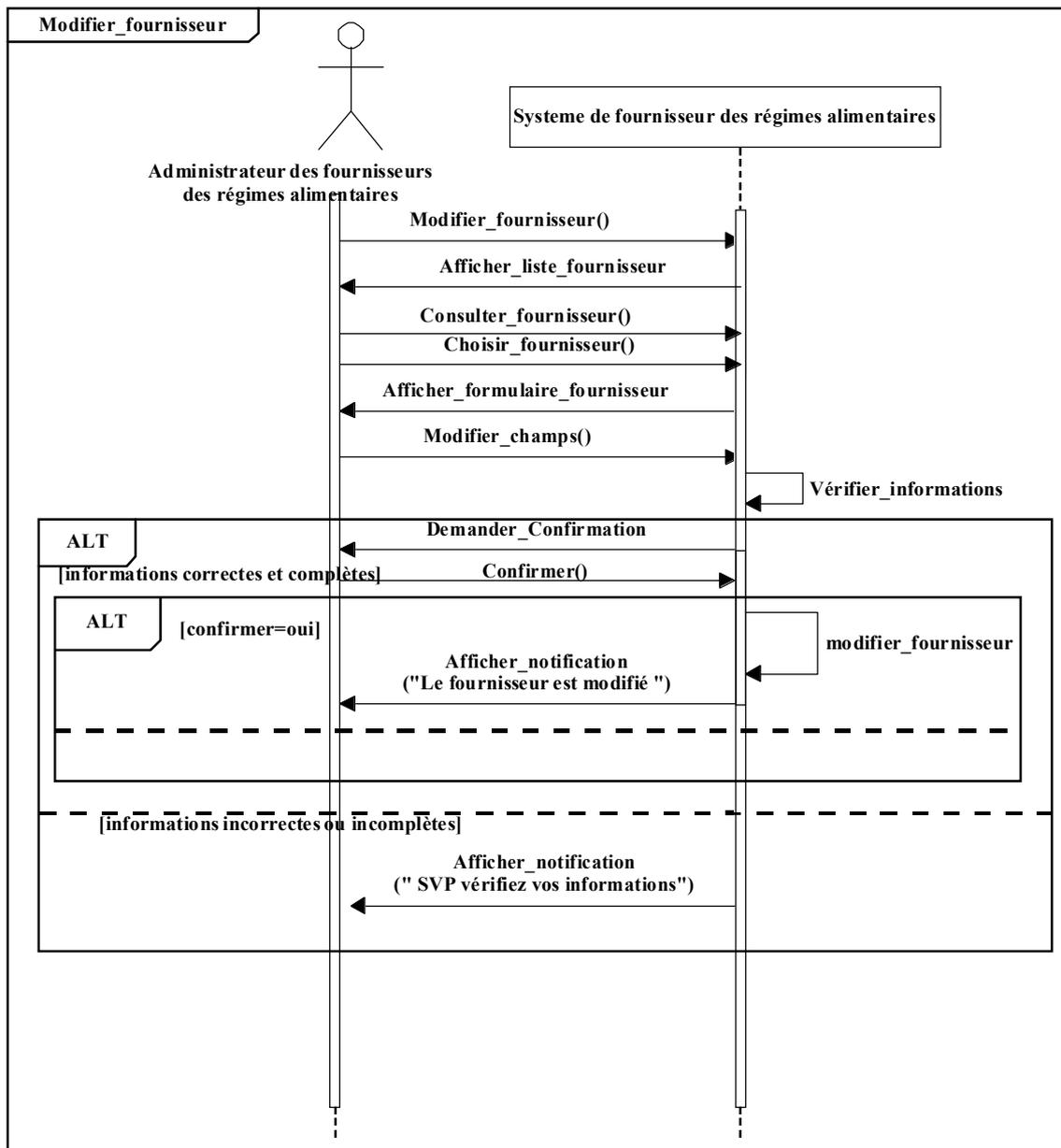


Figure 8. Diagramme de séquence système cas «Modifier fournisseur».

5.3.1.4. Cas d'utilisation « Supprimer un fournisseur des régimes alimentaires »

a. Fiche description

Sommaire d'identification
Titre : Supprimer_fournisseur. But : Supprimer un fournisseur des régimes alimentaires. Résumé : supprimer fournisseur des régimes alimentaire de la liste des usagers du système. Acteur : L'administrateur des fournisseurs des régimes alimentaires.
Des descriptions des enchainements
Pré conditions <ul style="list-style-type: none">• L'administrateur des fournisseurs des régimes alimentaires doit être authentifié.• Le fournisseur des régimes alimentaires existe déjà. Post conditions <ul style="list-style-type: none">• Un profil de fournisseur des régimes alimentaires est supprimé de la liste des usagers du système. Scénario nominal <p>Ce cas d'utilisation commence lorsque l'administrateur des fournisseurs des régimes alimentaires demande au système de fournisseur des régimes alimentaires une suppression d'un fournisseur des régimes alimentaires.</p> <ol style="list-style-type: none">1. L'administrateur des régimes alimentaires consulte la liste des fournisseurs des régimes alimentaires.2. Le système affiche la table des fournisseurs des régimes alimentaires.3. L'administrateur des fournisseurs des régimes alimentaires choisit un fournisseur des régimes alimentaires pour le supprimer.4. Le système demande une confirmation de la suppression.5. L'administrateur des fournisseurs des régimes alimentaires confirme la suppression.

6. Le système fait une mise à jour dans la base des données, puis il affiche notification «Le fournisseur est supprimé».

Scénario alternatif

- Lorsque l'administrateur des fournisseurs des régimes alimentaires ne confirme pas la suppression.
- Le système laisse la base des données avec son état primaire.

Scénario exceptionnel :

- Lorsque l'administrateur des fournisseurs des régimes alimentaires choisit l'option d'annuler la suppression.
- Le système quitte la page.

b. Diagramme d'activité

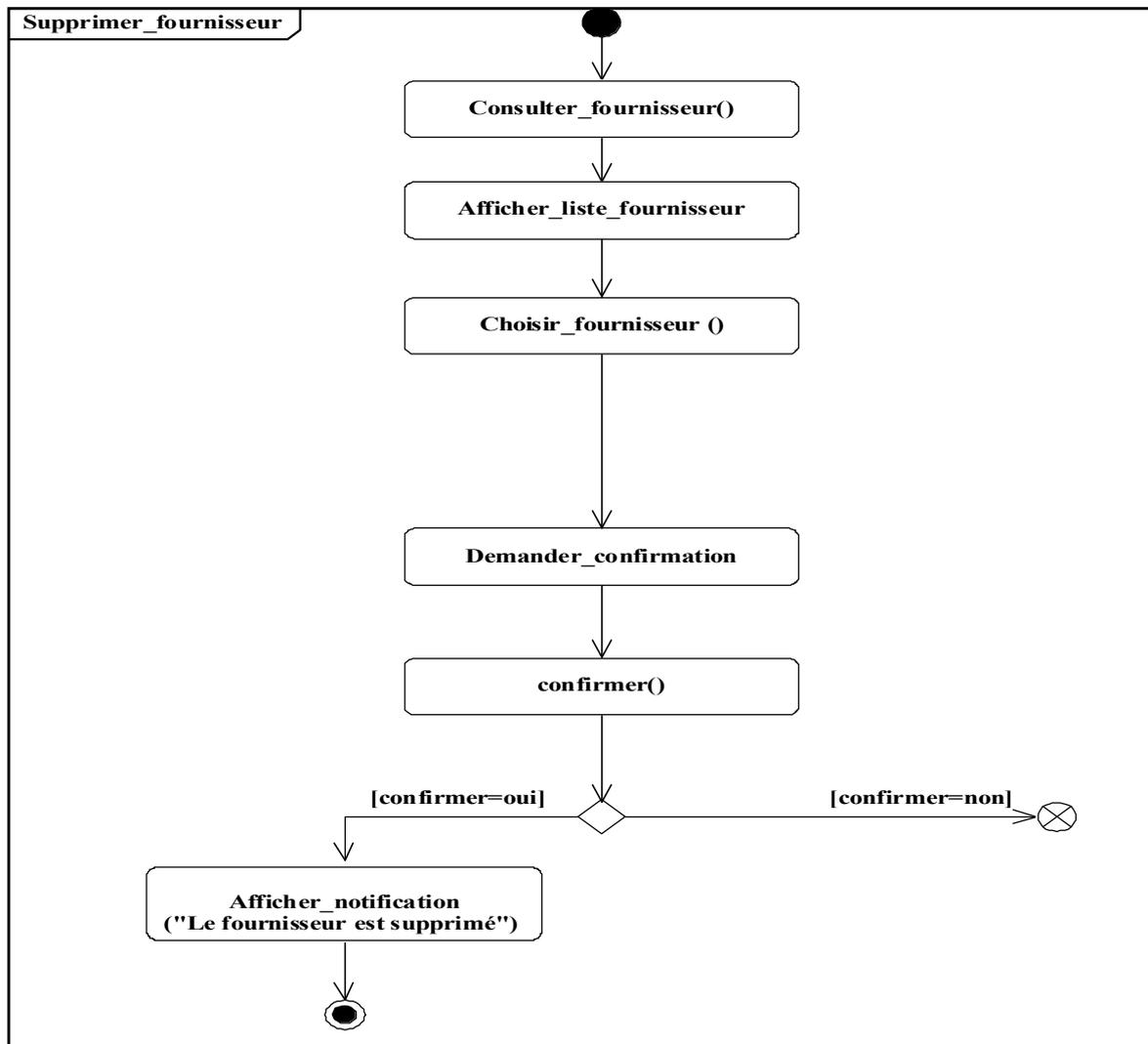


Figure 9. Diagramme d'activité cas «Supprimer fournisseur»

c. Diagramme de séquence

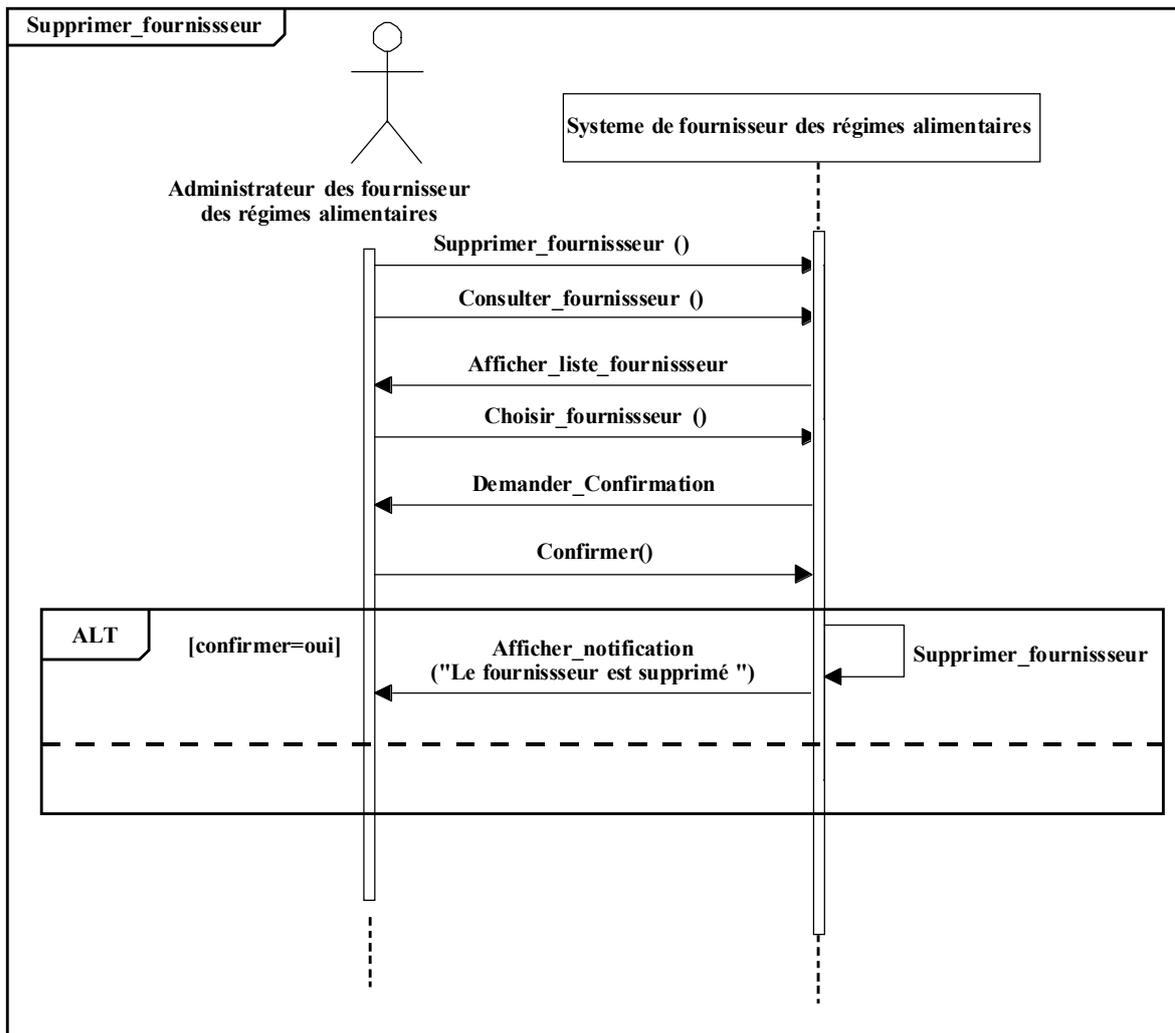


Figure 10. Diagramme de séquence système cas «Supprimer fournisseur».

5.3.1.5. Cas d'utilisation « Ajouter un régime alimentaire »

a. Fiche description

Sommaire d'identification :
Titre : Ajouter_régime. But : Ajouter un régime alimentaire. Résumé : Ajouter un régime alimentaire. Acteur : Le fournisseur des régimes alimentaires.
Description des Enchaînements :
Pré conditions : <ul style="list-style-type: none">• Le fournisseur des régimes alimentaires doit être authentifié. Post-condition : <ul style="list-style-type: none">• L'ajout est effectué. Scénario nominal : <p>Ce cas d'utilisation commence lorsque le fournisseur des régimes alimentaires demande au système de fournisseur des régimes alimentaires d'ajouter un régime.</p> <ol style="list-style-type: none">1. Le système affiche un formulaire d'ajout.2. Le fournisseur des régimes alimentaires remplit toutes les informations.3. Le système vérifie les informations saisies par le fournisseur des régimes alimentaires puis demande une confirmation.5. Si les informations complètes et correctes.6. Le fournisseur des régimes alimentaires confirme l'ajout.8. Le système emmagasine le régime et ces informations dans la base des données.9. Le système affiche une notification « Le régime est ajouté». Scénario alternatif : <ul style="list-style-type: none">➤ Lorsque le fournisseur des régimes alimentaires ne confirme pas l'ajout.

- Le système laisse la base des données avec son état primaire.
- Lorsque le fournisseur des régimes alimentaires saisie des informations incorrectes ou incomplètes.
- Le système affiche une notification «SVP vérifiez vos informations» et retourne à l'étape 1.

Scénario exceptionnel

- Lorsque le fournisseur des régimes alimentaires choisit l'option annuler.
- Le système quitte la page.

b. Diagramme d'activité

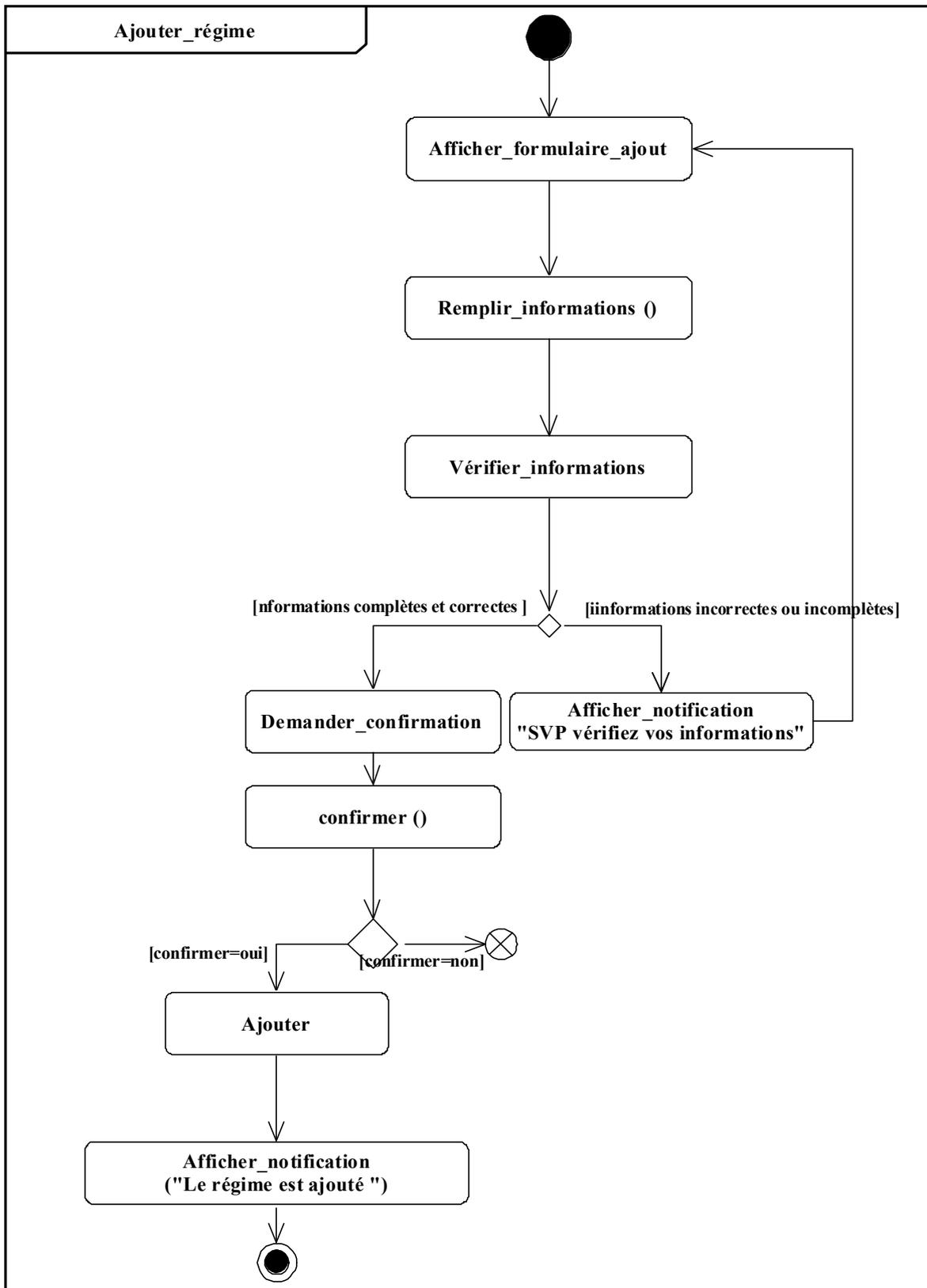


Figure 11. Diagramme d'activité cas «Ajouter régime»

c. Diagramme de séquence

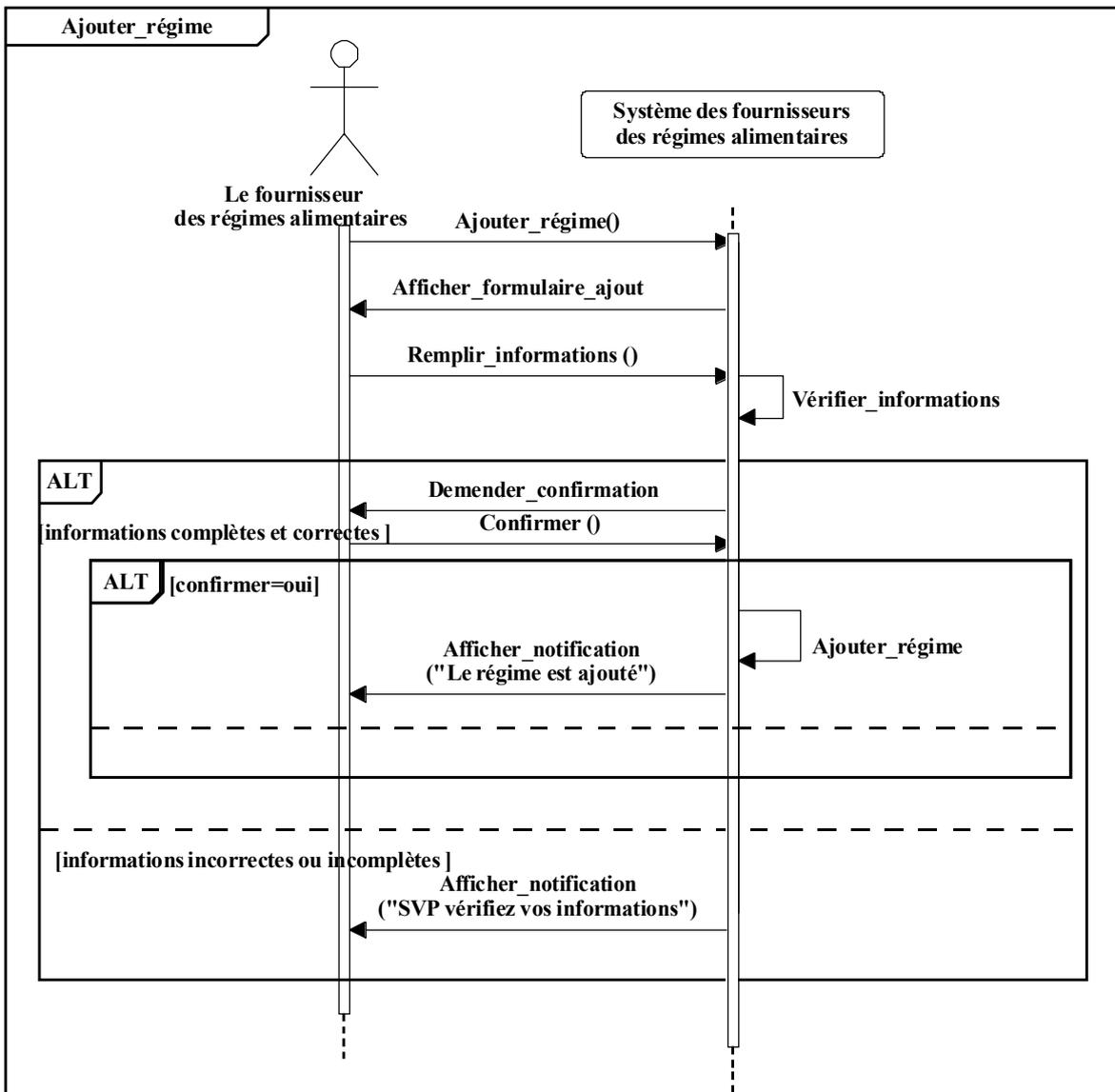


Figure 12. Diagramme de séquence système cas «Ajouter régime»

5.3.1.6. Cas d'utilisation « Consulter les régimes alimentaires »

a. Fiche description

Sommaire d'identification
Titre : Consulter_régimes. But : Afficher les régimes alimentaires existents dans la base des données. Résumé : Afficher les régimes alimentaires existents dans la base des données. Acteur : Le fournisseur des régimes alimentaires.
Description des Enchaînements
Pré condition <ul style="list-style-type: none">• Le fournisseur des régimes alimentaires doit être s'authentifier. Post condition <ul style="list-style-type: none">• La liste des régimes alimentaires est affichée. Scénario nominal <p>Ce cas d'utilisation commence lorsque le fournisseur des régimes alimentaires demande au système de fournisseur des régimes alimentaires une consultation des fournisseurs des régimes alimentaires.</p> <ul style="list-style-type: none">• Le système affiche le tableau des régimes alimentaires. Scénario alternatif <ul style="list-style-type: none">➤ Lorsque le système ne parvient pas à accéder à la base des données.➤ Le système affiche notification « Impossible d'afficher le contenu de la base ».

b. Diagramme d'activité

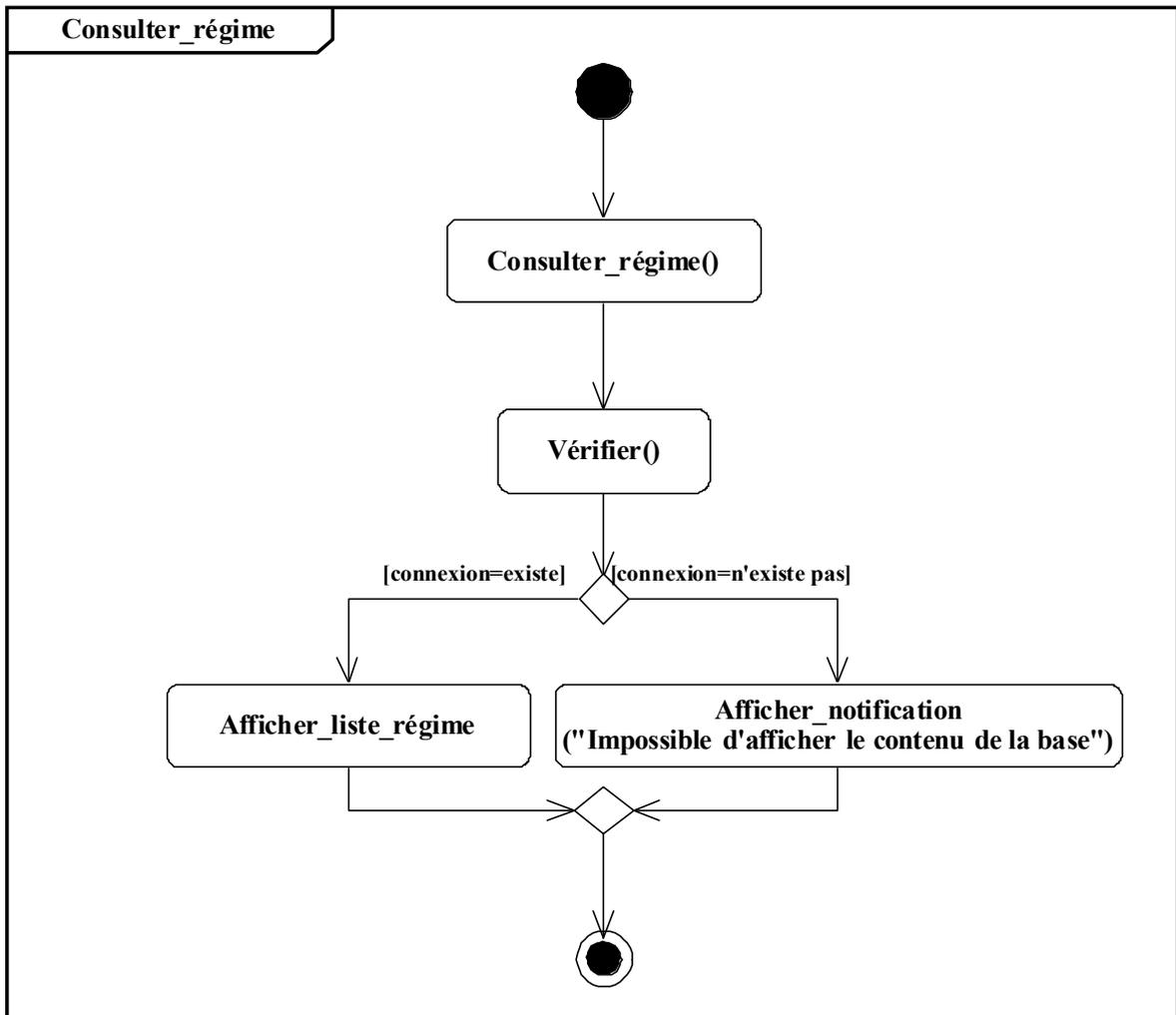


Figure 13. Diagramme d'activité cas «Consulter régime».

c. Diagramme de séquence

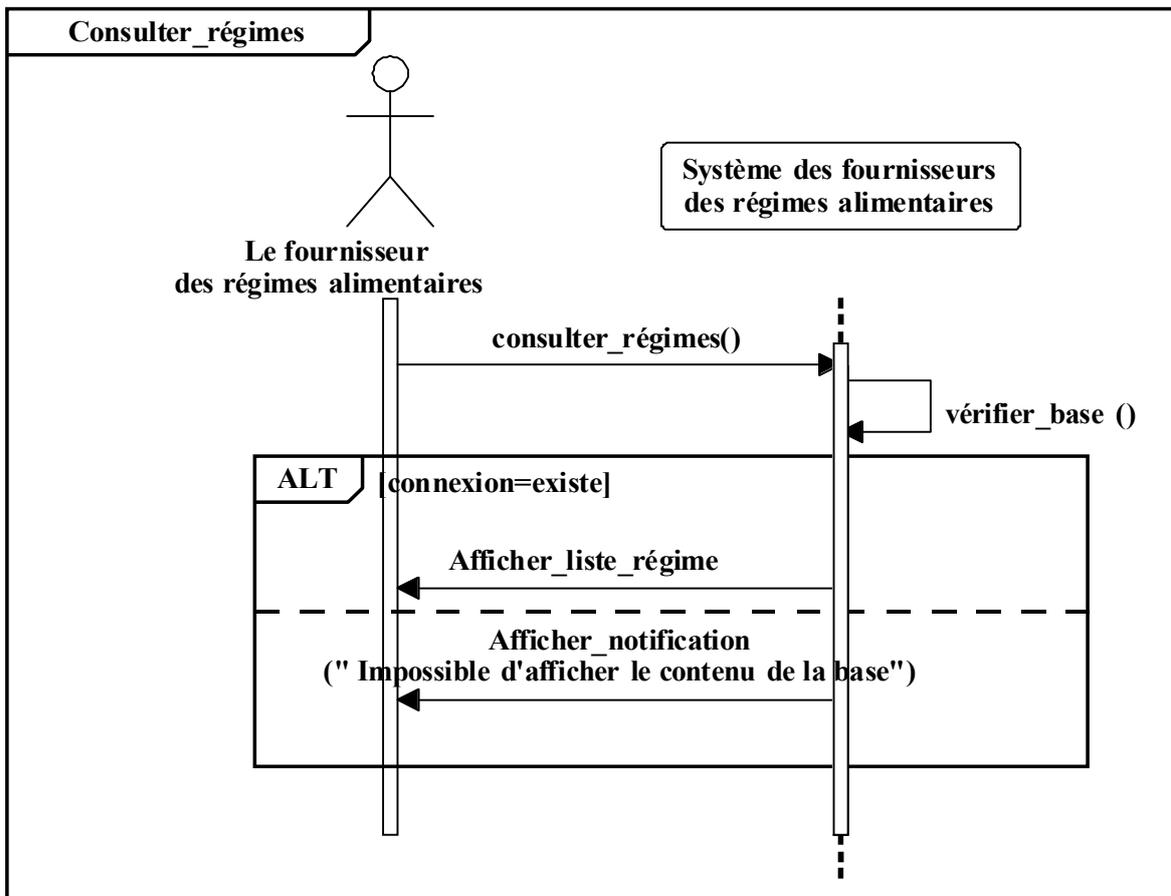


Figure 14. Diagramme de séquence système cas «Consulter régime».

5.3.1.7. Cas d'utilisation « Modifier un régime alimentaire »

a. Fiche description

Sommaire d'identification
Titre : Modifier_régime. But : Modifier les informations d'un régime alimentaire. Résumé : Changer le contenu d'un régime alimentaire. Acteur : Le fournisseur des régimes alimentaires.
Des descriptions des enchainements
Pré conditions <ul style="list-style-type: none">• Le fournisseur des régimes alimentaires doit être authentifié.• Le régime alimentaire existe déjà. Post conditions <ul style="list-style-type: none">• Le contenu d'un régime alimentaire est modifié. Scénario nominal <p>Ce cas d'utilisation commence lorsque le fournisseur des régimes alimentaires demande au système de fournisseur des régimes alimentaires une modification d'un régime alimentaire.</p> <ol style="list-style-type: none">1. Le fournisseur des régimes alimentaires consulte la liste des régimes alimentaires.2. Le système affiche la table des régimes alimentaires.3. Le fournisseur des régimes alimentaires choisit un régime alimentaire.4. Le système affiche un formulaire pour changer le contenu de ce régime.5. Le fournisseur des régimes alimentaires modifie les champs qu'il veut dans le formulaire.6. Le système vérifie les informations saisit par le fournisseur des régimes alimentaires puis demande une confirmation de la modification si les informations bien remplies.7. Le fournisseur des régimes alimentaires confirme la modification.

8. Le système fait une mise à jour concernant ce régime dans la base des données, puis il affiche notification «Le régime est modifié».

Scénario alternatif

- Lorsque le fournisseur des régimes alimentaires ne confirme pas la modification.
 - Le système laisse le régime avec son état primaire.
- Lorsque le fournisseur introduit des informations incorrectes ou incomplètes.
 - Le système affiche un message «SVP vérifiez vos informations» et retourne à l'étape 4.

Scénario exceptionnel :

- Lorsque le fournisseur des régimes alimentaires choisit l'option d'annuler la modification.
 - Le système quitte la page de la modification.

b. Diagramme d'activité

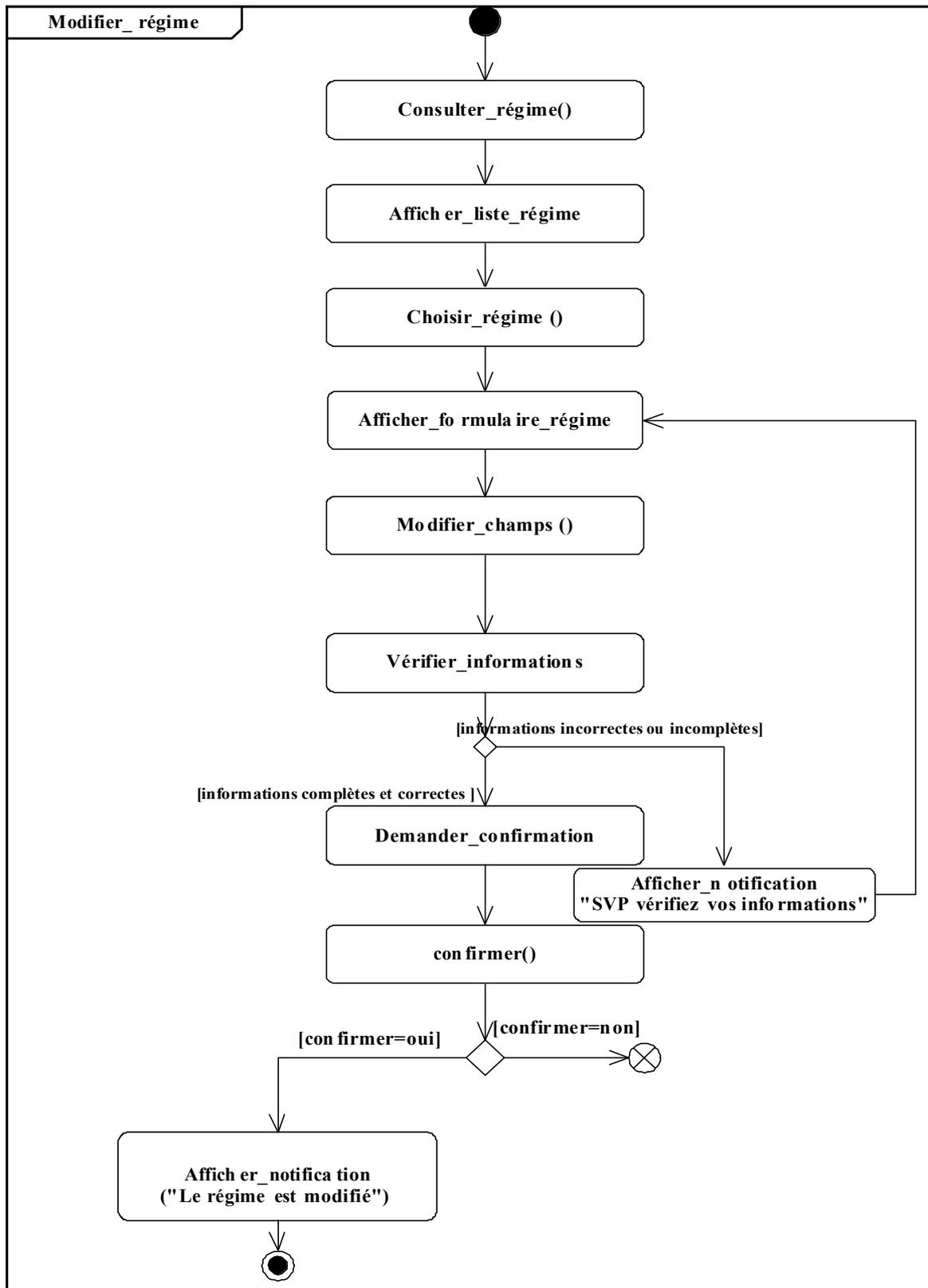


Figure 15. Diagramme d'activité cas «Modifier régime»

c. Diagramme de séquence

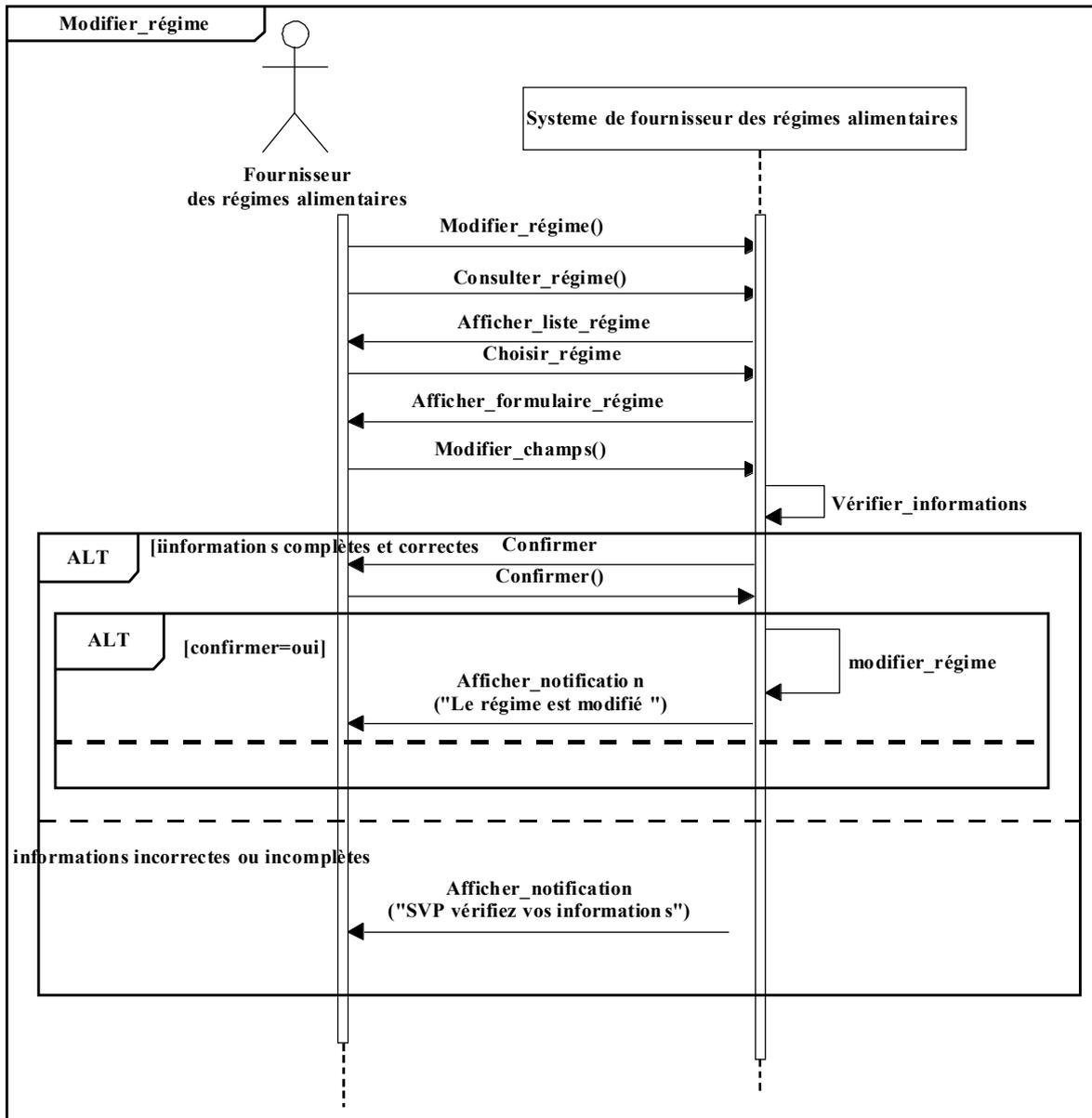


Figure 16. Diagramme de séquence système cas «Modifier régime»

5.3.1.8. Cas d'utilisation « Supprimer un régime alimentaire »

a. Fiche description

Sommaire d'identification
Titre : Supprimer_régime. But : Supprimer un régime alimentaire. Résumé : supprimer un régime alimentaire de la liste des régimes. Acteur : Le fournisseur des régimes alimentaires.
Des descriptions des enchainements
Pré conditions <ul style="list-style-type: none">• Le fournisseur des régimes alimentaires doit être authentifié.• Le régime alimentaire existe déjà. Post conditions <ul style="list-style-type: none">• Un régime alimentaire est supprimé de la liste des régimes alimentaires dans la base des données. Scénario nominal <p>Ce cas d'utilisation commence lorsque le fournisseur des régimes alimentaires demande au système de fournisseur des régimes alimentaires une suppression d'un régime alimentaire.</p> <ol style="list-style-type: none">1. Le fournisseur des régimes alimentaires consulte la table des régimes alimentaires.2. Le système affiche la table des régimes alimentaires.3. Le fournisseur des régimes alimentaires choisit un régime alimentaire pour le supprimer.4. Le système demande une confirmation de la suppression.5. Le fournisseur des régimes alimentaires confirme la suppression.6. Le système fait une mise à jour dans la base des données, puis il affiche notification «Le régime est supprimé».

Scénario alternatif

- Lorsque Le fournisseur des régimes alimentaires ne confirme pas la suppression.
 - Le système laisse la base des données avec son état primaire.

Scénario exceptionnel :

- Lorsque le fournisseur des régimes alimentaires choisit l'option annuler.
 - Le système quitte la page de la suppression.

b. Diagramme d'activité

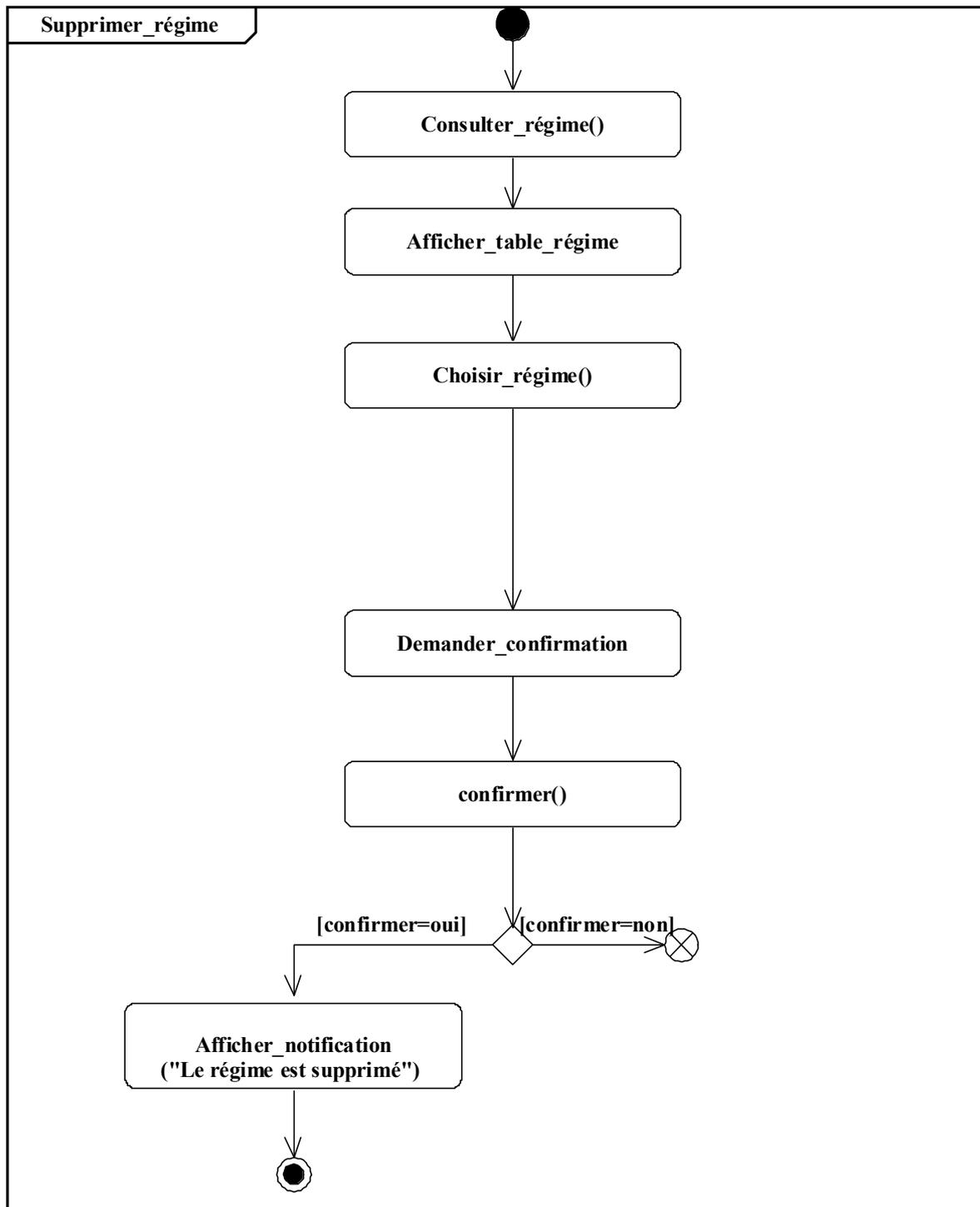


Figure 17. Diagramme d'activité cas «Supprimer régime».

c. Diagramme de séquence

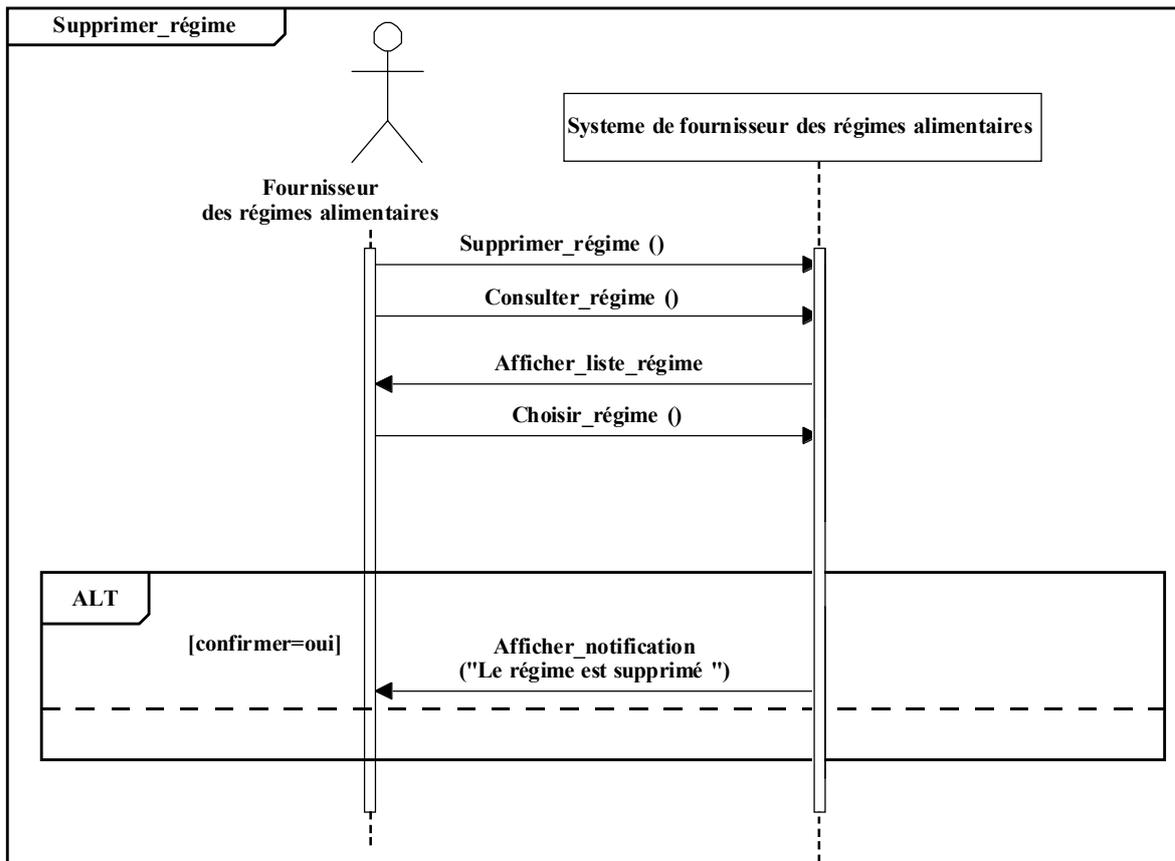


Figure 18. Diagramme de séquence système cas «Supprimer régime».

5.3.1.9. Cas d'utilisation « Ajouter un aliment/ une région/ une religion »

a. Fiche description

<p>Sommaire d'identification :</p>
<p>Titre : Ajouter_ aliment/ Ajouter_ région/ Ajouter_ religion.</p> <p>But : Ajouter un aliment/ Ajouter une région/ Ajouter une religion.</p> <p>Résumé : L'ajout d'un aliment, L'ajout d'une région ou L'ajout d'une religion.</p> <p>Acteur : Le fournisseur des régimes alimentaires.</p>
<p>Description des Enchaînements :</p>
<p>Pré conditions :</p> <ul style="list-style-type: none"> • Le fournisseur des régimes alimentaires doit être authentifié. <p>Post-condition :</p> <ul style="list-style-type: none"> • L'ajout est effectué. <p>Scénario nominal :</p> <p>Ce cas d'utilisation commence lorsque le fournisseur des régimes alimentaires demande au système de fournisseur des régimes alimentaires d'ajouter un aliment/ une région/ une religion.</p> <ol style="list-style-type: none"> 1. Le système affiche un formulaire d'ajout. 2. Le fournisseur des régimes alimentaires remplit toutes les informations. 3. Le système vérifie les informations saisies par le fournisseur des régimes alimentaires puis demande une confirmation si les informations complètes et correctes. 4. Le fournisseur des régimes alimentaires confirme l'ajout. 5. Le système emmagasine l'aliment/ la région/ la religion et ces informations dans la base des données. 6. Le système affiche une notification «l'aliment/ la région/ la religion est ajouté(e)». <p>Scénario alternatif :</p> <ul style="list-style-type: none"> ➤ Lorsque le fournisseur des régimes alimentaires ne confirme pas l'ajout :

- Le système laisse la base des données avec son état primaire.
- Lorsque les informations introduites sont incorrectes ou incomplètes.
- Le système affiche une notification «SVP vérifiez vos informations» et retourne à l'étape 1.

Scénario exceptionnel

- Lorsque le fournisseur des régimes alimentaires choisit l'option annuler.
- Le système quitte la page d'ajout.

b. Diagramme d'activité

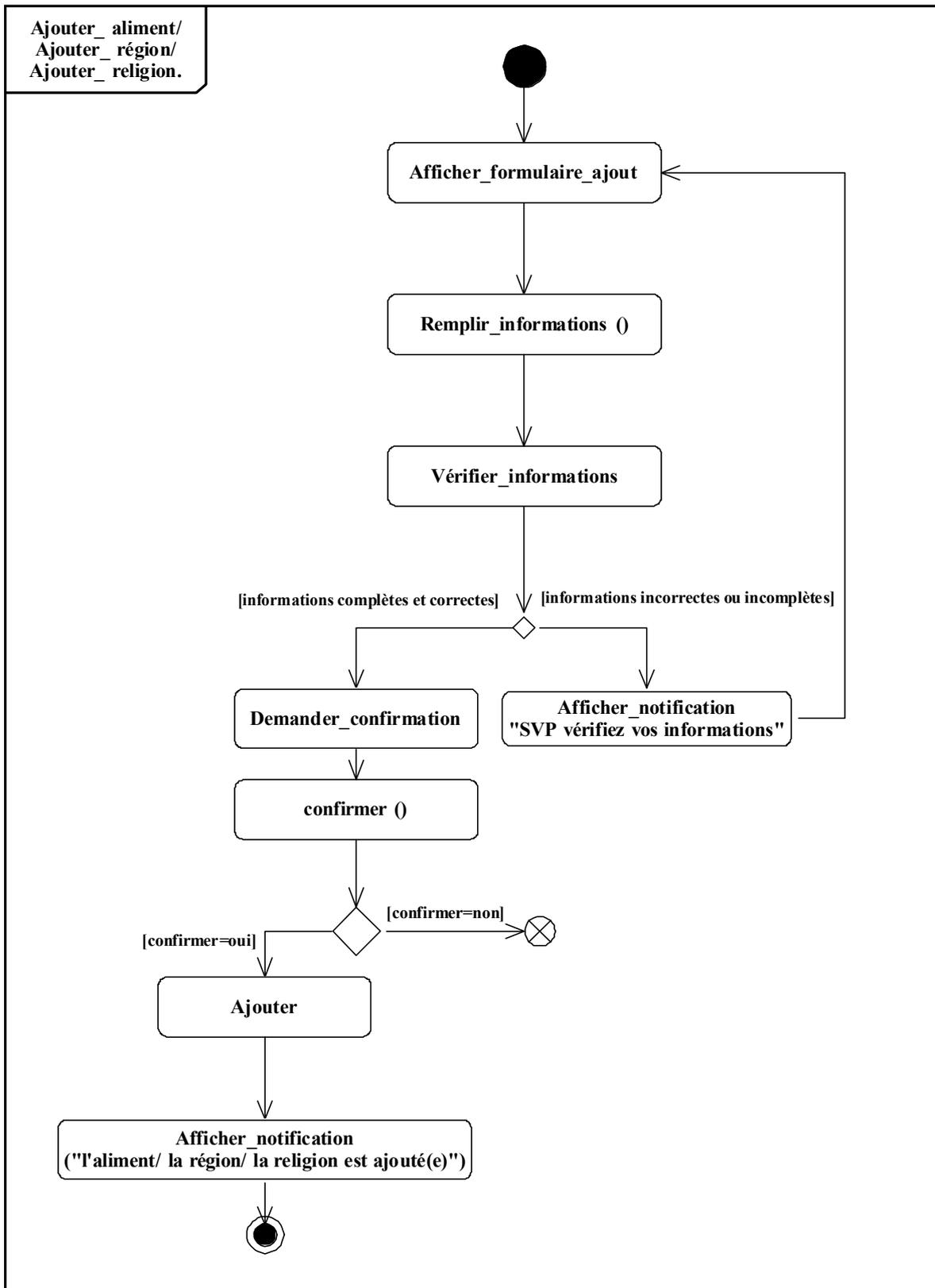


Figure 19. Diagramme d'activité cas «Ajouter aliment/ région/ religion»

c. Diagramme de séquence

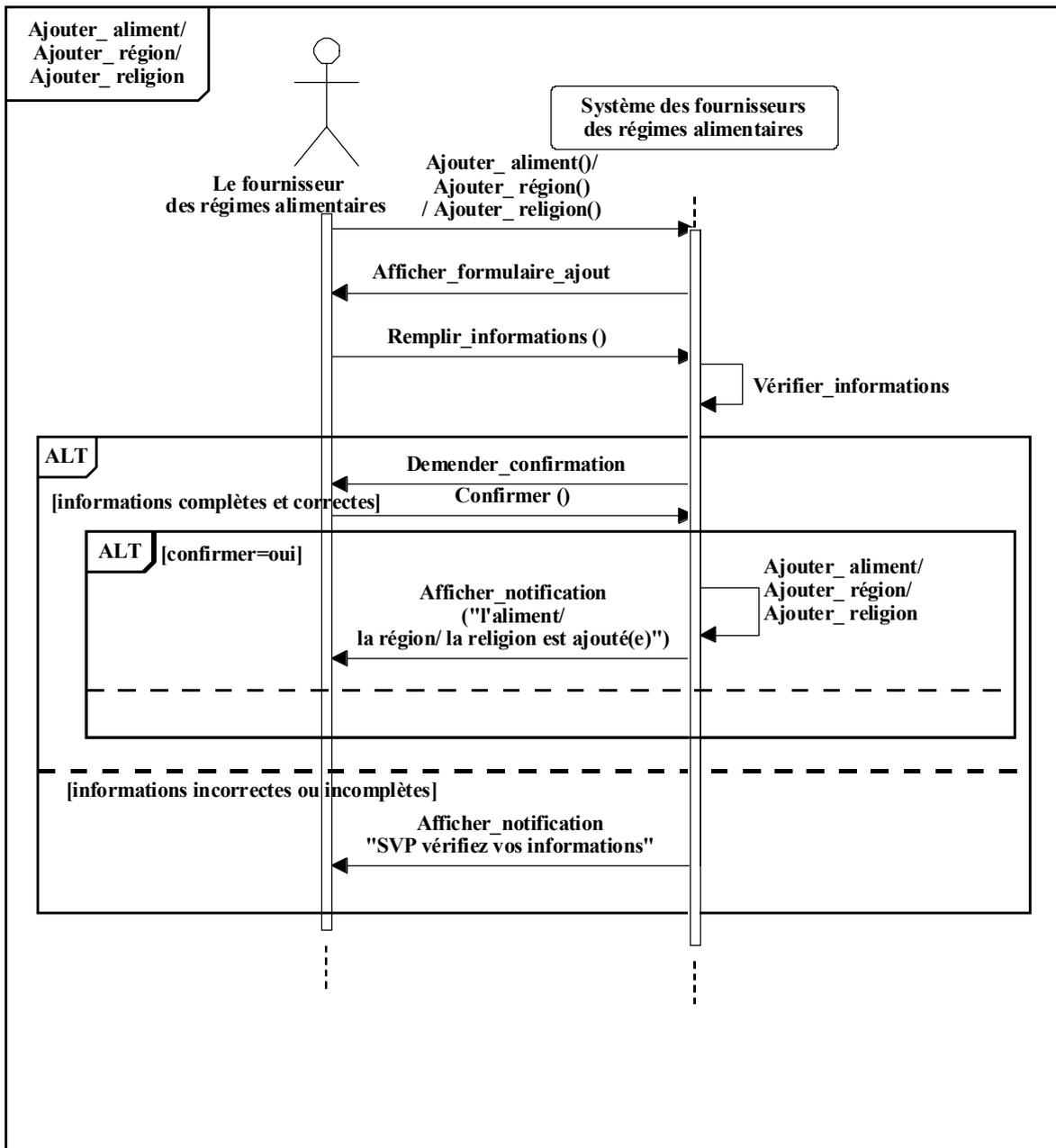


Figure 20. Diagramme de séquence système cas «Ajouter aliment/ religion/ région».

5.3.1.10. Cas d'utilisation « Consulter les aliments/ les régions/ les religions »

a. Fiche description

Sommaire d'identification
Titre : Consulter_ aliments/ Consulter_ régions / Consulter_ religions. But : Afficher aliments/ régions/ religions existents dans la base des données. Résumé : Afficher les aliments/ régions/ religions existents dans la base des données. Acteur : Le fournisseur des régimes alimentaires.
Description des Enchaînements
Pré condition <ul style="list-style-type: none">• Le fournisseur des régimes alimentaires doit être s'authentifier. Post condition <ul style="list-style-type: none">• La liste des aliments/ régions/ religions est affichée. Scénario nominal <p>Ce cas d'utilisation commence lorsque l'administrateur des fournisseurs des régimes alimentaires demande au système de fournisseur des régimes alimentaires une consultation des aliments/ régions/ religions.</p> <ul style="list-style-type: none">• Le système affiche la liste des aliments/ régions/ religions. Scénario alternatif <ul style="list-style-type: none">➤ Lorsque le système ne parvient pas à accéder à la base des données.• Le système affiche notification « Impossible d'afficher le contenu de la base».

b. Diagramme d'activité

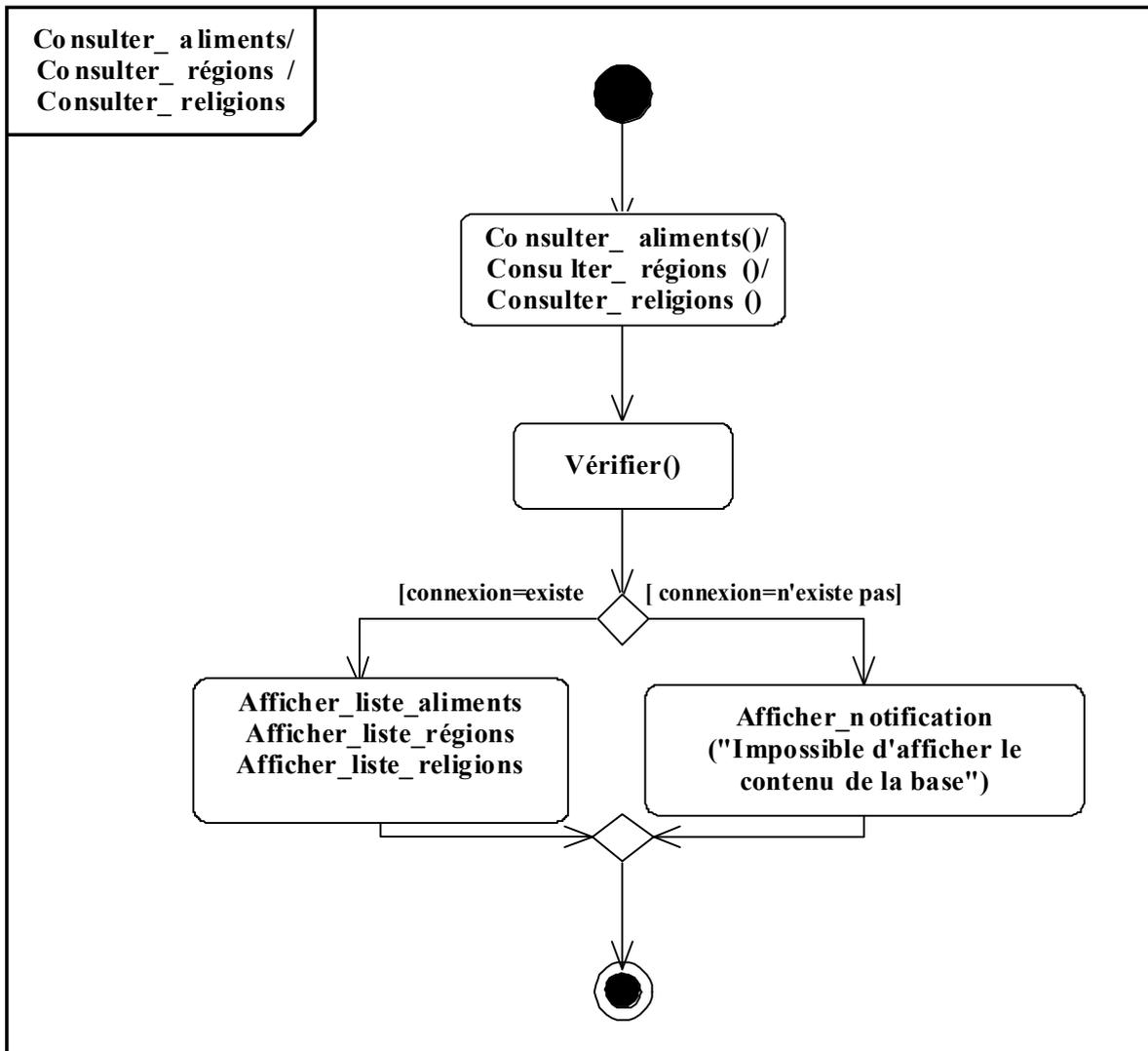


Figure 21. Diagramme d'activité cas «Consulter aliments/ régions/ religions».

c. Diagramme de séquence

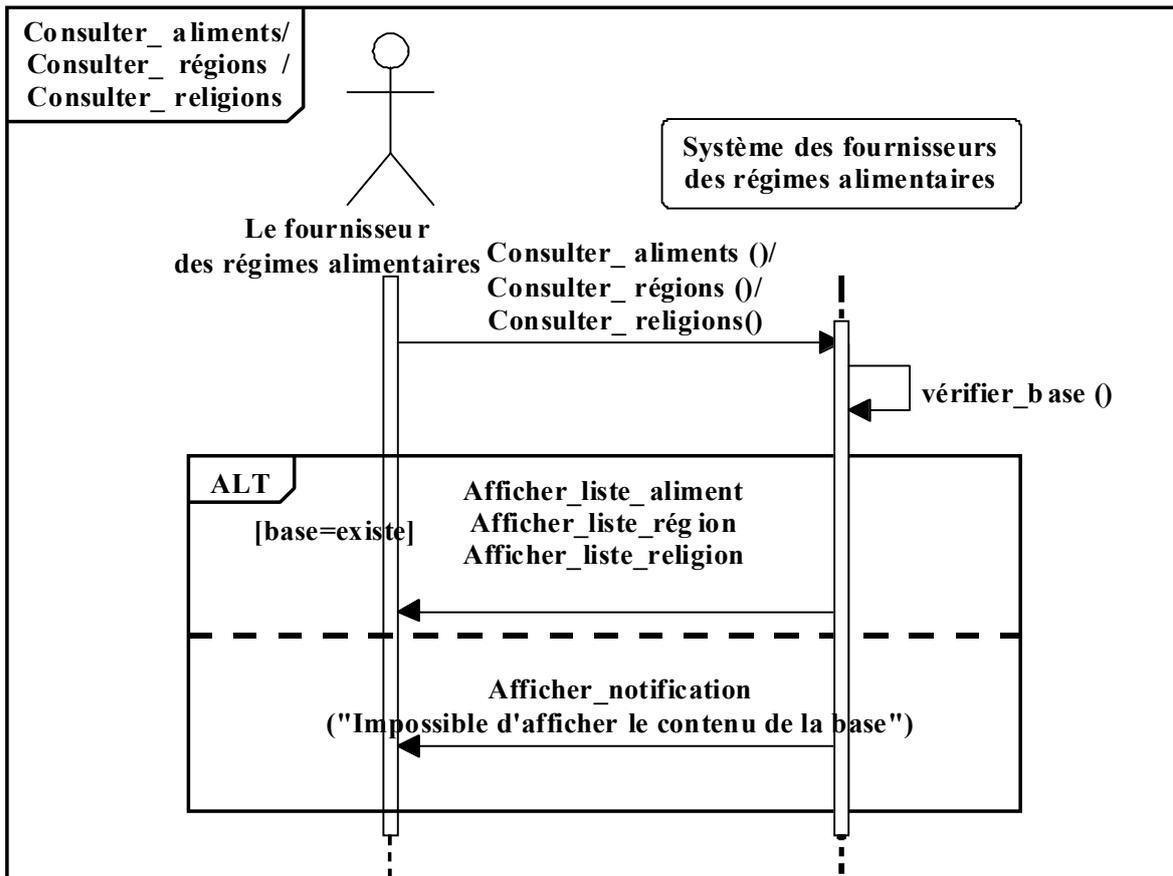


Figure 22. Diagramme de séquence système cas «Consulter aliments/régions/religions»

5.3.1.11. Cas d'utilisation « Modifier un aliment/ une région/ une religion»

a. Fiche description

Sommaire d'identification
<p>Titre : Modifier_aliment/ Modifier_région/ Modifier_religion.</p> <p>But : Modifier les informations d'un aliment/ une région/ une religion.</p> <p>Résumé : Changer le contenu d'un aliment/ une région/ une religion.</p> <p>Acteur : Le fournisseur des régimes alimentaires.</p>
Des descriptions des enchainements
<p>Pré conditions</p> <ul style="list-style-type: none"> • Le fournisseur des régimes alimentaires doit être authentifié. • L'aliment/ la région/ la religion existe déjà. <p>Post conditions</p> <ul style="list-style-type: none"> • Le contenu un aliment/ une région/ une religion. <p>Scénario nominal</p> <p>Ce cas d'utilisation commence lorsque le fournisseur des régimes alimentaires demande au système de fournisseur des régimes alimentaires une modification d'un aliment/ une région/ une religion</p> <ol style="list-style-type: none"> 1. Le fournisseur des régimes alimentaires consulte la liste des aliments/ des régions/ des religions. 2. Le système affiche la table des aliments/ des régions/ des religions. 3. Le fournisseur des régimes alimentaires choisit un aliment/ une région/ une religion. 4. Le système affiche un formulaire pour changer le contenu de ce (cette) un aliment/ une région/ une religion. 5. Le fournisseur des régimes alimentaires modifie les champs qu'il veut dans le formulaire. 6. Le système vérifie les informations saisis par le fournisseur des régimes alimentaires

puis demande une confirmation de la modification si les informations sont complètes et correctes.

7. Le fournisseur des régimes alimentaires confirme la modification.

8. Le système fait une mise à jour concernant ce régime dans la base des données, puis il affiche notification «L'aliment/ La région/ La religion est modifié (e)».

Scénario alternatif

- Lorsque le fournisseur des régimes alimentaires ne confirme pas la modification.
 - Le système laisse l'aliment/ la région/ la religion avec son état primaire.
- Lorsque le fournisseur des régimes alimentaires introduit des informations incorrectes ou incomplètes
 - Le système affiche une notification «SVP vérifiez vos informations» et retourne à l'étape 4.

Scénario exceptionnel :

- Lorsque le fournisseur des régimes alimentaires choisit l'option annuler.
 - Le système quitte la page.

b. Diagramme d'activité

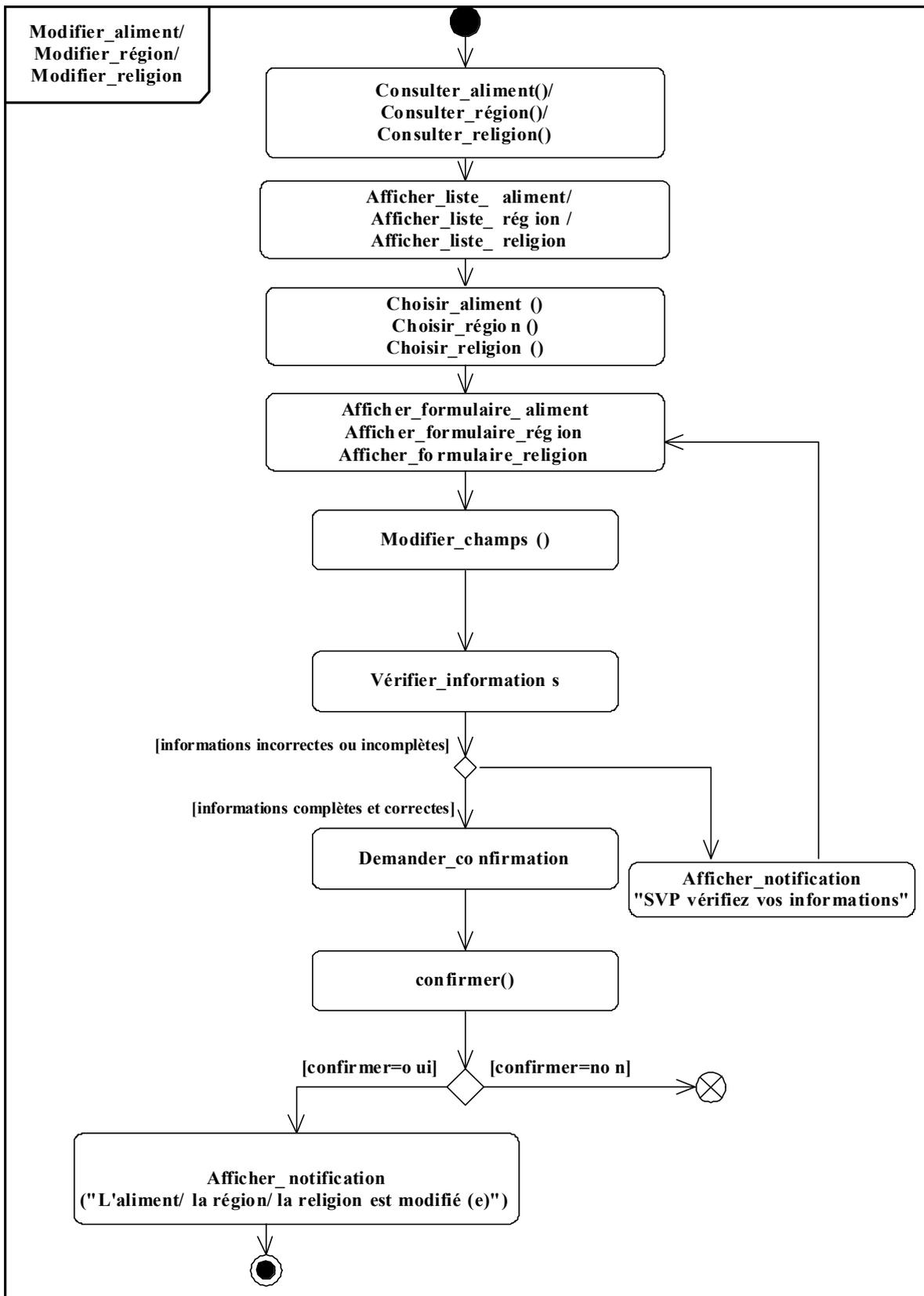


Figure 23. Diagramme d'activité cas «Modifier région/ religion/ aliment»

c. Diagramme de séquence

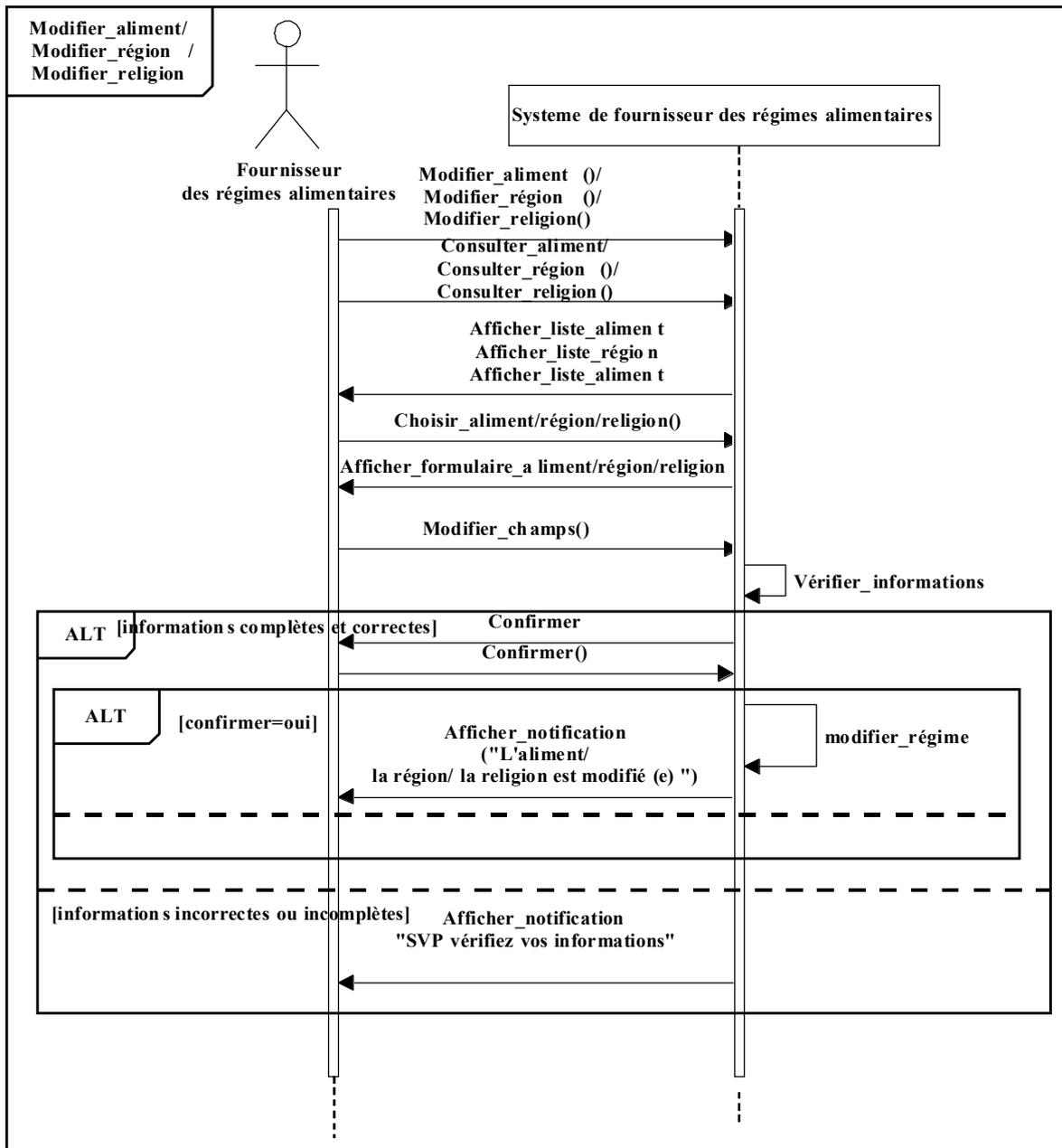


Figure 24. Diagramme de séquence système cas «Modifier aliment/ région/ religion»

5.3.1.12. Cas d'utilisation « Supprimer_aliment/ Supprimer_région/ Supprimer _religion»

a. Fiche description

Sommaire d'identification
<p>Titre : Supprimer_aliment/ Supprimer_région/ Supprimer _religion.</p> <p>But : Supprimer un aliment/ une région/ une religion.</p> <p>Résumé : Supprimer un aliment/ une région/ une religion.</p> <p>Acteur : Le fournisseur des régimes alimentaires.</p>
Des descriptions des enchainements
<p>Pré conditions</p> <ul style="list-style-type: none"> • Le fournisseur des régimes alimentaires doit être authentifié. • L'aliment/la région/ la religion existe déjà. <p>Post conditions</p> <ul style="list-style-type: none"> • Un aliment/ une région/ une religion est supprimé(e) de la liste des aliments/des régions/ la religion dans la base des données. <p>Scénario nominal</p> <p>Ce cas d'utilisation commence lorsque le fournisseur des régimes alimentaires demande au système de fournisseur des régimes alimentaires une suppression d'un aliment/ une région/ une religion.</p> <ol style="list-style-type: none"> 1. Le fournisseur des régimes alimentaires consulte la table des aliments/ des régions/ des religions. 2. Le système affiche la table des aliments/ des régions/ des religions. 3. Le fournisseur des régimes alimentaires choisit un aliment/ une région/ une religion pour le/la supprimer. 4. Le système demande une confirmation de la suppression. 5. Le fournisseur des régimes alimentaires confirme la suppression.

6. Le système fait une mise à jour dans la base des données, puis il affiche notification «L'aliment/ la région/ la religion est supprimé(e)».

Scénario alternatif

- Lorsque Le fournisseur des régimes alimentaires ne confirme pas la suppression.
 - Le système laisse la base des données avec son état primaire.

Scénario exceptionnel :

- Lorsque le fournisseur des régimes alimentaires choisit l'option d'annuler la suppression.
 - Le système quitte la page de la suppression.

b. Diagramme d'activité

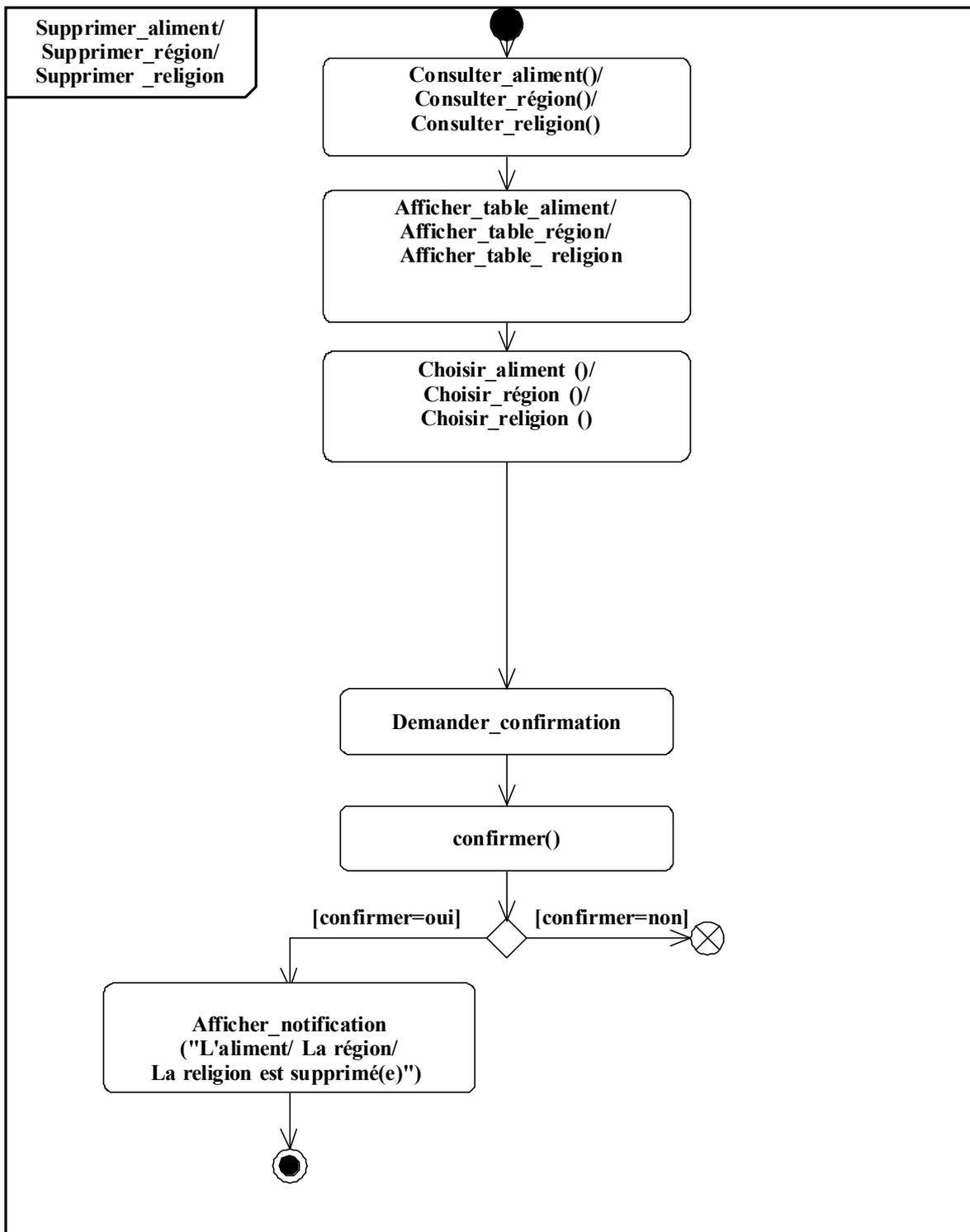


Figure 25. Diagramme d'activité cas «Supprimer région/ religion/ aliment»

c. Diagramme de séquence

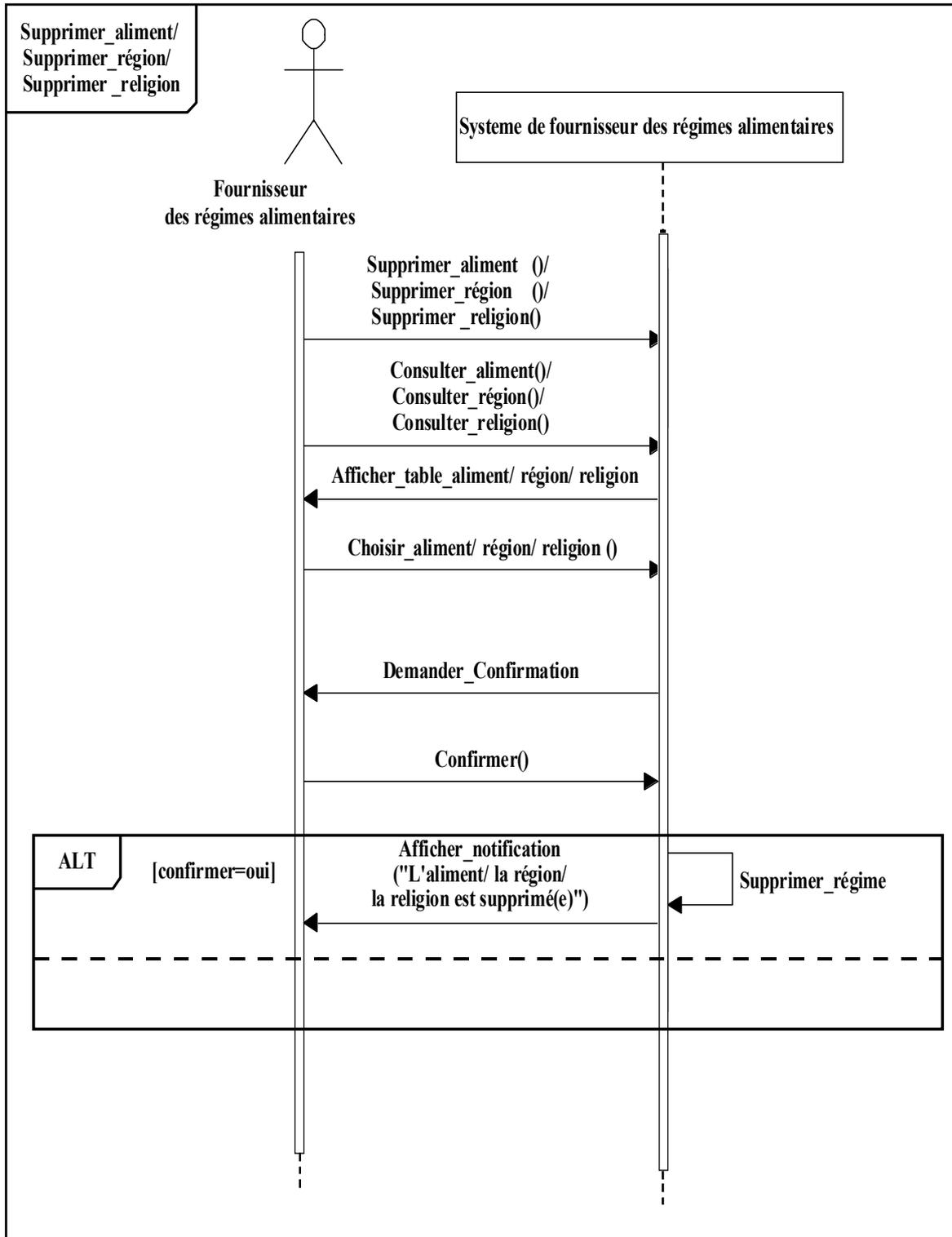


Figure 26. Diagramme de séquence système cas «Supprimer aliment/ région/ religion»

5.3.2. Au niveau de l'utilisateur des régimes alimentaires

5.3.2.1. Cas d'utilisation « Ajouter annonce »

Sommaire d'identification
Titre : Ajouter annonce. But : Ajouter annonce. Résumé : Ajouter annonce dans le site web. Acteur : L'administrateur de site web des régimes alimentaires.
Description des enchainements
Pré conditions <ul style="list-style-type: none">• L'administrateur de site web des régimes alimentaires doit être s'authentifier. Post condition <ul style="list-style-type: none">• Le system affiche un message d'enregistrement. Scénario nominal <p>Ce cas d'utilisation commence lorsque l'administrateur de site web des régimes alimentaires demande l'ajout d'une annonce.</p> <ol style="list-style-type: none">1. Le système affiche le formulaire.2. L'administrateur de site web saisie les informations de l'annonce.3. Le système vérifie les informations4. Si les informations sont correctes système affiche une notification : «annonce est ajoutée avec succès ». Scénario alternatif <ul style="list-style-type: none">➤ Si il y'a une information saisie incorrecte.<ul style="list-style-type: none">• Le système retourne à l'étape 1. Scénario exceptionnel <ul style="list-style-type: none">➤ Lorsque l'administrateur choisit l'option d'annuler l'ajout.

- Le système quitte la page d'ajout.

c. Diagramme de séquence

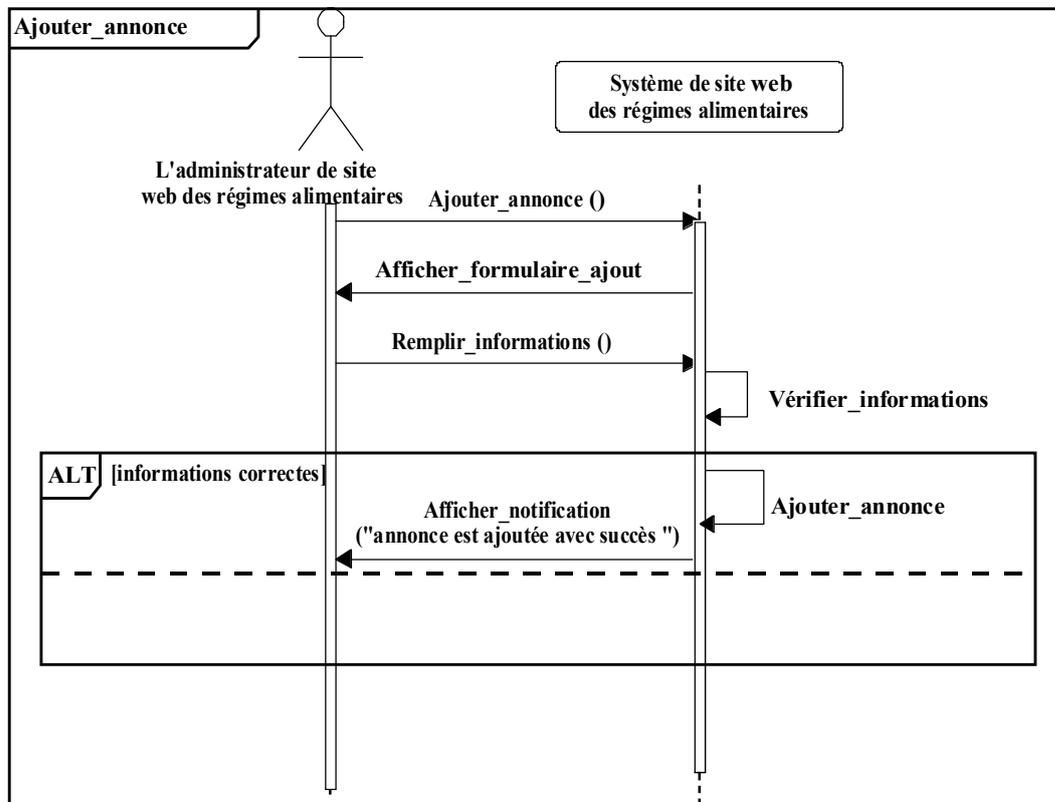


Figure 27. Diagramme de séquence système cas «Ajouter annonce».

5.3.2.2. Cas d'utilisation « Inscire»

a. Fiche description

Sommaire d'identification
Titre : Inscire. But : Devenir un client. Résumé : Demander l'inscription dans le site web pour obtenir un compte client. Acteur : Le visiteur de site web
Description des enchainements
Pré conditions <ul style="list-style-type: none">• connexion existe. Post condition <ul style="list-style-type: none">• Le system affiche un message de confirmation. Scénario nominal <p>Ce cas d'utilisation commence lorsque le visiteur de site web lance l'option d'inscrire.</p> <ol style="list-style-type: none">1. Le système affiche un formulaire d'inscription.2. Le visiteur de site web remplit les champs de formulaire et valide les informations saisies.3. Le système vérifie les informations saisies.4. Si toutes les informations sont correctes le système affiche une notification «l'inscription a réussi». Scénario alternatif <ul style="list-style-type: none">➤ Si il y'a une information saisie incorrecte.<ul style="list-style-type: none">• Le système retourne à l'étape 1. Scénario exceptionnel

- Lorsque le visiteur choisit l'option annuler.
- Le système quitte la page de l'inscription.

b. Diagramme d'activité

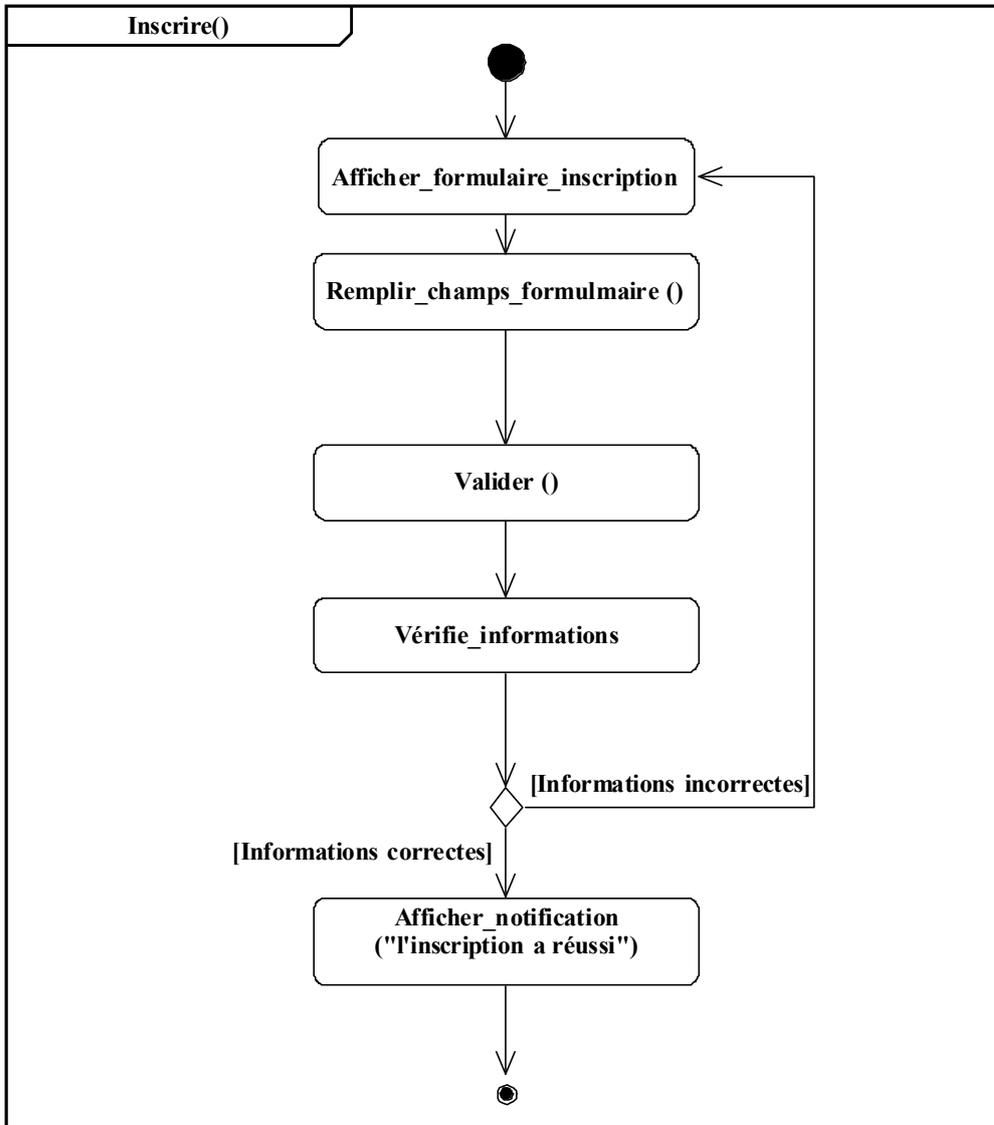


Figure 28. Diagramme d'activité cas «Inscrire».

5.3.2.3. Cas d'utilisation « Ajouter fichier»

a. Fiche description

Sommaire d'identification
Titre : Ajouter_fichier. But : créer un nouvel fichier. Résumé : Ajouter un nouvel fichier dans la liste des fichiers dans la base des données. Acteur : Administrateur de site web des régimes alimentaires.
Des descriptions des enchainements
Pré conditions <ul style="list-style-type: none">• L'administrateur de site web des régimes alimentaires doit être authentifié. Post condition <ul style="list-style-type: none">• Le fichier est ajouté à la base des données. Scénario nominal <p>Ce cas d'utilisation commence lorsque l'administrateur de site web veut faire un ajout d'un fichier.</p> <ol style="list-style-type: none">1. Le système affiche un formulaire.2. L'administrateur remplit le formulaire et valide les informations saisies.3. Le système vérifie les informations saisies.4. Si les informations saisies incorrectes le système affiche une notification « fichier est ajouté ». Scénario alternatif <ul style="list-style-type: none">➤ Si l'administrateur saisit des informations incorrectes<ul style="list-style-type: none">• Le système retourne à l'étape 1. Scénario exceptionnel

- Lorsque l'administrateur choisit l'option annuler l'ajout.
- Le système quitte la page de l'ajout.

b. Diagramme de séquence

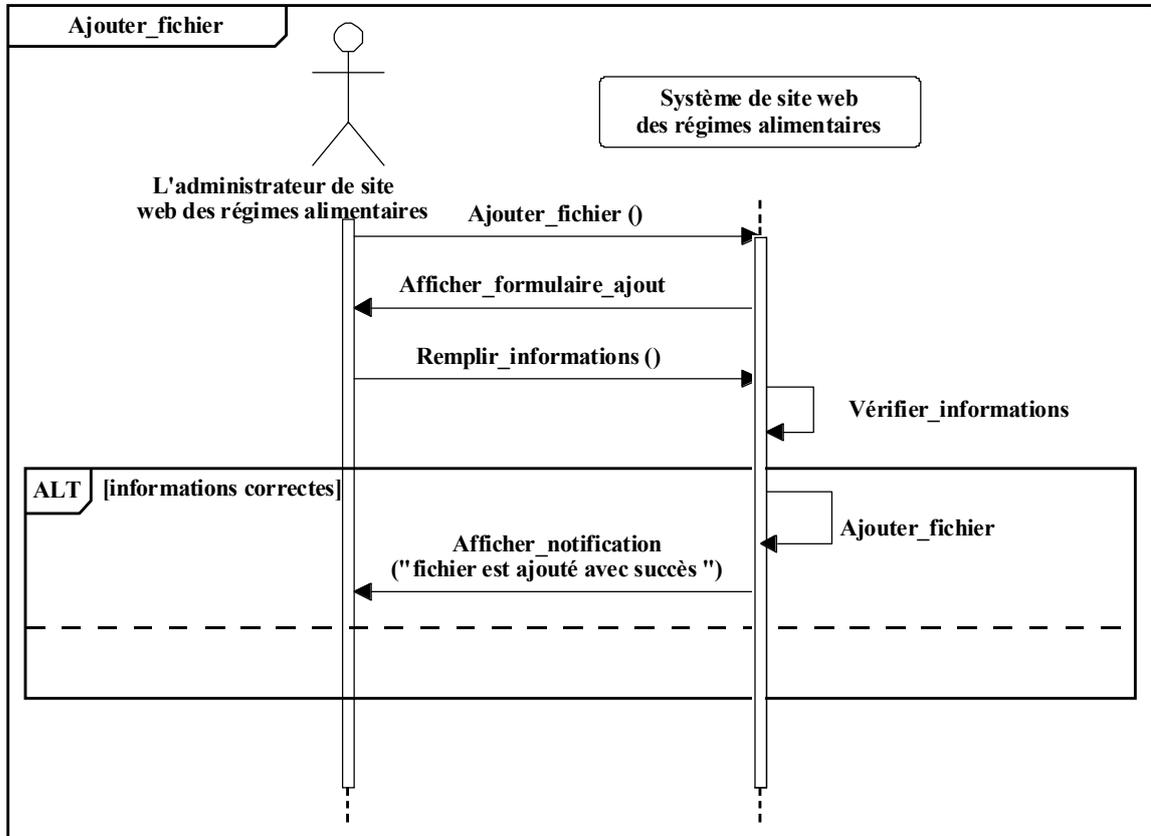


Figure 29. Diagramme de séquence système cas «Ajouter fichier»

5.3.2.4. Cas d'utilisation « Modifier fichier »

a. Fiche description

Sommaire d'identification
Titre : Modifier_fichier. But : Modifier les informations d'un fichier. Résumé : changer les informations d'un fichier. Acteur : L'administrateur de site web de régimes alimentaires.
Des descriptions des enchainements
Pré conditions <ul style="list-style-type: none">• L'administrateur doit être authentifié.• Le fichier existe déjà. Post conditions <ul style="list-style-type: none">• Les informations de fichier sont modifiées. Scénario nominal <p>Ce cas d'utilisation commence lorsque l'administrateur demande au système de site web des régimes alimentaires une modification d'un fournisseur des régimes alimentaires.</p> <ol style="list-style-type: none">1. l'administrateur consulte la table des fichiers.2. Le système affiche la table des fichiers.3. L'administrateur choisit un fichier.4. Le système affiche un formulaire pour changer les informations de ce fichier.5. L'administrateur modifie les champs qu'il veut dans le formulaire.6. Le système vérifie les informations saisis par l'administrateur.7. Si les informations saisies sont correctes le système affiche une notification « Le fichier est modifié ». Scénario alternatif

- Lorsque L'administrateur fait une erreur quand il remplit les champs.
- Le système retourne à l'étape 4.

Scénario exceptionnel :

- Lorsque l'administrateur choisit l'option annuler.
- Le système quitte la page.

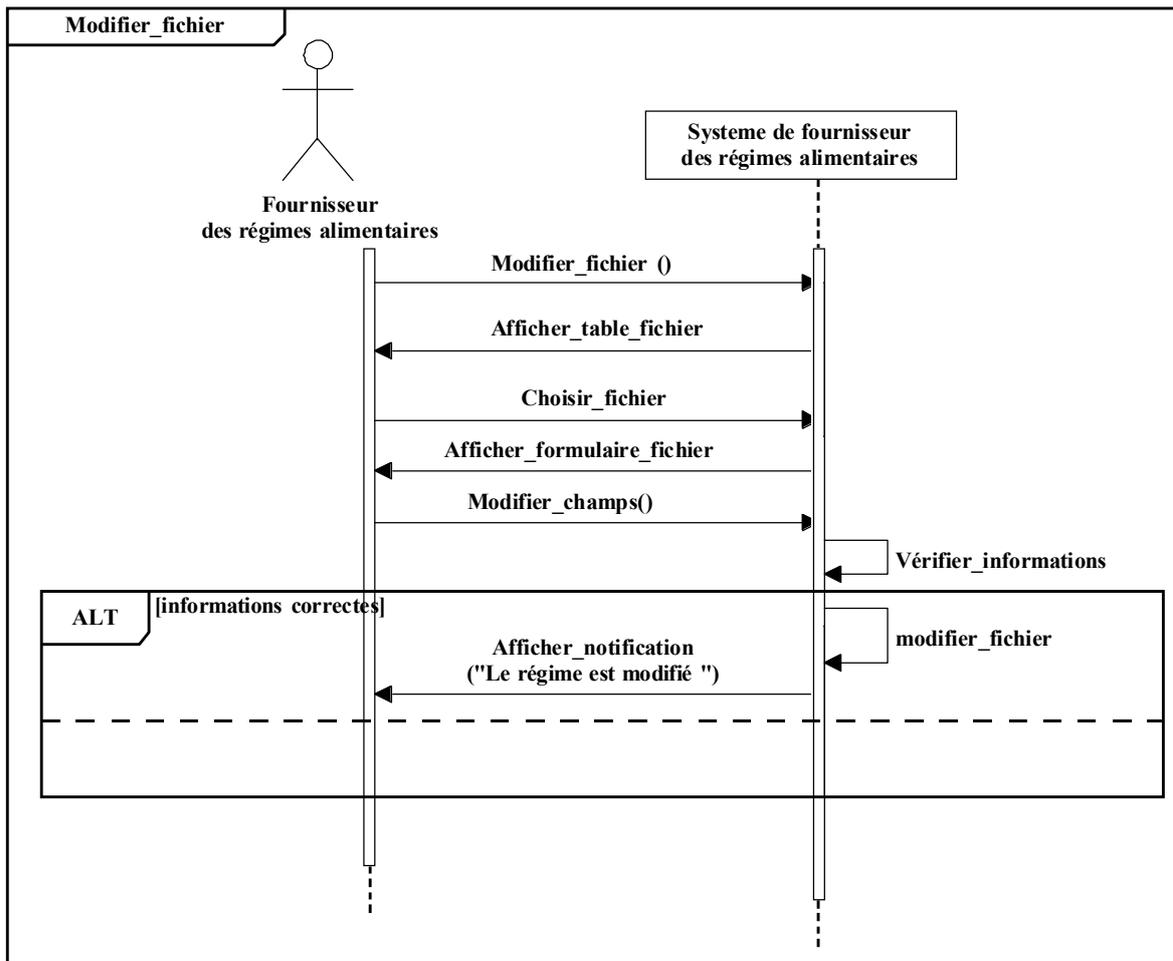
b. Diagramme de séquence

Figure 30. Diagramme de séquence système cas «Modifier fichier»

5.3.2.5. Cas d'utilisation « Supprimer un fichier »

a. Fiche description

Sommaire d'identification
Titre : Supprimer_fichier. But : Supprimer un fichier. Résumé : Supprimer un fichier de la liste des fichiers dans la base des données. Acteur : L'administrateur de site web des régimes alimentaires.
Des descriptions des Enchainements
Pré conditions <ul style="list-style-type: none">• L'administrateur doit être authentifié.• Le fichier existe déjà. Post conditions <ul style="list-style-type: none">• Un fichier est supprimé. Scénario nominal <p>Ce cas d'utilisation commence lorsque l'administrateur demande au système de site web des régimes alimentaires une suppression d'un fichier.</p> <ol style="list-style-type: none">1. L'administrateur consulte la table des fichiers.2. Le système affiche la table des fichiers.3. L'administrateur choisit un fichier pour le supprimer.4. Le système fait une mise à jour dans la base des données, puis il affiche notification «Le fichier est supprimé».

b. Diagramme de séquence

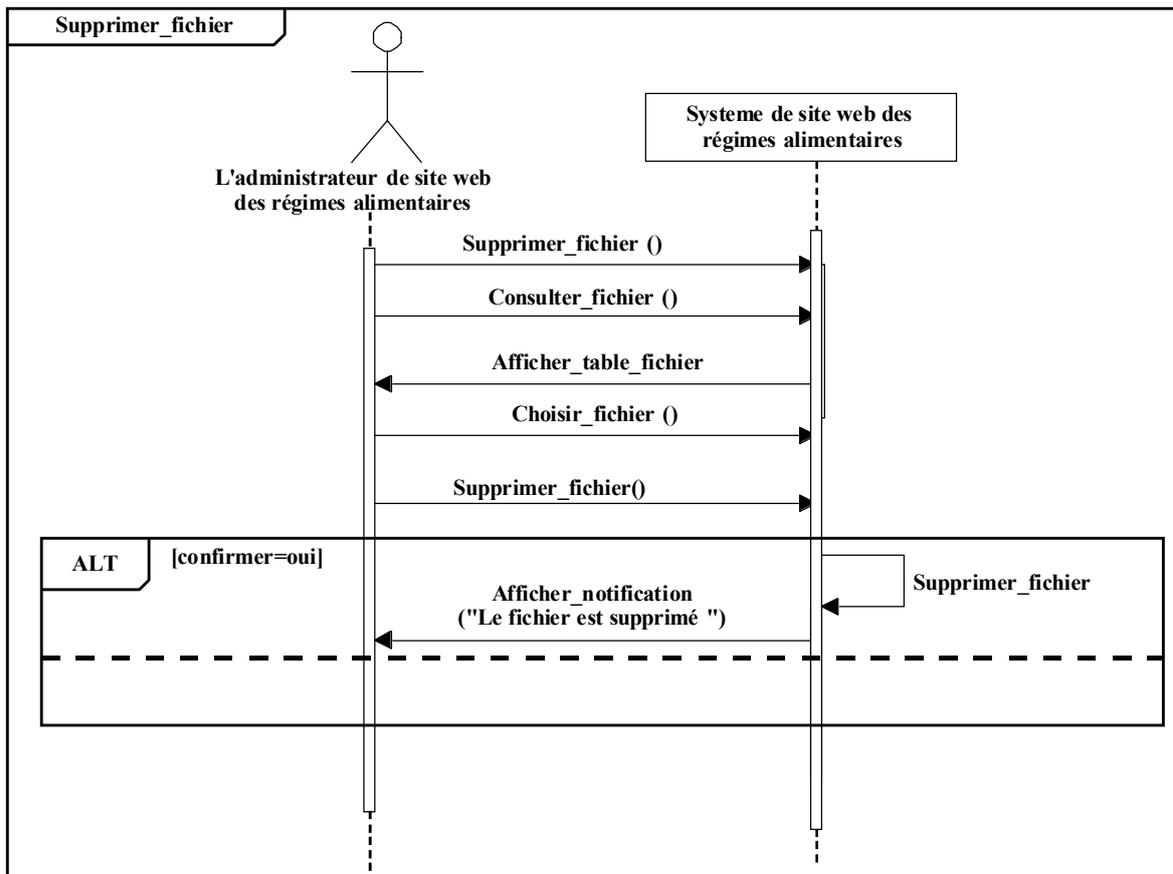


Figure 31. Diagramme de séquence système cas «Supprimer fichier»

5.3.2.6. Cas d'utilisation « Consulter fichier»

a. Fiche description

Sommaire d'identification
Titre : Consulter_fichiers. But : Afficher les fichiers existents dans la base des données. Résumé : Afficher les fichiers existents dans la base des données. Acteur : L'administrateur de site web des régimes alimentaires.
Description des Enchaînements
Pré condition <ul style="list-style-type: none">• L'administrateur doit être s'authentifier. Post condition <ul style="list-style-type: none">• La liste des fichiers est affichée. Scénario nominal <p>Ce cas d'utilisation commence lorsque l'administrateur demande au système de site web des régimes alimentaires une consultation des fichiers.</p> <ul style="list-style-type: none">• Le système affiche le tableau des fichiers.

b. Diagramme de séquence

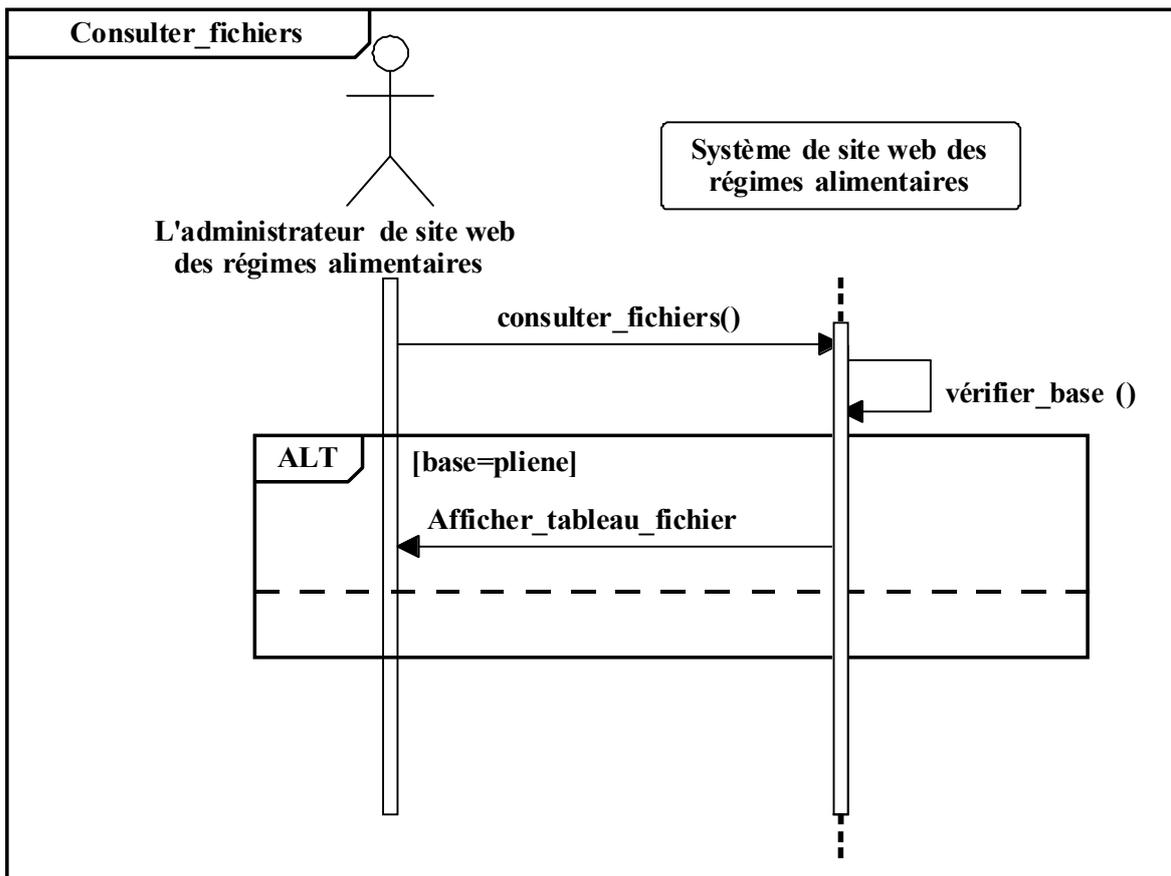


Figure 32. Diagramme de séquence système cas «Consulter fichier»

5.3.3. Au niveau de l'interaction entre les trois sous-systèmes

(Le sous-système de fournisseur de régime alimentaire, Le sous-système de l'utilisateur de régime alimentaire et l'UDDI)

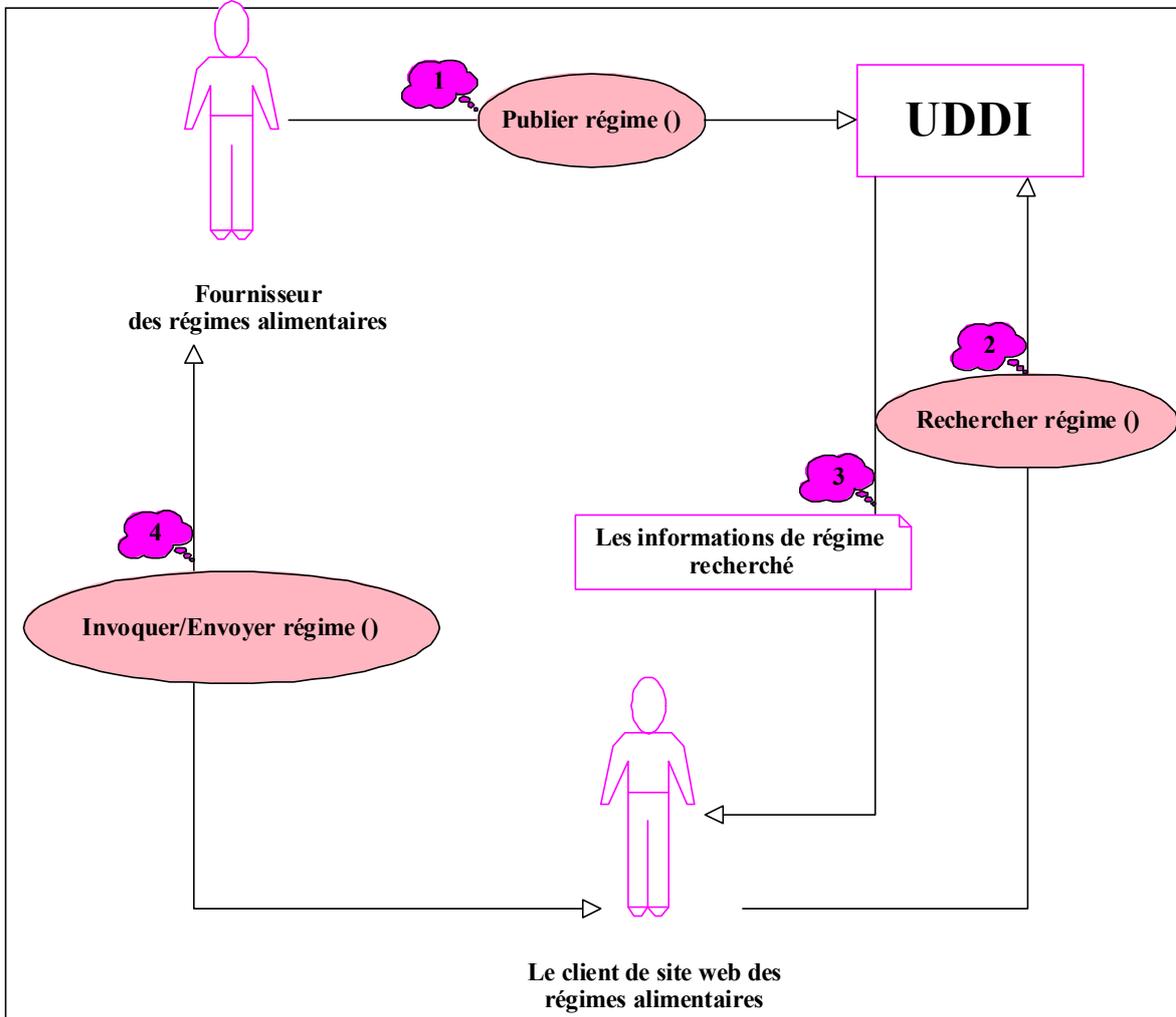


Figure 33. L'interaction entre les trois sous-systèmes

5.3.3.1. Cas d'utilisation « Rechercher un régime alimentaire »

a. Fiche description

Sommaire d'identification
Titre : Rechercher_régime. But : Trouver les informations concernant un régime alimentaire. Résumé : Trouver les informations concernant un régime alimentaire au niveau de l'UDDI. Acteur : Le client de site web des régimes alimentaires.
Des descriptions des enchainements
Pré conditions <ul style="list-style-type: none">• Le client de site web des régimes alimentaires doit être authentifié. Post condition <ul style="list-style-type: none">• Réception des informations d'un régime alimentaire envoyées par l'UDDI. Scénario nominal <p>Ce cas d'utilisation commence lorsque le client de site web des régimes alimentaires lance la recherche d'un régime alimentaire au niveau de l'UDDI.</p> <ol style="list-style-type: none">1. Le système de site web des régimes alimentaires affiche un formulaire pour le remplir.2. Le client de site web des régimes alimentaire remplit le formulaire et le valide.3. Le système de site web des régimes alimentaire envoie la requête vers l'UDDI.4. L'UDDI vérifie l'existence des informations.5. Si les informations existent dans l'UDDI il envoie les informations concernant le régime demandé.

b. Diagramme d'activité

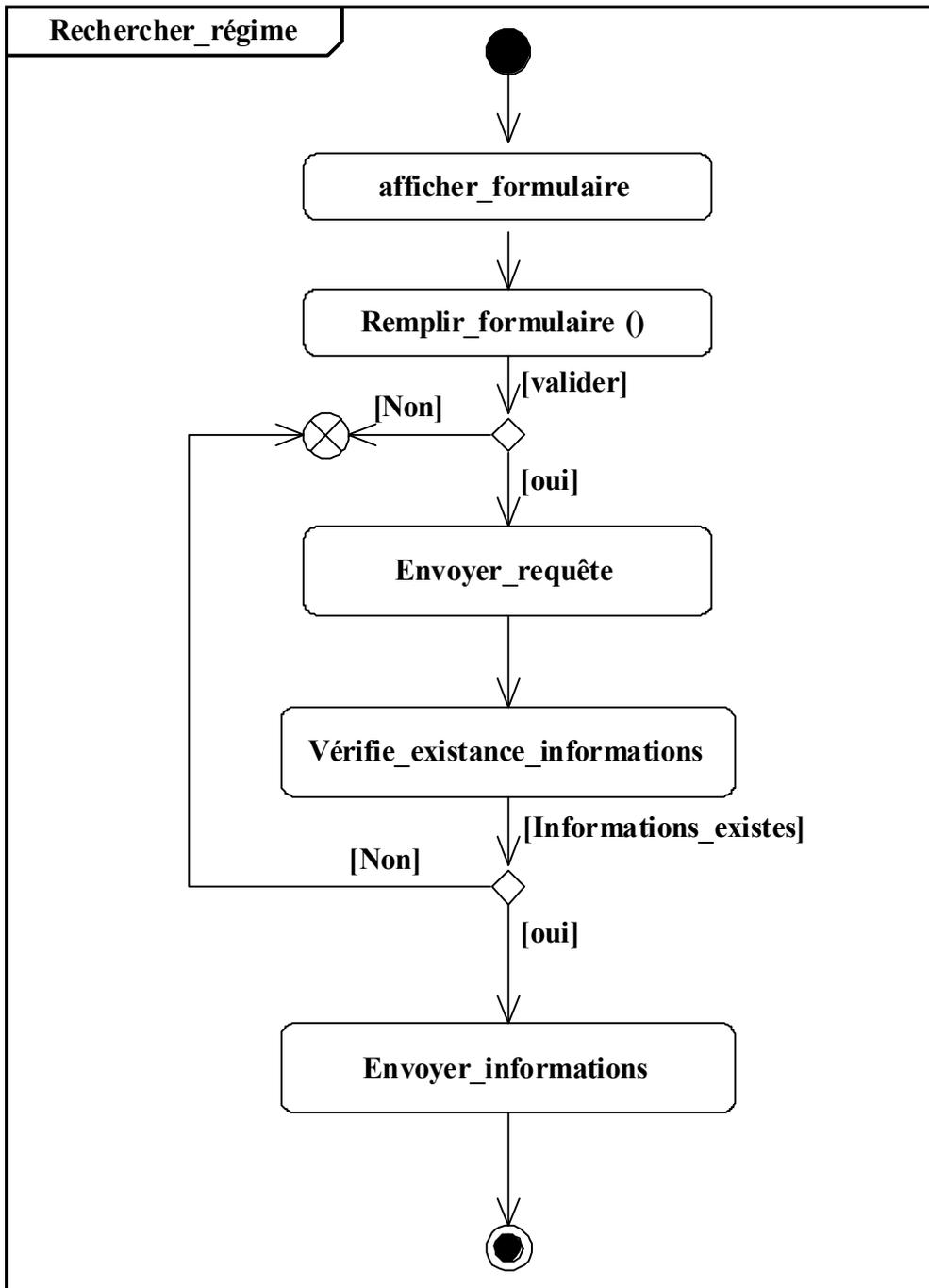


Figure 34. Diagramme d'activité cas «Rechercher régime»

c. Diagramme de séquence

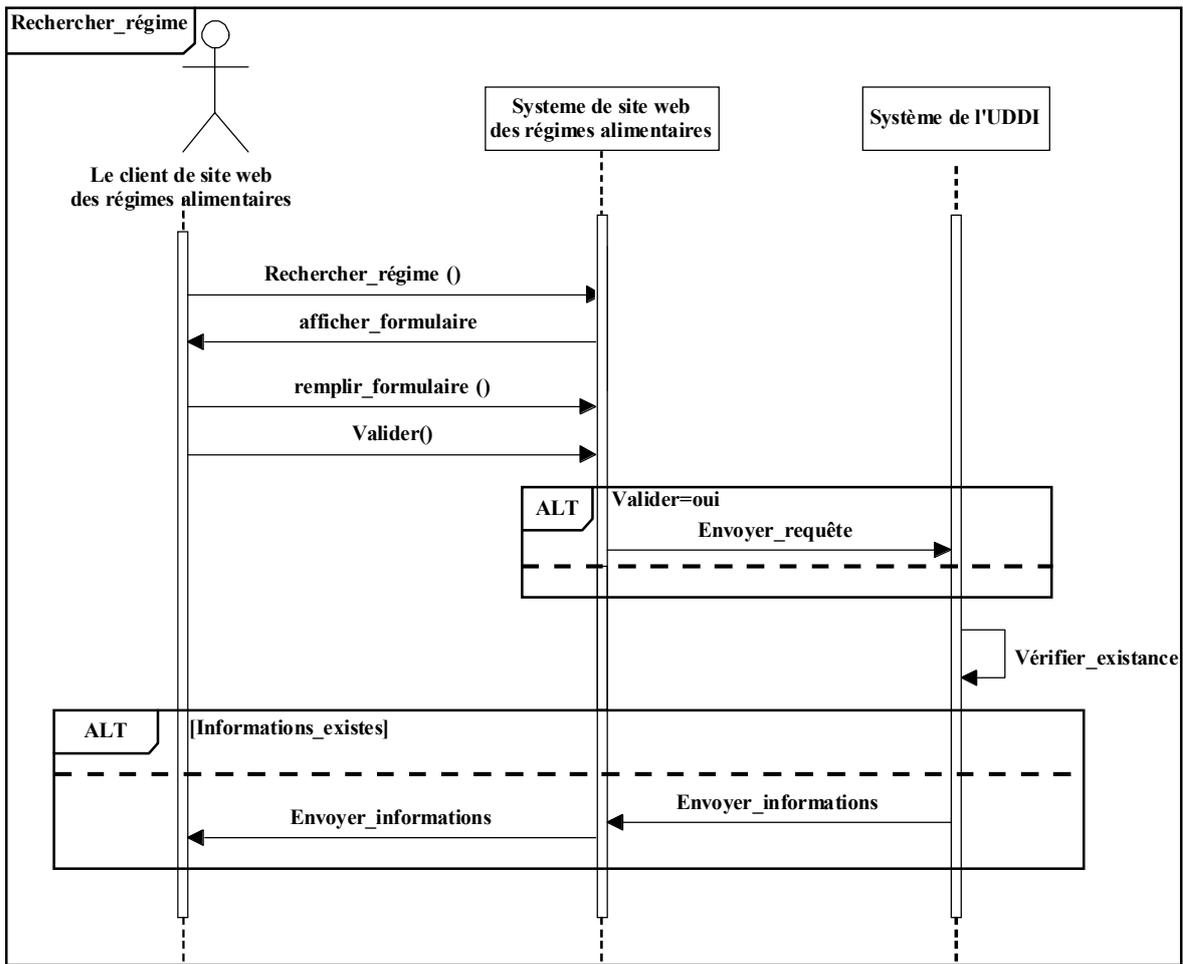


Figure 35. Diagramme de séquence système cas «Rechercher»

5.3.3.4. Cas d'utilisation « Invoquer régime »

a. Fiche description

Sommaire d'identification
Titre : Invoquer_régime. But : Envoyer un régime au client de site web des régimes alimentaires, après une demande vers le fournisseur des régimes alimentaires. Résumé : Le client de site web des régimes alimentaires invoque un régime. Acteur : le client de site web des régimes alimentaires.
Descriptions des enchainements
Pré conditions <ul style="list-style-type: none">• Le client de site web des régimes alimentaires doit être authentifié.• Le régime est publié dans l'UDDI,• Le client a recherché le régime auprès de l'UDDI. Post condition <ul style="list-style-type: none">• Le régime est envoyé vers le client de site web des régimes alimentaires. Scénario nominal <p>Ce cas d'utilisation commence lorsque le client de site web des régimes alimentaires invoque un régime alimentaire auprès système de fournisseur des régimes alimentaires.</p> <ol style="list-style-type: none">1. Le système de site web des régimes alimentaires affiche un formulaire pour le remplir.2. Le client de site web des régimes alimentaire remplit le formulaire et le valide.3. Le système de site web des régimes alimentaire envoie la requête vers le système des fournisseurs des régimes alimentaires.4. Le système des fournisseurs régimes alimentaires extrait le régime alimentaire et l'envoie.

b. Diagramme d'activité

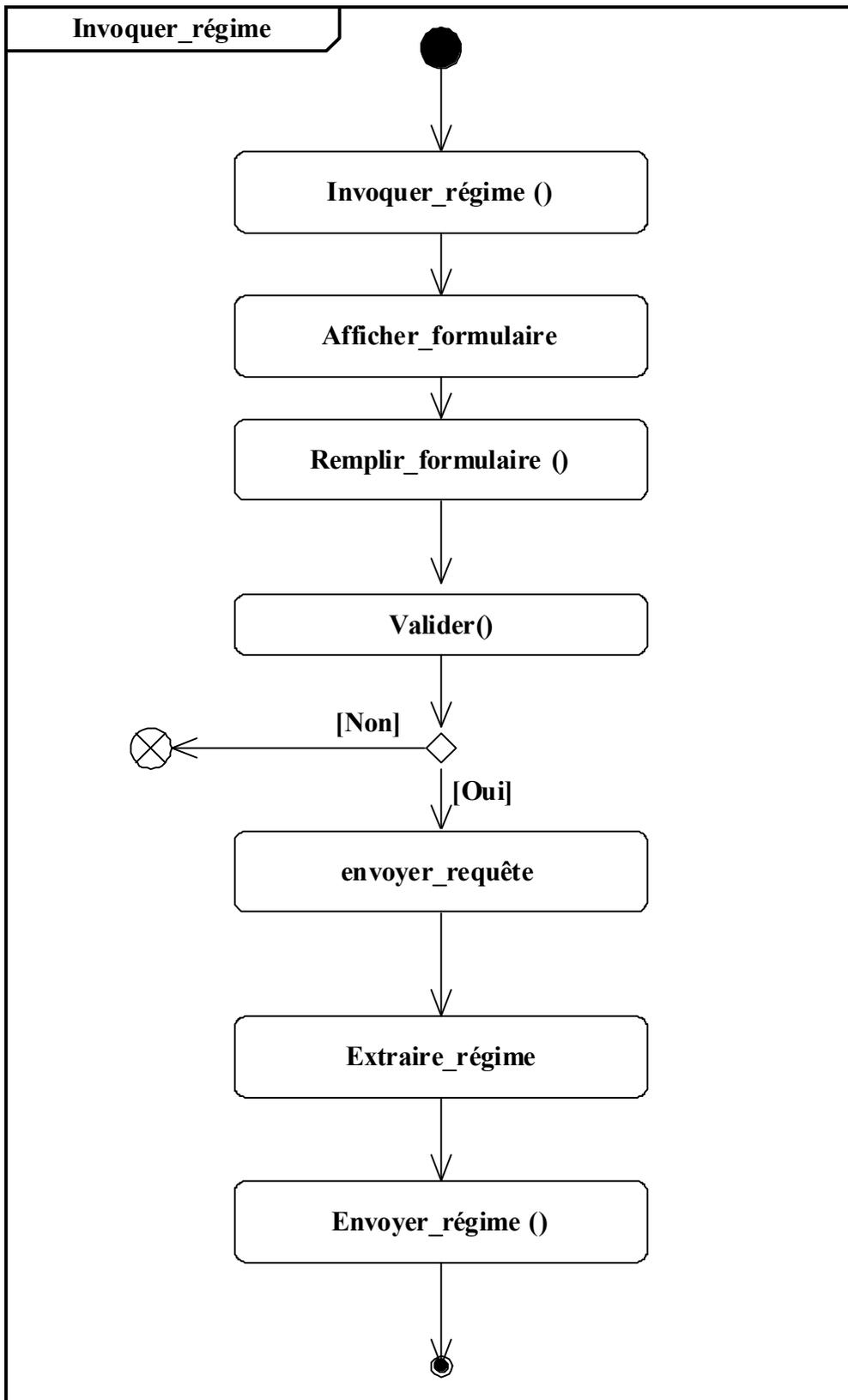


Figure 36. Diagramme d'activité cas «Invoquer régime»

c. Diagramme de séquence

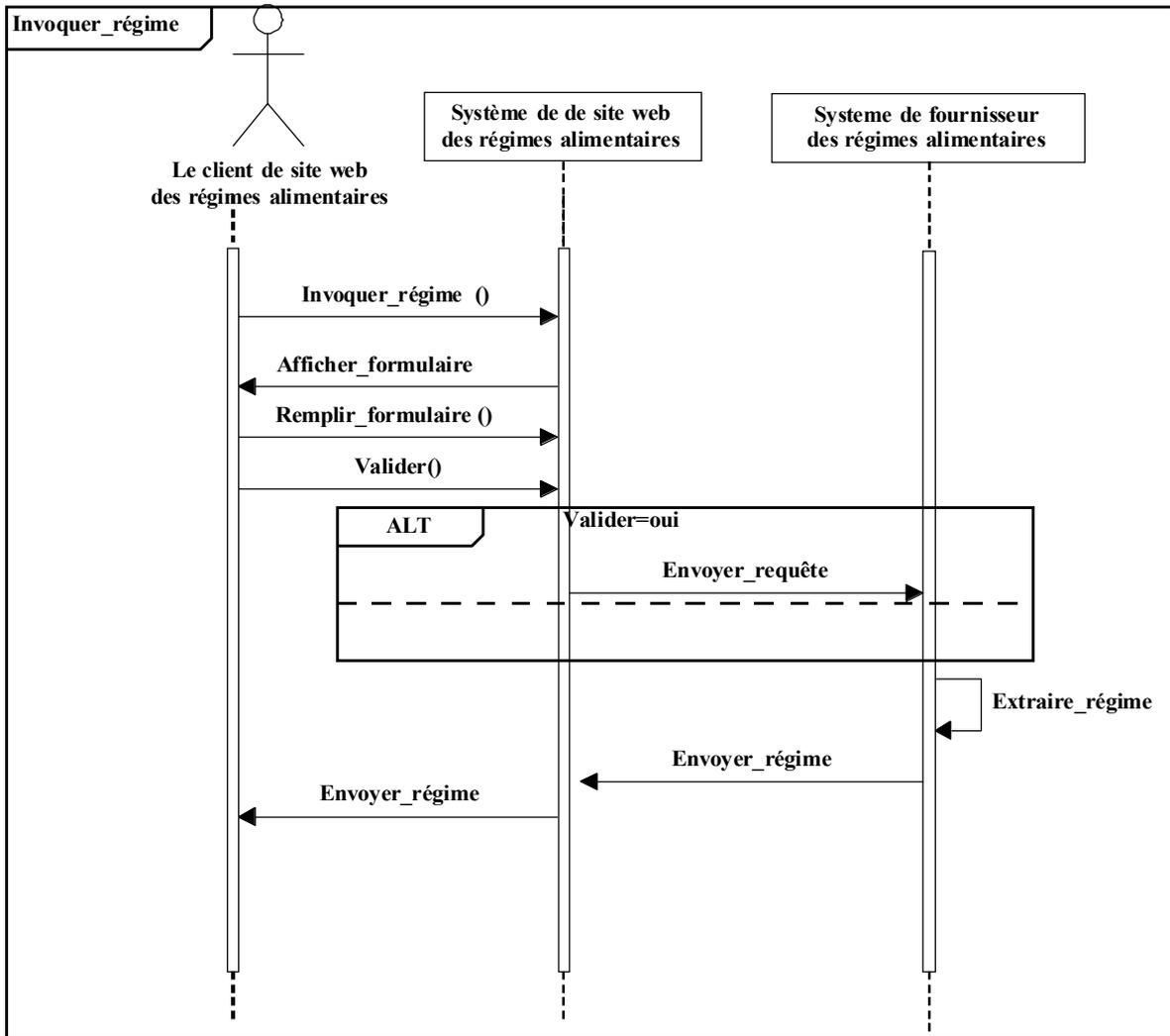


Figure 37. Diagramme de séquence système cas «Invoquer régime».

Introduction

On va s'intéresser à la branche droite du cycle en Y qui est « la capture des besoins techniques » en couvrant les contraintes qui ne traitent pas la description applicative.

Nous choisissons lors de cette phase l'environnement de travail ainsi que l'architecture globale utilisée pour notre système. La capture des besoins techniques se présente selon deux points de vue :

- ✓ **La spécification technique du point de vue matériel.**
- ✓ **La capture des spécifications logicielles.**

6.1. Spécification technique du point de vue matériel

6.1.1. Configuration matérielle

- **Caractéristiques techniques**
- ❖ **Au niveau des fournisseurs des régimes alimentaires**

Nous avons utilisé trois ordinateurs :

a. Un ordinateur serveur : dont les caractéristiques suivantes :

Processeur : Intel(R) Core (TM) i3 CPU @ 2.10GHz.

Mémoire installée (RAM) : 4,00 Go (3,85 G utilisable).

Type du système : Système d'exploitation 64 bits, processeur x64.

b. Un ordinateur client :

Processeur : Intel(R) Core (TM) Pentium (R) @ 21274@ 1,90GHz.

Mémoire installée (RAM) : 2,00 Go (1,88 G utilisable).

Type du système : Système d'exploitation 32 bits, processeur x64.

- ❖ **Au niveau de l'utilisateur des régimes alimentaires**

Processeur : Intel(R) Core (TM) i3 CPU @ 2.10GHz.

Mémoire installée (RAM) : 4,00 Go (3,85 G utilisable).

Type du système : Système d'exploitation 64 bits, processeur x64.

- **Un réseau dont les caractéristiques**
- ❖ **Au niveau des fournisseurs des régimes alimentaires**
- ✓ Débit 100 Mbit/s (simple duplexe)
- ✓ Fast Ethernet
- ✓ Protocole TCP/IP
- ❖ **Au niveau des utilisateurs es régimes alimentaires des régimes alimentaires**

Le réseau utilisé normalement est l'internet avec un protocole http, mais dans notre cas nous travaille dans le local host.

6.1.2. Spécification d'architecture web service

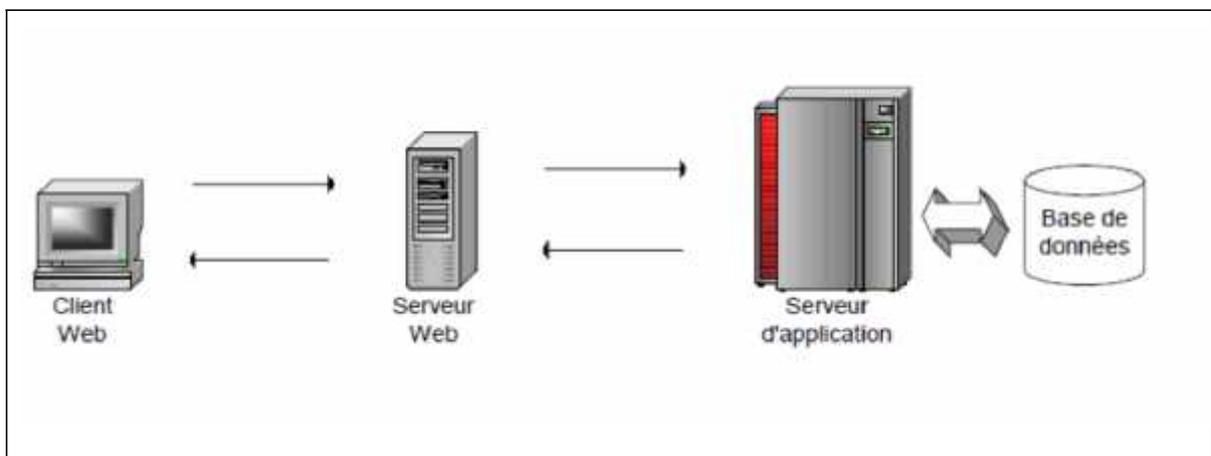


Figure 1. Organisation du model de spécification logicielle

6.2. Capture des spécifications logicielles

Ce sont des fonctionnalités techniques que le système va assurer à l'utilisateur indépendamment des fonctionnalités métier.

6.2.1. Les exploitants

Appelés aussi « acteurs techniques » du système, ils sont les acteurs qui bénéficient des fonctionnalités techniques du système :

- **Au niveau de fournisseur des régimes alimentaires**

Les exploitants sont l'administrateur de fournisseur des régimes alimentaires et le fournisseur des régimes alimentaires.

- **Au niveau de site web des régimes alimentaires**

L'exploitant est l'administrateur et le client de site web des régimes alimentaires.

6.2.2. Identification des cas d'utilisation techniques

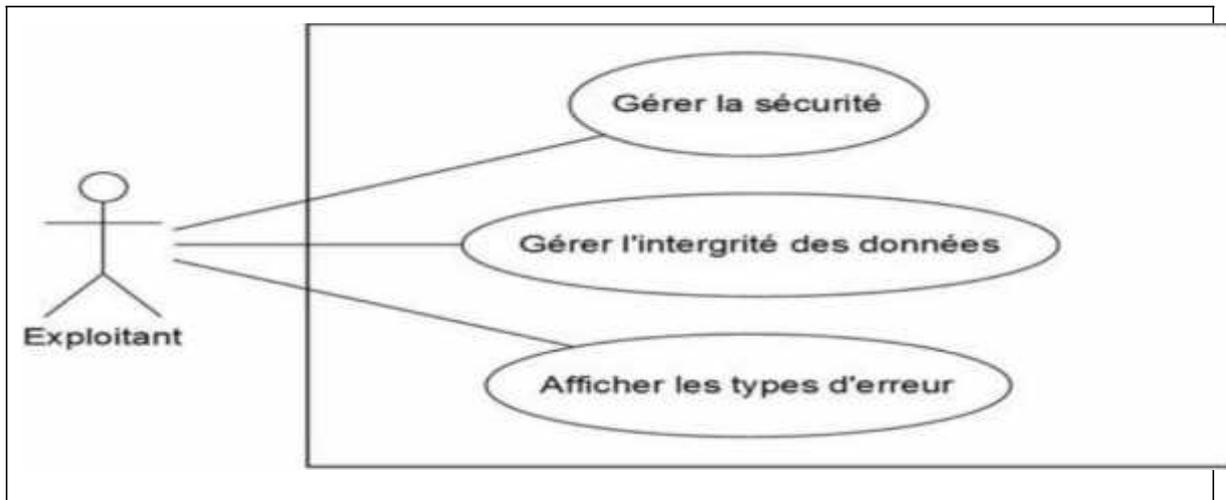


Figure 2. Diagramme de cas technique

6.2.3. Description des cas d'utilisation techniques

Pour des raisons de délais du projet et d'espace dans ce mémoire, nous donnons l'aspect technique «Gérer la sécurité ».

6.2.3.1. Cas d'utilisation : «Gérer la sécurité »

➤ S'authentifier

a. Fiche description

Intention : l'utilisateur désire accéder au système avec sécurité.

Action : l'utilisateur s'authentifie pour accéder au système.

Identification des besoins :

1. authentification

Précondition :

- L'exploitant est inscrit dans le système

Scénario nominal :

- L'exploitant saisit ses identifiants dans le formulaire d'authentification.
- Le système le dirige vers l'espace qui lui est approprié.

2. Gestion des comptes d'utilisateur

Pré condition :

- L'exploitant est inscrit dans le système.

Scénario nominal

- L'exploitant saisir ses identifications dans la page de gestion d'un compte
- Le système confirme la mise à jour des comptes.

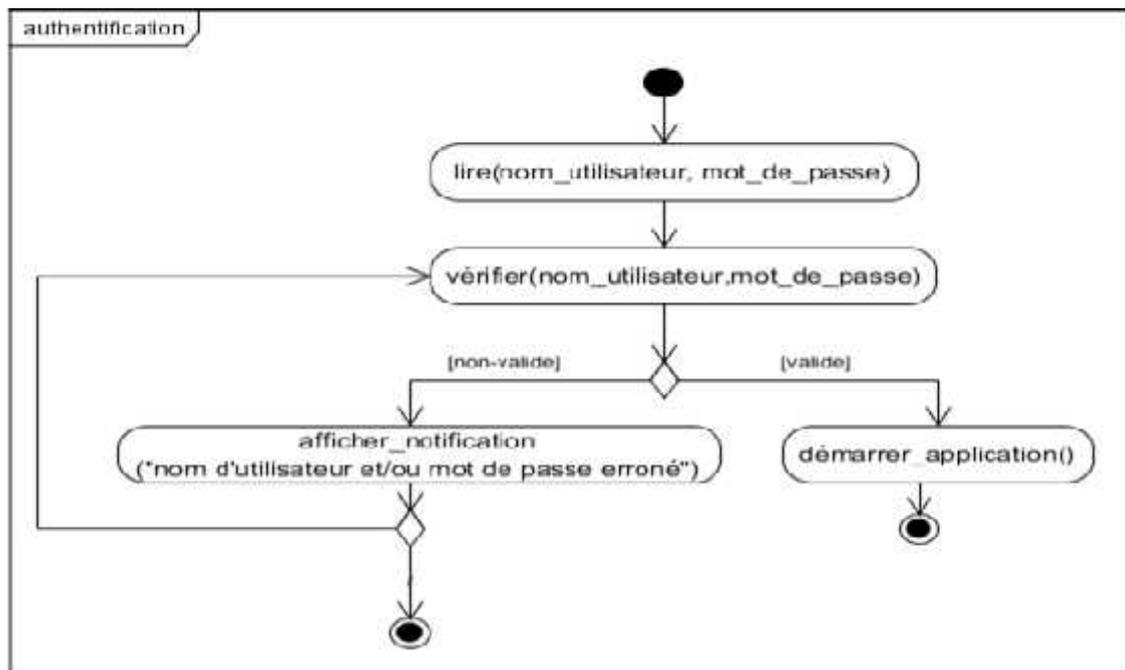
b. Diagramme d'activité

Figure 3. Diagramme d'activité de cas d'utilisation technique « authentification »

nom_utilisateur = {nom de fournisseur des régimes alimentaires, nom de l'administrateur des fournisseur des régimes alimentaires, le client de site web des régimes alimentaires}

6.2.3.2. Cas d'utilisation : « Gérer l'intégrité des données »

➤ Vérifier Champ formulaire

a. Fiche descriptive

Sommaire d'identification

Intention : L'utilisateur désire introduire des données dans le système.

Action : L'utilisateur saisie les données.

Identification du besoin : ajouter fournisseur, modifier fournisseur, ajouter régime alimentaire, modifier régime alimentaire, inscrire visiteur, modifier un compte client, ajouter fichier, modifier fichier, ajouter. annonce.

Des descriptions des enchainements

Pré condition :

- L'exploitant est authentifié et accéder à un formulaire.

Scénario nominal :

- L'exploitant saisit les données.
- Le système contrôle chaque champ, s'il n'est pas vide, ou invalide.

Scénario alternatif :

- Le système affiche une notification « les champs non valide ».

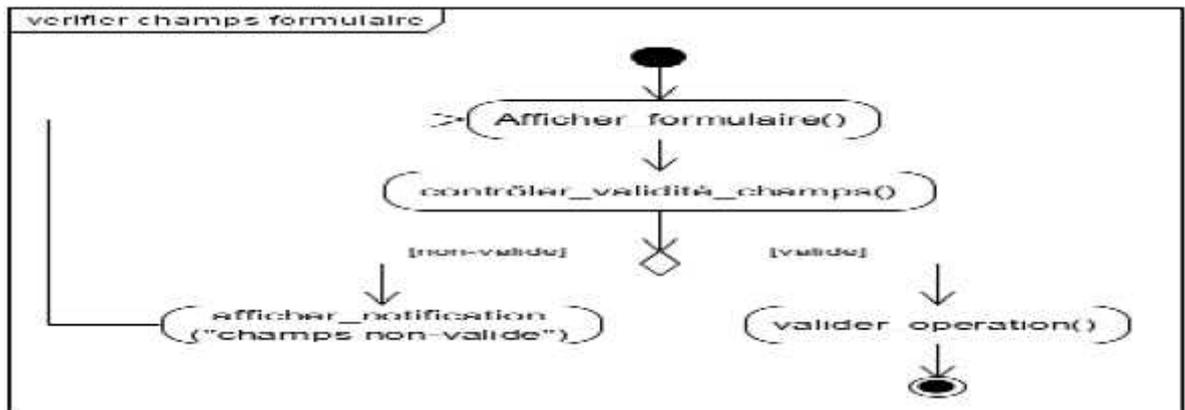
b. Diagramme d'activité

Figure 4. Diagramme d'activité de cas d'utilisation technique « vérifier champs formulaire »

6.3. Organisation du modèle de spécification logicielle

Le modèle de spécification logiciel est conditionné par l'architecture déployé. Dans notre cas avec l'architecture 3-tiers, notre système est séparé trois couches logicielles :

- **La présentation des données** : correspondant à l'affichage, le dialogue avec l'utilisateur.
- **Le traitement métier** : correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative, c'est à ce niveau que se trouvent toutes les règles de gestion, et toute la logique de la démarche.
- **L'accès aux données** : correspondant aux données qui sont destinées à être conservées.

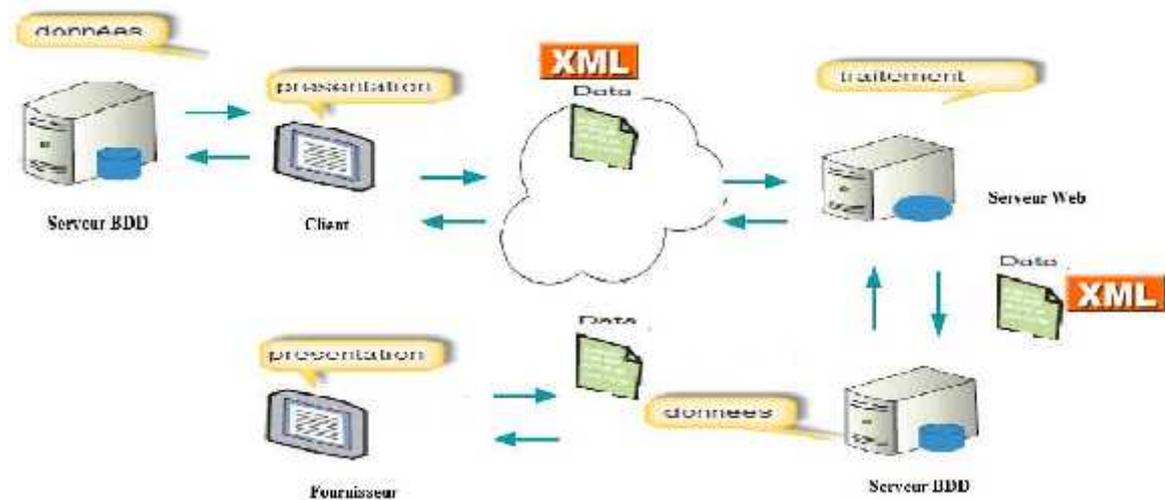


Figure 5. Organisation du modèle de spécification logicielle

Conclusion

Au cours de ce chapitre, l'architecture physique a été choisie selon l'environnement adopté, on a pris en compte toutes les contraintes techniques et logicielles pour le choix le plus adapté de notre architecture. Dans le prochain chapitre nous allons présenter La phase d'analyse.

Introduction

La phase d'analyse succède à la capture des besoins fonctionnels et un a pour but d'insérer une première phase sur le but de l'étape d'analyse. Elle est constituée de trois activités :

- Le découpage en catégories ;
- Le développement du modèle statique ;
- Le développement du modèle dynamique.

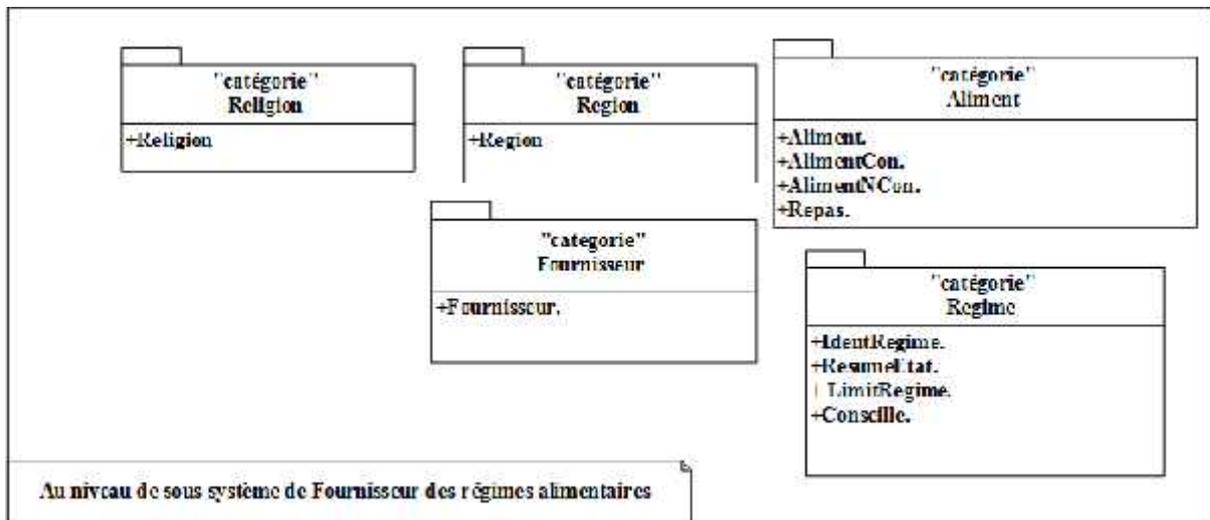
7.1. Découpage en catégories

Cette phase utilise la notion de package pour définir des catégories de classes d'analyse et découper le modèle UML en blocs logiques les plus indépendants possibles.

- **Définition** : une catégorie consiste en un regroupement logique de classes à forte cohérence interne et faible couplage externe.

Le découpage en catégories se fait en trois étapes :

7.1.1. La répartition des classes candidates en catégories



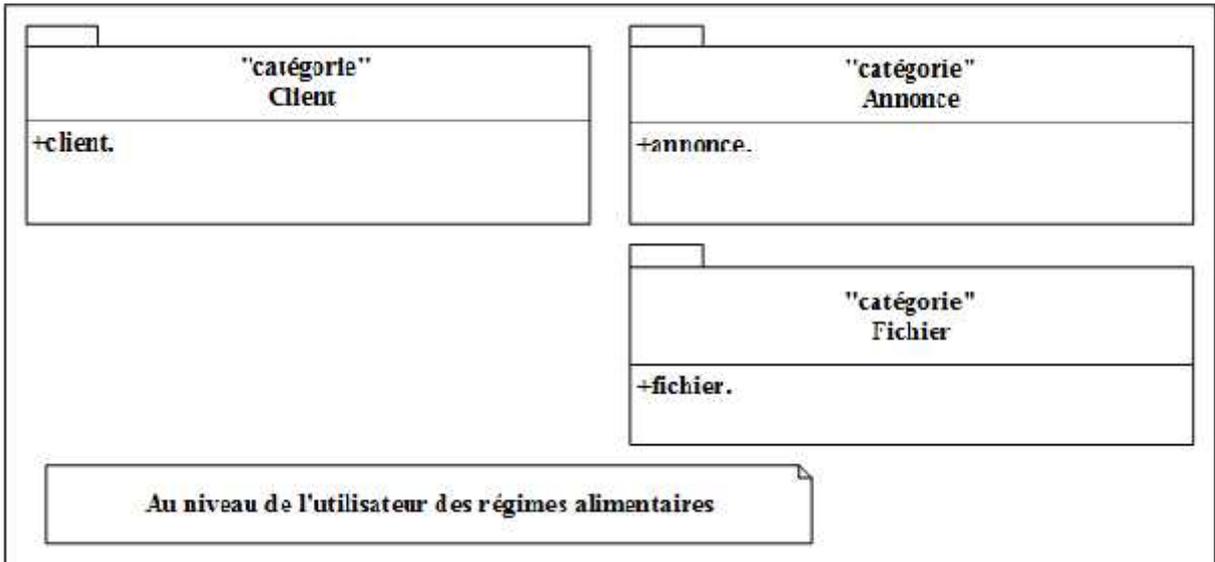
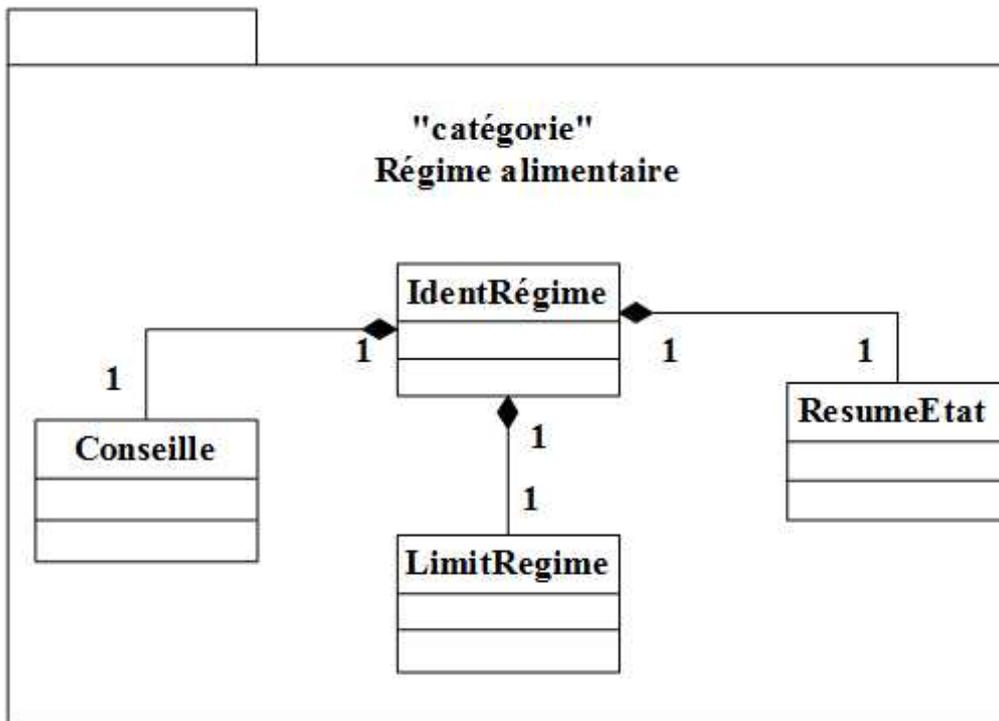


Figure 1. Découpage en catégorie de notre système.

7.1.2. Elaboration des diagrammes de classes préliminaires par catégorie



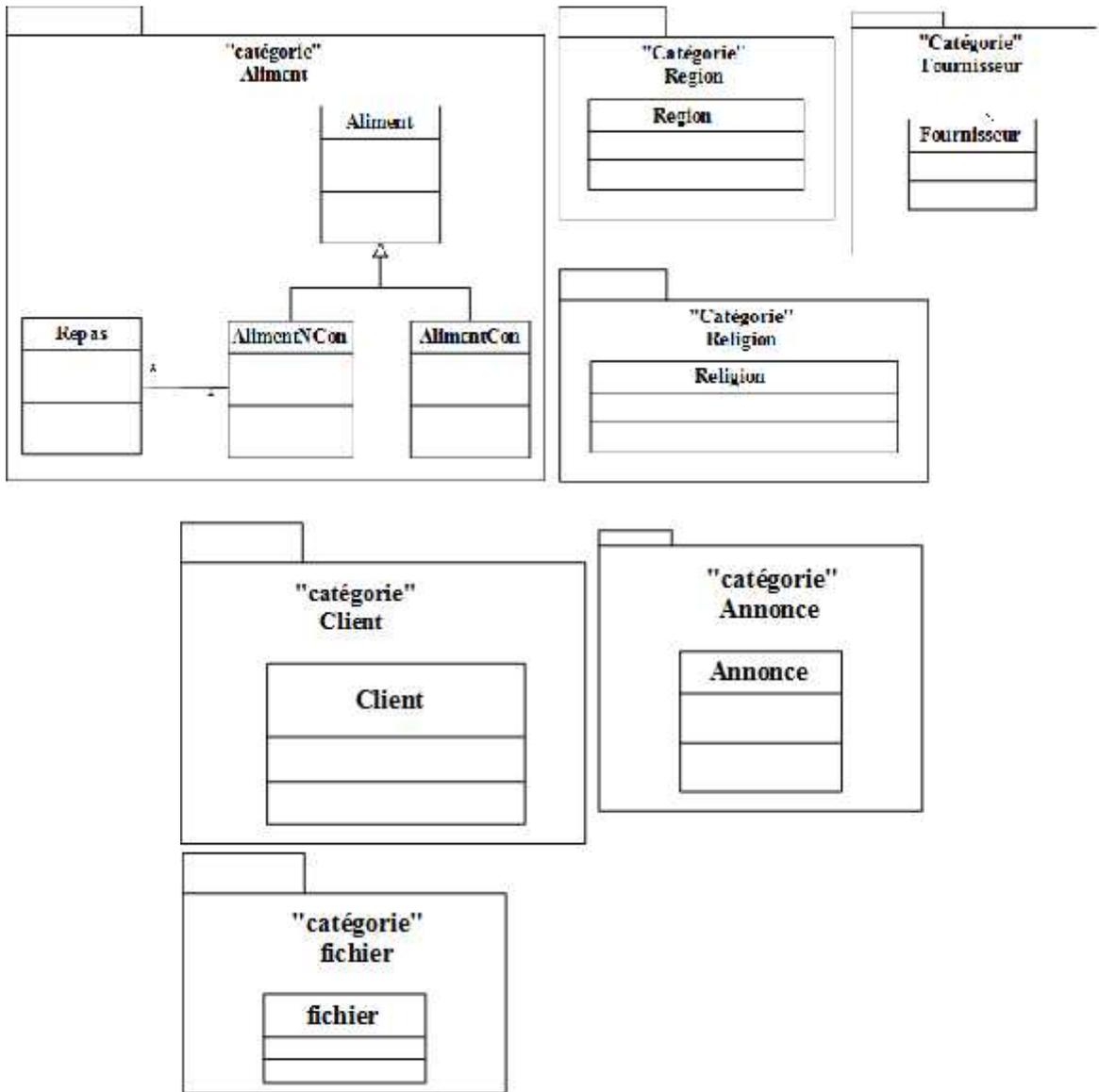


Figure 2. Découpage des classes de chaque catégorie.

7.1.3. Dépendance entre catégories

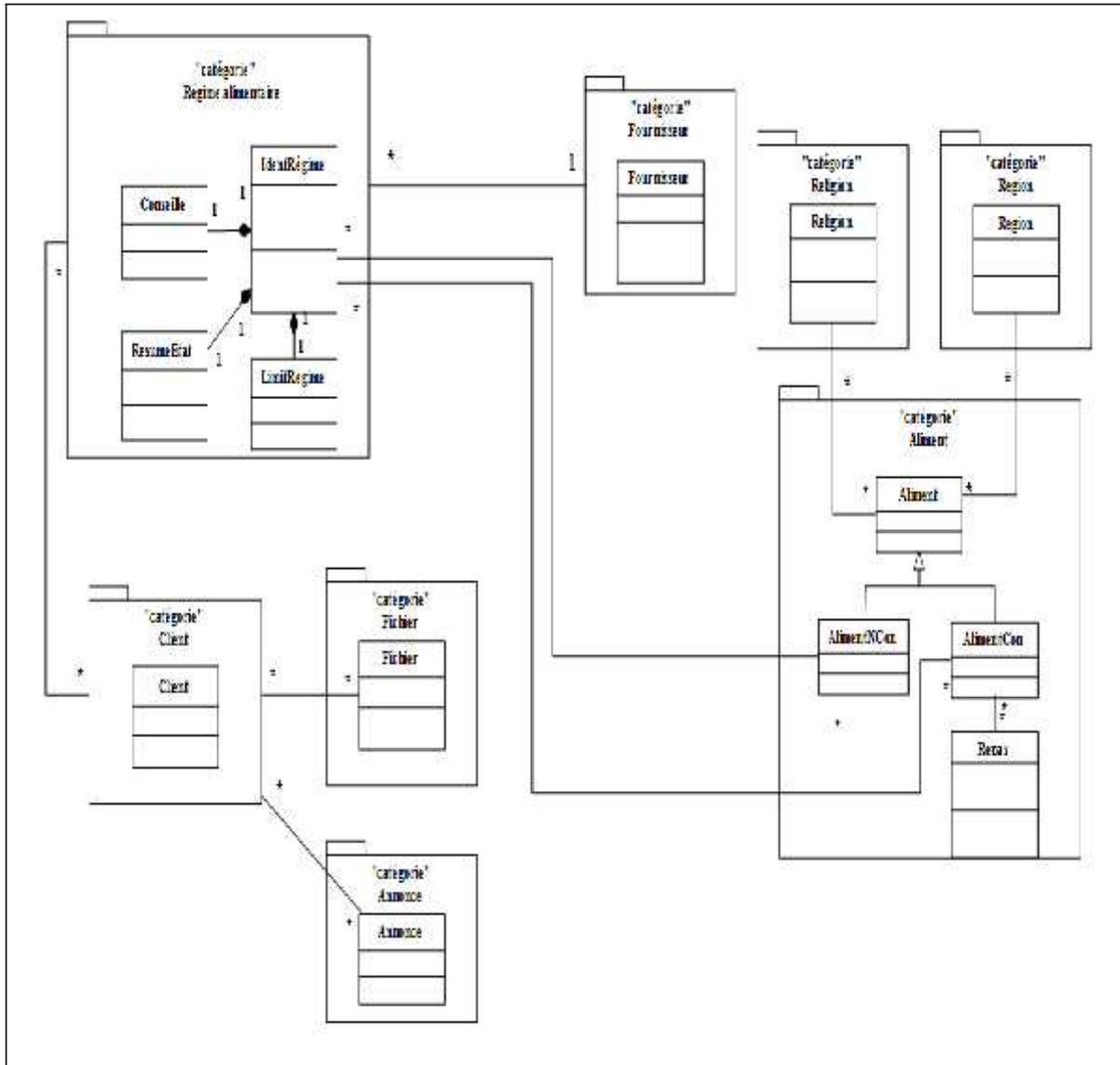
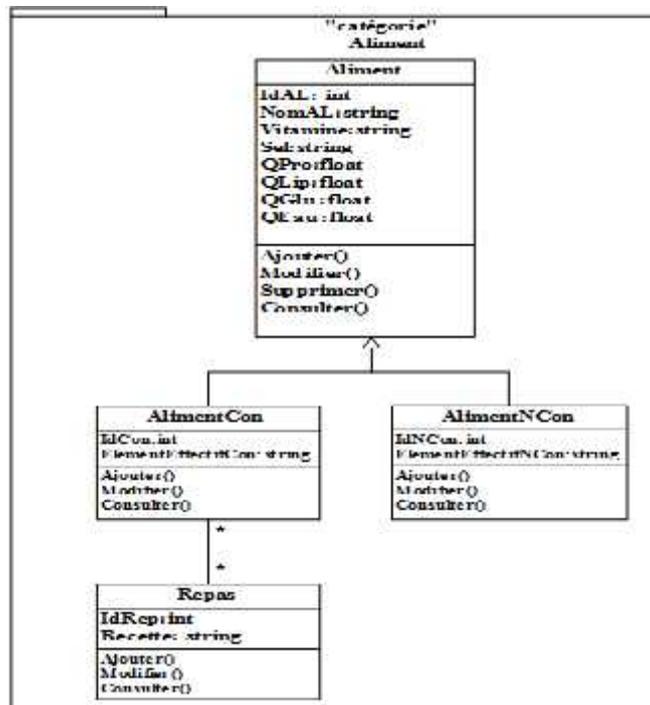
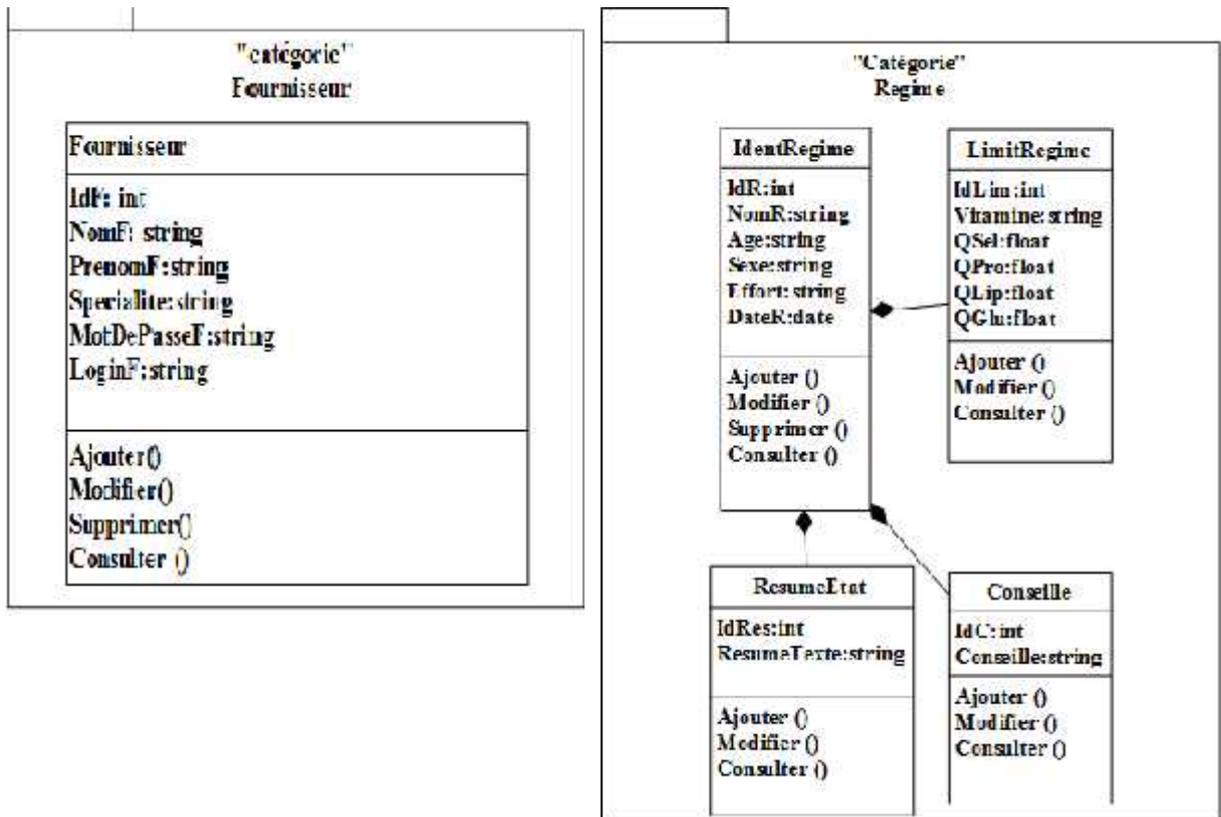


Figure 3. Dépendances entre les catégories.

7.2. Développement du modèle statique



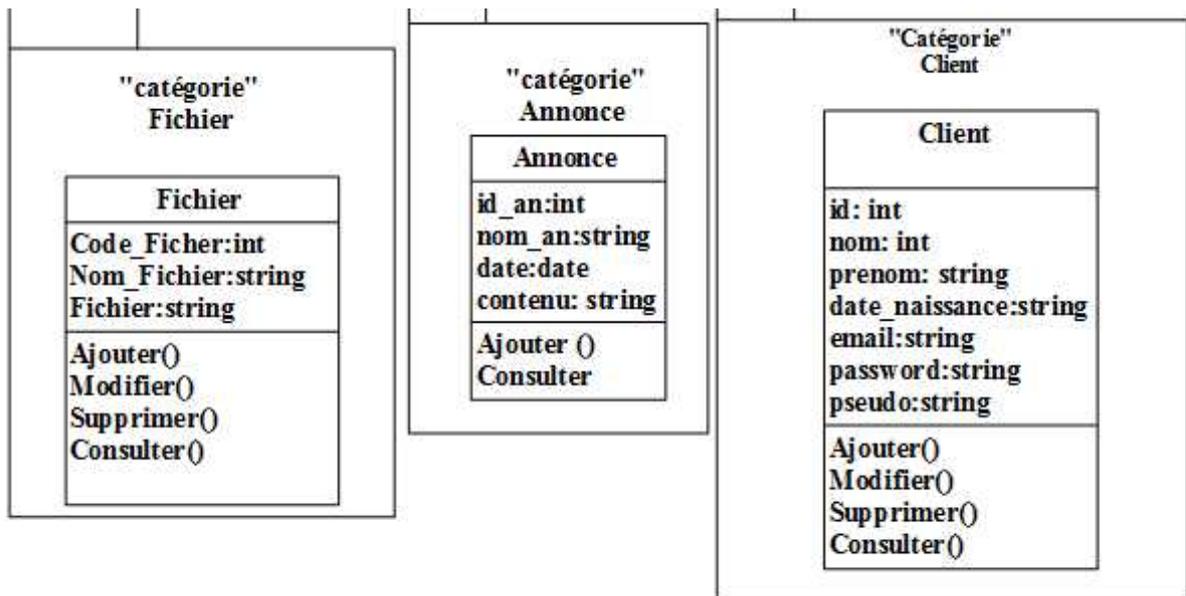


Figure 4. Diagramme de classe de chaque catégorie

7.3. Développement du modèle dynamique

Le développement du modèle dynamique est la troisième activité de l'étape d'analyse. Cette activité est en relation avec l'activité de modélisation statique.

Lors de cette étape, nous décrivons les différentes interactions entre les objets de notre application. En effet, nous avons utilisés le modèle dynamique : le diagramme de séquence détaillé.

7.3.1. Diagrammes de séquences

Le diagramme de séquence est un diagramme d'interaction entre les objets, qui met l'accent sur le classement des messages par ordre chronologique durant l'exécution du système. Un diagramme de séquence est un tableau dans lequel les objets sont rangés sur l'axe des abscisses et des messages par ordre d'apparition sur l'axe des ordonnées.

Il est utilisé pour représenter certains aspects dynamiques d'un système : dans le contexte d'une opération, d'un système, d'un sous-système, d'un cas d'utilisation (un scénario d'un cas d'utilisation) selon un point de vue temporel.

En effet dans cette phase, et après identification des cas d'utilisation, nous représentons à l'aide des diagrammes de séquences les différents cas d'utilisation.

7.3.1. 1. Diagramme de séquence du cas «Ajouter régime»

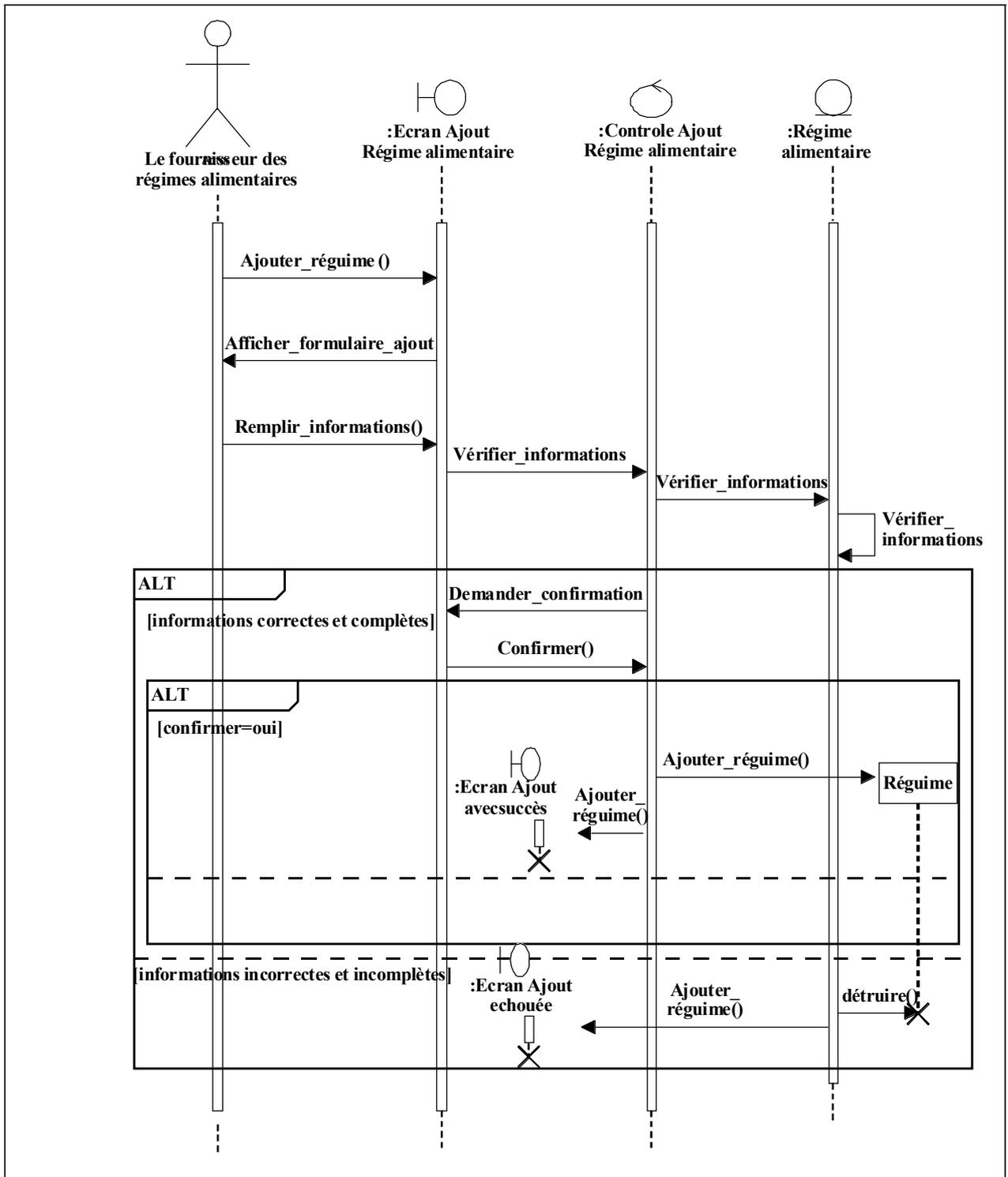


Figure 5. Diagramme de séquence du cas «Ajouter régime»

Remarque

Pour ajouter un régime il faut suivre les étapes suivantes :

1. Ajouter identification du régime.
2. Ajouter la limite du régime.
3. Ajouter le résumé de l'état de santé.
4. Ajouter le conseil.
5. Ajouter les aliments conseillés.
6. Ajouter les aliments non conseillés.
7. Ajouter le repas.

Chaque étape a les mêmes scénarios que nous avons déjà expliqués de régime, ainsi pour ajouter un régime il faut suivre toutes les étapes précédentes.

7.3.1. 2. Diagramme de séquence du cas «Modifier régime»

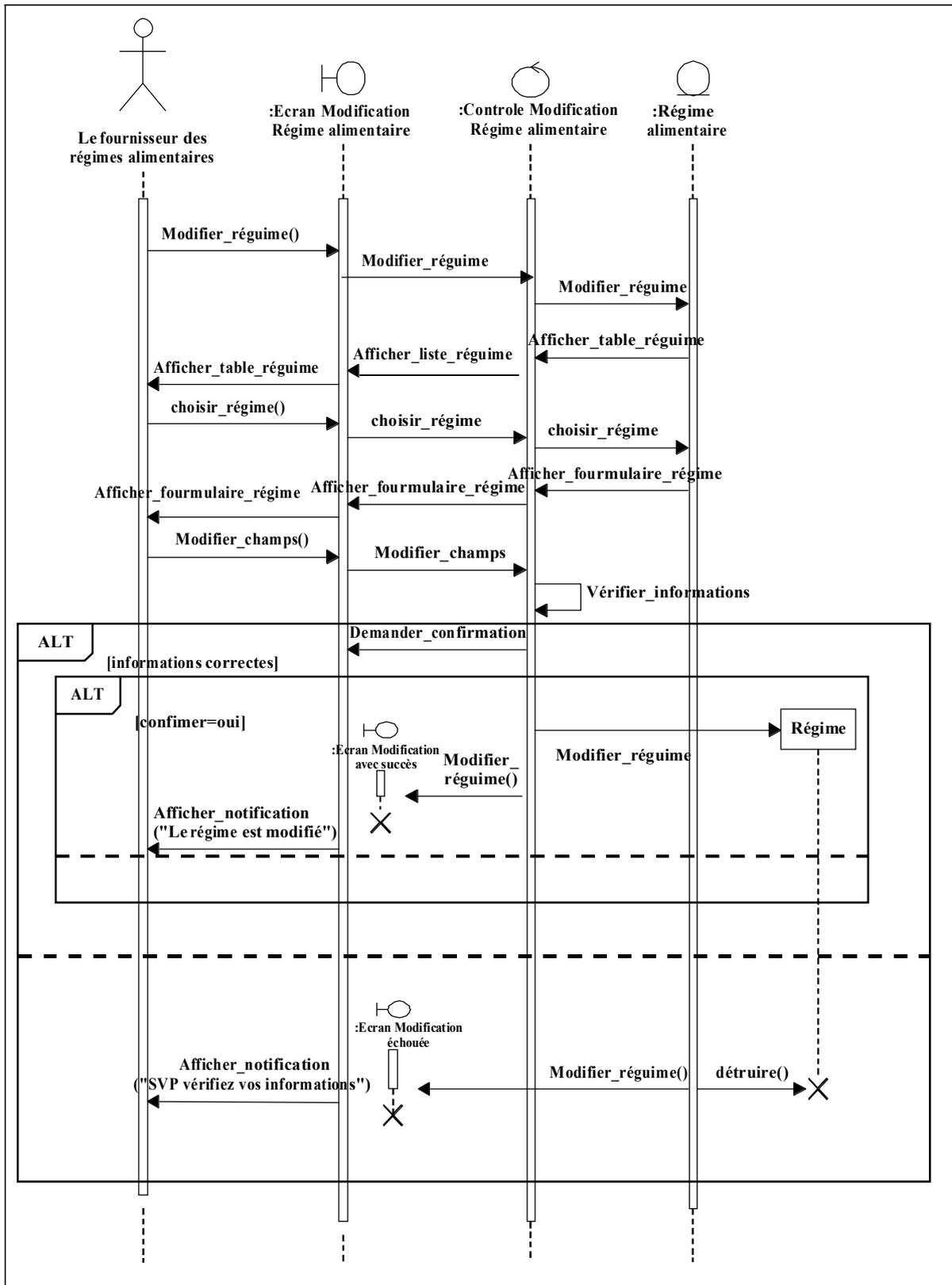


Figure 6. Diagramme de séquence du cas «Modifier régime»

Remarque

Pour modifier un régime il suffit de modifier dans l'une de ses composantes (un composante ou plusieurs)

1. Modifier identification du régime.
2. Modifier la limite du régime.
3. Modifier le résumé de l'état de santé.
4. Modifier le conseil.
5. Modifier les aliments conseillés.
6. Modifier les aliments non conseillés.
7. Modifier le repas.

7.3.1. 3. Diagramme de séquence du cas «Supprimer régime»

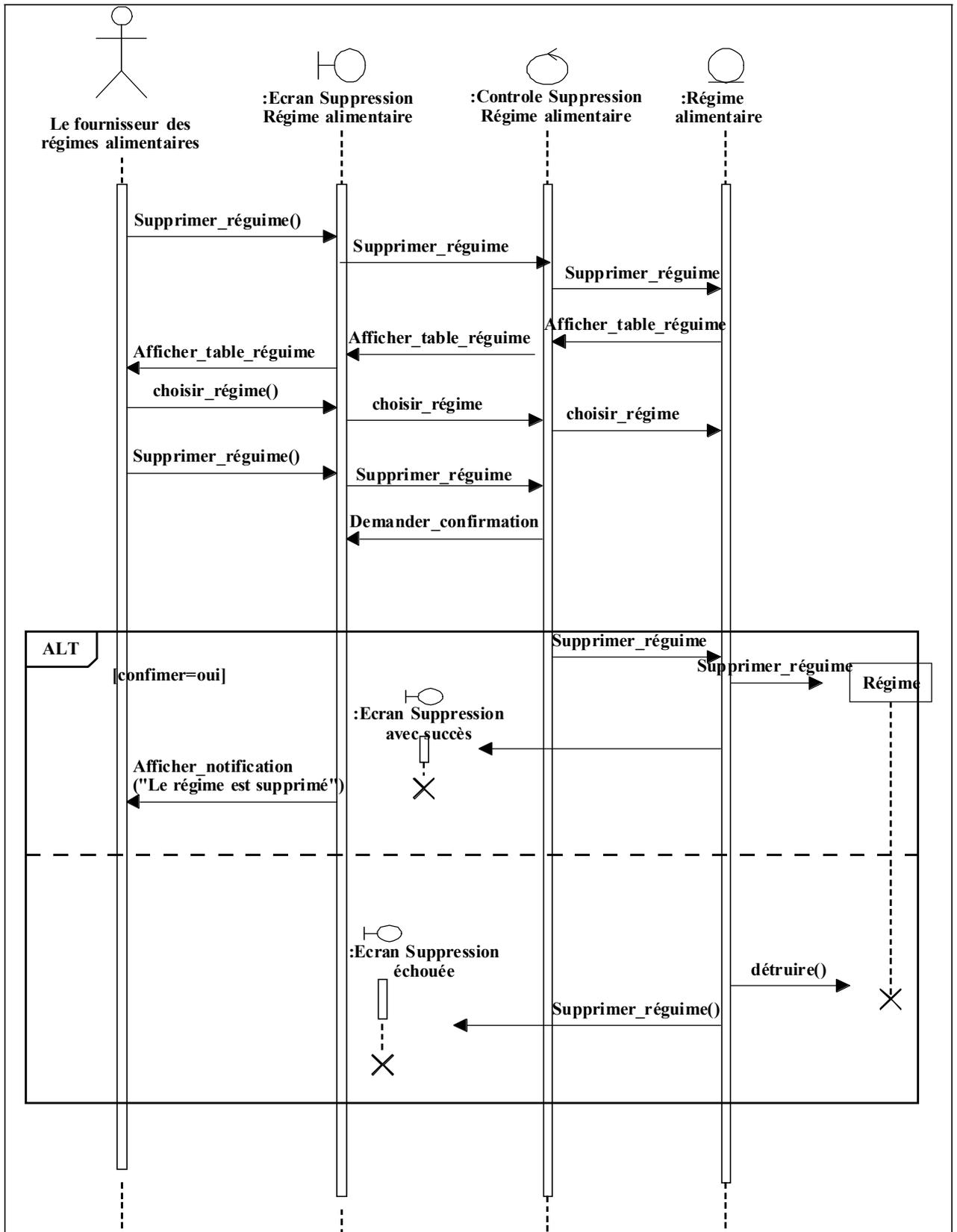


Figure 7 . Diagramme de séquence du cas «Supprimer régime»

Remarque

La suppression de régime indique que toutes ses composantes seront supprimées.

7.3.1.3. Diagramme de séquence du cas «Invoquer régime»

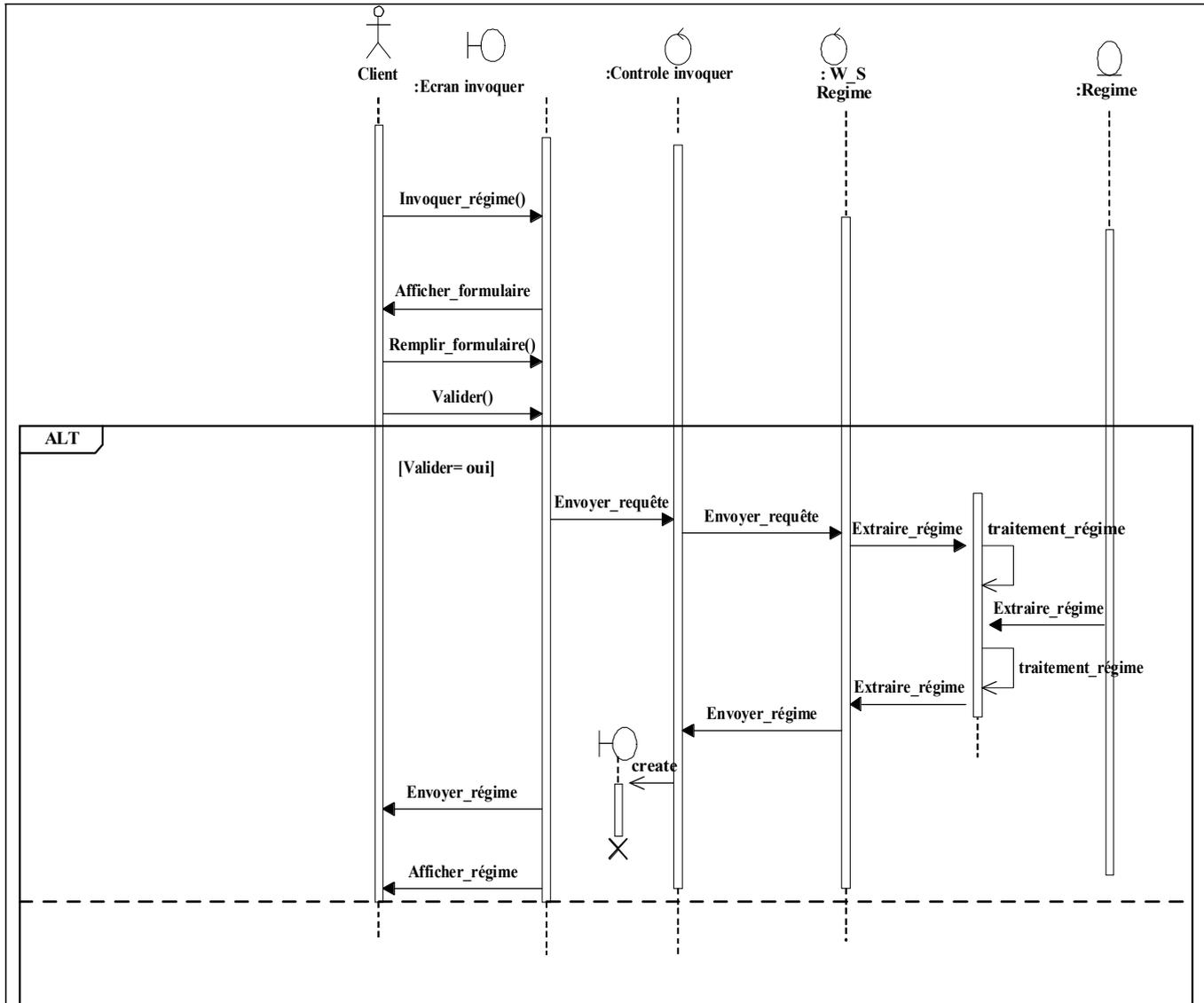


Figure 8. Diagramme de séquence du cas «Invoquer régime»

7.3.2. Diagramme état de transition

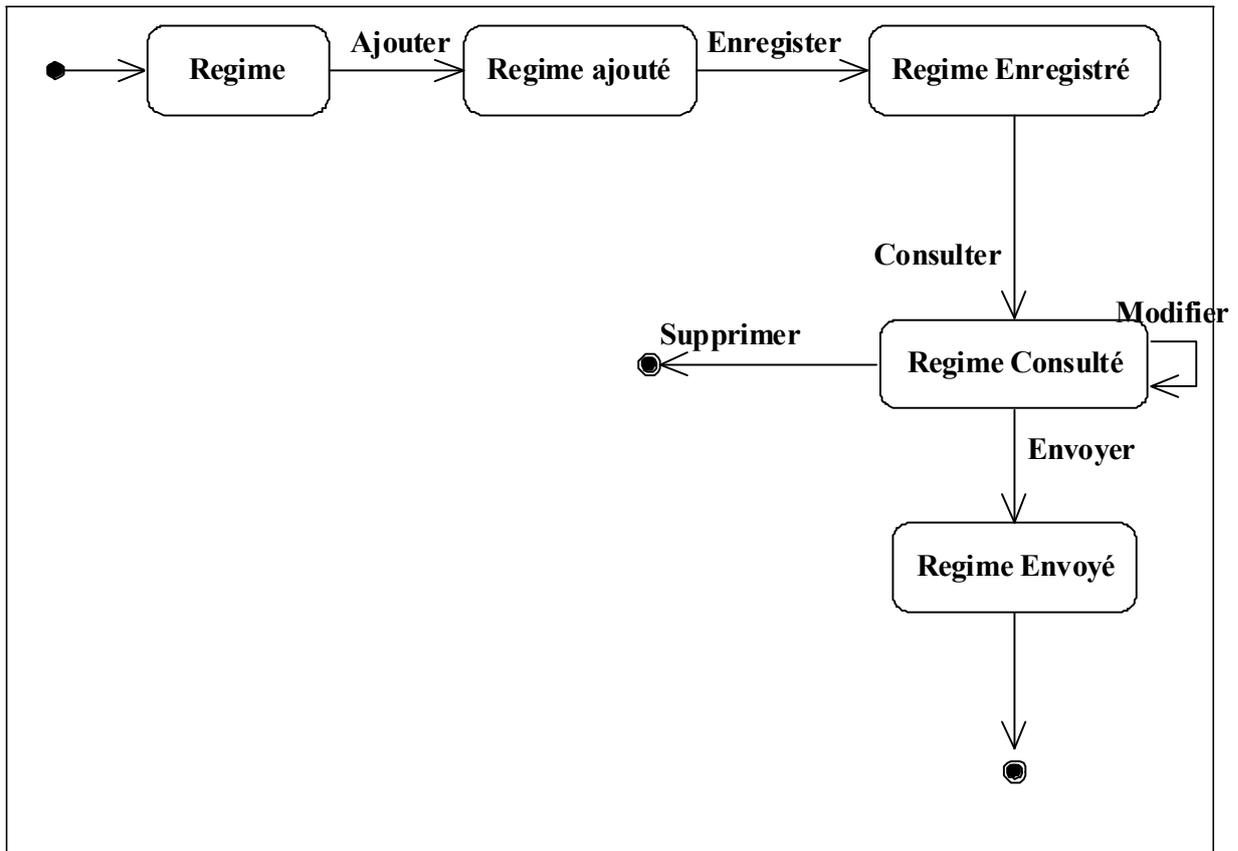


Figure 9. Diagramme état de transition

Conclusion

Au cours de ce chapitre, nous avons présenté l'étape d'analyse qui nous a permis de passer d'une structuration fonctionnelle via les cas d'utilisations et les packages à une structuration objet via les classes et les catégories, Dans le prochain chapitre nous allons suivre la conception préliminaire.

Conception préliminaire

Introduction

La conception préliminaire est certainement l'étape la plus délicate du processus 2TUP car elle en représente le cœur. C'est en effet à cette occasion que s'effectue la fusion des études fonctionnelles et techniques. En conséquence, cette étape permet de passer de l'analyse objet à la conception, c'est-à-dire adapter la conception aux spécifications fournies par l'analyse et intégrer les fonctions métier et applicatives du système dans l'architecture technique.

La première étape de ce chapitre est la conception du modèle de déploiement du système projeté. A partir du déploiement, nous pouvons définir les composants qui seront administrés par l'exploitant du système.

8.1. Développement du modèle du déploiement

Dans cette étape on présente l'architecture physique supportant l'exploitation du système. Cette architecture comprend des nœuds correspondant aux supports physiques (serveurs, routeurs) ainsi que la répartition des artefacts logiciels (bibliothèques, exécutables...) sur ces nœuds, le déploiement d'une solution orientée service se construit sur la définition des postes de travail.

➤ **Architecture adoptée et la technologie utilisée**

Le choix de notre solution s'est porté sur une La notion de SOA (Architecture Orientée Services) définit un modèle d'interactions au niveau logiciel mettant en œuvre des connexions entre des composants logiciels «fournisseurs» à l'attention de composants logiciels «consommateurs». Cette approche a largement été adoptée pour le développement de systèmes de traitement de l'information favorisant par là même, la distribution de fonctionnalités indépendantes, leur réutilisabilité et leur intégration.

Pour la technologie utilisée est les technologies des services Web sont émergentes permettant une interopérabilité entre les différents acteurs (fournisseurs et demandeurs de services) du fait de leur architecture reposant sur des technologies standard.

➤ Notre Diagramme de déploiement

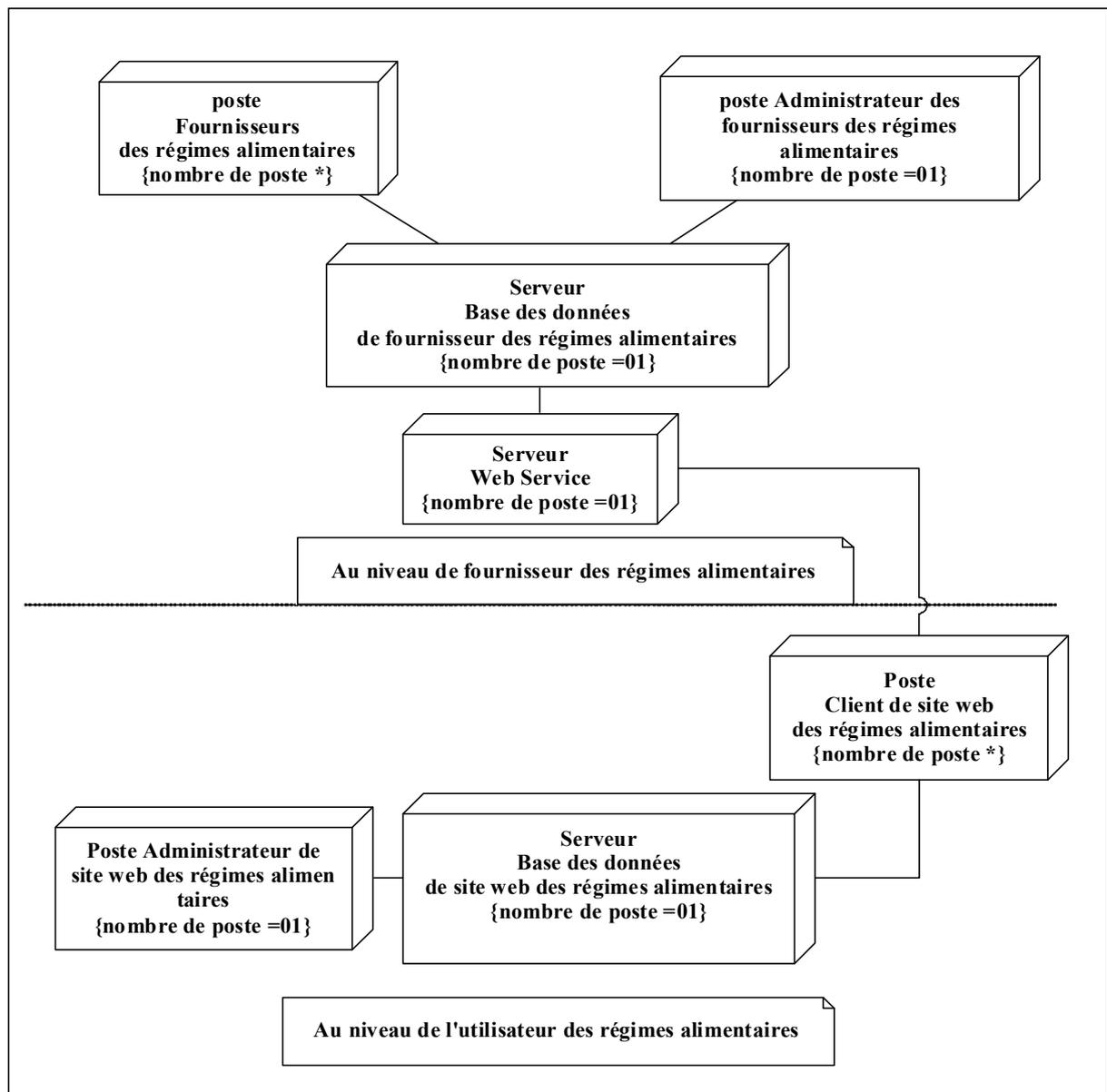


Figure 1. Diagramme de déploiement.

8.1.2. Déploiement du modèle d'exploitation

Le modèle d'exploitation va définir les applications installées sur les postes de travail, les composants métier déployés sur les serveurs et les instances de base de données implantées sur les serveurs également.

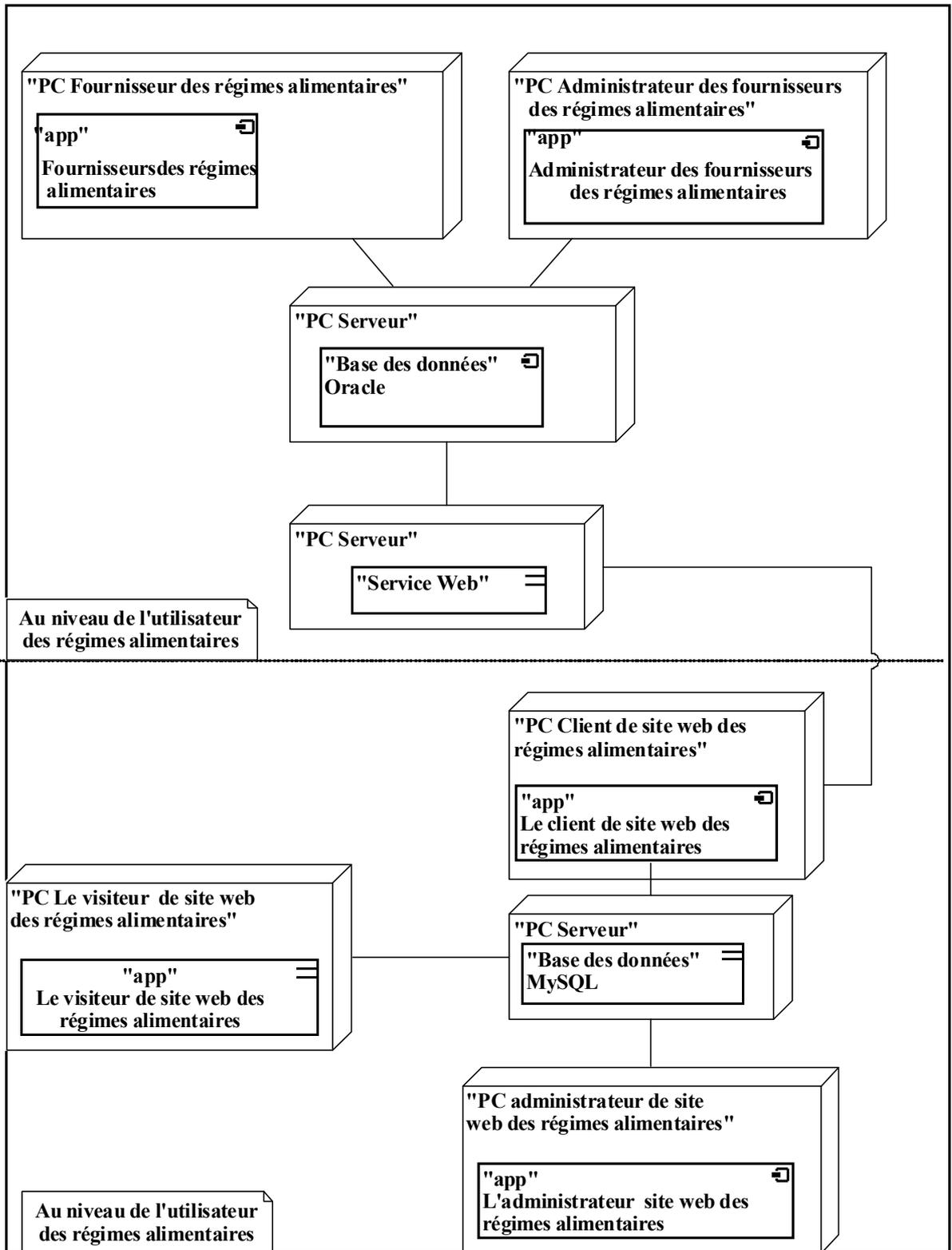


Figure 2. Définition des applications dans modèle statique.

*Conception détaillée***8.2. Dictionnaire des données****8.2.1. Les classes et les attributs**

	Description	Code	Type
Au niveau de fournisseur des régimes alimentaires			
IdentRegime	Numéro de régime	IdR	Entière
	Nom de régime	NomR	Chaîne de caractère
	L'intervalle d'âge d'un client	Age	Chaîne de caractère
	Sexe d'un client	Sexe	Chaîne de caractère
	Effort physique du client	Effort	Chaîne de caractère
	Date de création de régime	dateR	Date
Fournisseur	Numéro de fournisseur	IdF	Entière
	Nom de fournisseur	NomF	Chaîne de caractère
	Prénom de fournisseur	PrenomF	Chaîne de caractère
	Spécialité	Specialite	Chaîne de caractère
	Mot de passe	MotDePasseF	Chaîne de caractère
	Login	LoginF	Chaîne de caractère

Aliment	Numéro de l'aliment	IdAL	Entière
	Nom de l'aliment	NomAL	Chaîne de caractère
	Vitamine	Vitamine	Chaîne de caractère
	Sel	Sel	Chaîne de caractère
	Quantité des protéines	QPro	Chaîne de caractère
	Quantité des lipides	QLip	Chaîne de caractère
	Quantité des glucides	QGlu	Chaîne de caractère
	Quantité d'eau	QEau	Chaîne de caractère
AlimentCon	Numéro de l'aliment conseillé	IdCon	Entière
	Élément effectif	ElementEffectifCon	Chaîne de caractère
AlimentNCon	Numéro de l'aliment non conseillé	IdNCon	Entière
	Élément effectif	ElementEffectifNCon	Chaîne de caractère
Repas	Numéro repas	IdRep	Entière
	Recette	Recette	Chaîne de caractère
Conseille	Numéro de conseille	IdC	Entière
	Conseille	Conseille	Chaîne de caractère
ResumeEtat	Numéro du résumé de l'état de santé	IdRes	Entière

	Le résumé	ResumeTexte	Chaîne de caractère
LimitRegime	Numéro de limite	IdLim	Entière
	Vitamin	Vitamine	Chaîne de caractère
	Quantité des sels	QSel	Flaot
	Quantité des protéines	QPro	Flaot
	Quantité des lipides	QLip	Flaot
	Quantité des glucides	QGlu	Flaot
AdminF	Nom de l'administrateur des fournisseurs	NomAF	Chaîne de caractère
	Login	LoginAF	Chaîne de caractère
	Mot de Passe	MotDePasse	Chaîne de caractère
	Numéro de l'administrateur	IdAF	Entière
Au niveau de site web des régimes alimentaires			
Client	Numéro de client	Id	Entière
	Nom de client	Nom	Chaîne de caractère
	Prénom de client	Prenom	Chaîne de caractère
	Date de naissance	date_naissance	Chaîne de caractère
	Email	Email	Chaîne de caractère

	Passer word	Passerword	Chaîne de caractère
	Pseudo	Pseudo	Chaîne de caractère
Fichier	Numéro de fichier	Code_Fichier	Entière
	Nom de fichier	Nom_Fichier	Chaîne de caractère
	Fichier	Fichier	Chaîne de caractère
Annonce	Numéro de l'annonce	id_an	Entière
	Nom de l'annonce	nom_an	Chaîne de caractère
	Date de création de l'annonce	Date	Date
	Contenu	Contenu	Chaîne de caractère

Tableau 1. Dictionnaire de donnée avec les classes et les attributs.

8.2.1. Les opérations

Classe	Opération	Description
Au niveau de fournisseur des régimes alimentaires		
IdentRegime	Ajouter	Ajouter une identification de régime
	Modifier	Modifier les informations dans l'identification de régime
	Supprimer	Supprimer l'identification de régime (par défaut le régime)
	Consulter	Consulter l'identification de régime
Fournisseur	Ajouter	Ajouter un fournisseur des régimes alimentaires

	Modifier	Modifier les informations d'un fournisseur des régimes alimentaires
	Supprimer	Supprimer un fournisseur
	Consulter	Consulter la table des fournisseurs
Aliment	Ajouter	Ajouter un aliment
	Modifier	Modifier les informations d'un aliment
	Supprimer	Supprimer un aliment
	Consulter	Consulter la table des aliments
AlimentCon	Ajouter	Ajouter un aliment conseillé
	Modifier	Modifier les informations d'un aliment conseillé
	Supprimer	Supprimer un aliment conseillé
	Consulter	Consulter la table des aliments conseillés
AlimentNCon	Ajouter	Ajouter un aliment non conseillé
	Modifier	Modifier les informations d'un aliment non conseillé
	Supprimer	Supprimer un aliment non conseillé
	Consulter	Consulter la table des aliments non conseillés
Repas	Ajouter	Ajouter un repas
	Modifier	Modifier les informations d'un repas
	Supprimer	Supprimer un aliment un repas
	Consulter	Consulter la table des repas

Conseille	Ajouter	Ajouter une conseille
	Modifier	Modifier les informations d'une conseille
	Supprimer	Supprimer une conseille
	Consulter	Consulter la conseille
ResumeEtat	Ajouter	Ajouter une résumé de l'état de santé
	Modifier	Modifier les informations d'une résumé de l'état de santé
	Supprimer	Supprimer une résumé de l'état de santé
	Consulter	Consulter la résumé de l'état de santé
LimitRegime	Ajouter	Ajouter une limite pour un régime alimentaire
	Modifier	Modifier les informations d'une limite d'un régime alimentaire
	Supprimer	Supprimer une limite d'un régime alimentaire
	Consulter	Consulter la limite
Au niveau de site web des régimes alimentaires		
Client	Inscrire	Le visiteur de site web des régimes alimentaires
	Modifier	Modifier dans les informations du client de site web des régimes alimentaire
	Supprimer	Supprimer un client de site web des régimes alimentaire
	Consulter	Consulter la table des clients de site web des régimes alimentaire

Fichier	Ajouter	Ajouter un fichier
	Modifier	Modifier dans les informations d'un fichier
	Supprimer	Supprimer un fichier
	Consulter	Consulter la table des fichiers
Annonce	Ajouter	Ajouter une annonce

Tableau 2. Dictionnaire de donnée avec les opérations.

8.3. Le modèle relationnel

A partir de la description conceptuelle que nous avons effectuée, on peut réaliser le modèle relationnel ; vu que le système d'information ne peut pas le manipuler directement ; et ça en utilisons des règles de passage de l'UML vers le relationnel.

Quelque notion essentielle :

- **Domaine** : c'est l'ensemble de valeurs d'un attribut.
- **Relation** : c'est un sous ensemble du produit cartésien d'une liste de domaines.

C'est en fait un tableau à deux dimensions dont les colonnes correspondent aux domaines et dont les lignes contiennent des tuples. On associe un nom à chaque colonne.

- **Attribut** : c'est une colonne d'une relation, caractérisé par un nom.
- **Tuples** : c'est la liste des valeurs d'une ligne d'une relation.
- **Cardinalité** : elle permet de définir les conditions de participation d'une entité à une relation. Toutefois, une entité peut participer à plusieurs relations
- **L'arité** : est le nombre d'attributs d'une relation.
- **Clé** : on distingue deux types de clés :
- **Clé primaire** : ensemble d'attributs dont les valeurs permettent de distinguer les nuplets les uns des autres (notion d'identifiant).
- **Clé étrangère** : attribut qui est clé primaire d'une autre entité.

8.3.1. Les règle de passage

La réalisation du modèle relationnel peut être faite à partir de la description conceptuelle. Ce passage de modèle doit respecter un certain nombre de règle.

- **Règle1** : Toute entité devient une relation ayant pour clé primaire son identifiant.

Chaque propriété se transforme en attribut.

- **Règle2** : Toute association hiérarchique (de type [1, n]) se traduit par une clé étrangère
- **Règle3** : Toute association non hiérarchique (de type [n, n] ou de dimension > 2) devient une relation
- **Règle4** : Les entités n'ayant que leur identifiant comme attribut ne deviennent pas des relations, mais des attributs dans les autres relations liées.
- **Règle5** : cas de héritage, transformer chaque sous classe en une relation, la clé primaire de la super classe devient clé primaire de chaque sous classe.
- **Règle6** : cas de composition, la clé primaire de la classe composée devient clé étrangère de la classe composant.
- **Règle7** : cas d'agrégation, le même principe que la Règle 2.

8.3.2. Les règles de gestion

- **Au niveau de fournisseur des régimes alimentaires**
 - Un fournisseur peut créer plusieurs régimes, plusieurs aliments, plusieurs religions, et plusieurs régions.
 - Un régime est créé par un seul fournisseur.
 - Un régime est composé d'une résume de l'état de santé, un conseille, une seule limite.
 - Un régime contient plusieurs aliments conseillés, et plusieurs aliments non conseillés.
 - Un aliment conseillé (respectivement non conseillé) peut apparaitre dans plusieurs régimes.
 - Un aliment conseillé (respectivement non conseillé) hérite les propriétés de l'aliment.
 - Un repas contient plusieurs aliments conseillés.

- Un aliment conseillé peut apparaitre dans plusieurs repas.
- **Au niveau de site web des régimes alimentaires**
 - Un client peut invoquer plusieurs régimes.
 - Un régime peut invoquer par plusieurs clients.
 - Un client peut consulter plusieurs fichiers.
 - Un fichier peut consulter par plusieurs clients.
 - Un client peut consulter plusieurs annonces.
 - Une annonce peut consulter par plusieurs clients.

8.3.3. Les tables de la base de données

En se basant sur les règles ci-dessus, nous avons converti les classes entités et leurs associations, à des tables dans la base données. Les tables générées sont :

➤ **Au niveau de fournisseur des régimes alimentaires**

Fournisseur (IdF, NomF, PrenomF, Specialite , MotDePasseF, LoginF)

IdentRégime (IdR, NomR, Age, Sexe, Effort, dateR, #IdF, #NomF)

Aliment(IdAL, NomAL, Sel, QPro, QLip, QGlu, QEau, #IdF, #NomF)

AlimentNCon (IdNCon, #IdR, #NomR)

AlimentCon (IdNCon, #IdR, #NomR)

Repas (IdRep, Recette, #IdNCon)

Region (IdReg, NomRegion, #IdF, #NomF)

Religion (IdRL, NomReligion, #IdF, #NomF)

LimitRegime (IdLim, Vitamin, QSel, QPro, QLip, QGlu, #IdR, #NomR)

ResumeEtat (IdRes, ResumeTexte, #IdR, #NomR)

Conseille (IdC, Conseille, #IdR, #NomR)

ElementEffectifNCon (ElementEffectifNCon, # IdNCon, #IdR, #NomR)

ElementEffectifCon (ElementEffectifCon, # IdCon, #IdR, #NomR)

➤ **Au niveau de l'utilisateur des régimes alimentaires**

Client (id, nom, prenom, date_naissance, email, password)

Fichier (Code_Fichier, Nom_Fichier, Fichier)

Annonce (id_an, nom_an, date)

Client_Fichier (id, Code_Ficher)

Client_Annonce (id_an, id)

➤ **Au niveau de l'interaction entre les deux systèmes**

Client_IdentRégime (#IdR, #NomR,# id)

Introduction

On commence la phase d'implémentation. Cette dernière phase a pour but de générer un code exécutable répond aux besoins utilisateur.

9.1. Présentation des outils de développement de l'application

❖ Au niveau de fournisseurs des régimes alimentaires

9.1.1. Le langage de programmation java

Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Ses caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d'être très largement utilisé

Pour le développement d'applications de types très disparates. Java est notamment largement utilisée pour le développement d'applications d'entreprises.

- JAVA est Java est interprété, Java est portable : il est indépendant

De toute plate-forme, Java est orienté objet, Java est simple, Java est fortement typé, Java assure la gestion de la mémoire, Java est sûr, Java est économe, Java est multitâche.

- Les services web font leur apparition dans la version SE de Java, ils n'étaient intégrés que dans la version EE.
- Java Standard Edition (J2SE / Java SE) : environnement d'exécution et ensemble complet d'api pour des applications de type desktop. Cette plate-forme sert de base en tout ou partie aux autres plate-formes
- Java Enterprise Edition (J2EE / Java EE) : environnement d'exécution reposant intégralement sur Java SE pour le développement d'applications d'entreprises
- API pour les services web est JAX_WS

9.1.2. Eclipse

➤ Eclipse est un environnement de développement intégré (Integrated Development Environment) dont le but est de fournir une plate-forme modulaire pour permettre de réaliser des développements informatiques.

➤ Eclipse utilise énormément le concept de modules nommés "plug-ins" dans son architecture. D'ailleurs, hormis le noyau de la plate-forme nommé "Runtime", tout le reste de la plate-forme est développé sous la forme de plug-ins. Ce concept permet de fournir un mécanisme pour l'extension de la plate-forme et ainsi fournir la possibilité à des tiers de développer des fonctionnalités qui ne sont pas fournies en standard par Eclipse.

➤ Nous utilisons eclipse car :

- ✓ Une plate-forme ouverte pour le développement d'applications et extensible grâce à un mécanisme de plug-ins
- ✓ Possibilité d'utiliser des outils open source
- ✓ Malgré son écriture en Java, Eclipse est très rapide à l'exécution
- ✓ En plus de Java, il permet également de supporter différents autres langages,

comme Python, C, C++, XML et HTML

9.1.3. Implémentation de la base de données

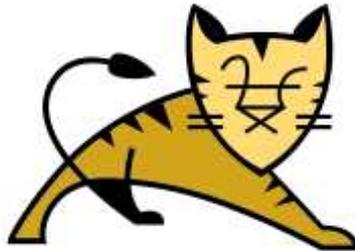
Pour l'implémentation de la base de données que notre système utilise, nous avons sélectionné le SGBD Oracle, qu'est un système de gestion de base de données relationnel reconnu mondialement.

Ce système complexe permet la gestion d'une manière efficace, de volumes importants de données structurées, accessible par des utilisateurs simultanés locaux ou non. Il assure :

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

9.1.4. Le serveur

Apache Tomcat est une implémentation open source d'un conteneur web qui permet donc d'exécuter des applications web reposant sur les technologies servlets et JSP.



❖ Au niveau de l'utilisateur des régimes alimentaires

9.2.1. Le PHP

Qui signifie « personnel home page » est apparu en 1994, sous forme de sous forme de petits outils pour faciliter la vie de programmeur web. PHP est un langage de scripts, il est interprété, par conséquent il ne nécessite pas d'être compilé pour obtenir un objet, un exécutable avant d'être utilisable (comme C par exemple). PHP est un module supporté par le serveur web Apache, le plus répandu dans le monde, il est donc développé pour être facilement de très nombreuses bases de données notamment MySQL gratuit et performante.

- La principale utilité de PHP est l'interfaçage de bases de données sur Internet.
- PHP est également utile pour programmer le site en modules que l'on assemble ensuite via une page. La mise à jour de parties revenant sur un site ainsi facilitée.

Les avantages de PHP :

- Rapidité de développement : sa syntaxe simple (héritée de C et de Perl), ses fonctions particulièrement adaptées aux applications web et son excellente documentation permettent une grande productivité.
- Richesse : PHP dispose d'un grand nombre d'extensions qui lui permettent de couvrir l'essentiel, si non la totalité, des besoins relatifs aux applications web (base de données, mail, expressions régulières, PDF, XML).

- Des applications librement réutilisables : elles sont très nombreuses, et sont aussi diverses s de gestion de base de données, des interfaces mail, des systèmes de publication,.....etc.
- La documentation : le manuel annoté est l'excellente qualité, et d'innombrables documents, sont disponibles sur internet.
- Portabilité et interfaçage avec les principaux serveurs http.

9.2.2. My sql

Le recours à my sql permet de masquer les détails complexes et fastidieux liés à l'utilisation des fichiers, my sql gère les fichiers constituant une base de données, prend en charge les fonctionnalités de protection et de sécurité, et fournit un ensemble d'interfaces de programmation facilitant l'accès aux données.

9.2.3. L'interface my sql/php

Php communique avec my sql par l'intermédiaire d'un ensemble de fonctions qui permettent de récupérer, modifier, ou créer un peu près toutes les informations relatives à une base de données. Parmi ces informations, il faut compter bien étendu le contenu des tables, mais également leur description(le schéma de base).l'utilitaire php my admin utilise par exemple les fonctions permettant d'obtenir le schéma pour présenter une interface d'administration, générer à la volée des formulaires de saisie.

9.2.4. L'interface php my admin

Php my admin est un outil entièrement écrit en php qui fournit une interface simple et très complète pour administrer une base my sql. La plus part des commandes de l'utilitaire my admin, les opérations possibles dépendant bien sur des droits de l'utilisateur qui se connecte à la base voici une liste des principales possibilités :

- Créer, détruire des bases de données (sous le compte root de my sql)
- Créer, détruire, modifie la description des tables.
- Consulter le contenu des tables, modifier certaines lignes ou les détruire, etc.
- Charger des fichiers dans des tables et réciproquement, récupère ASCII.

a. Dream weaver :

Logiciel crée par macro media (et géré maintenant par adobe) Permettant la conception de site web. Dreamweaver fonctionne en mode WYSIWYG ou en mode de code et dispose d'un client FTP permettant le transfert des pages créés sur le serveur hébergeant le site web.

9.3. Quelques interfaces de l'application

Dans cette section nous présentons quelque fenêtre de notre application développée :

➤ Page d'accueil de site web

Figure 1. Page d'accueil de site web

➤ Page d'accueil de l'application des fournisseurs des régimes alimentaires



Figure 2. Page d'accueil de l'application des fournisseurs.

➤ page d'authentification



Figur1. Page d'authentification.

Cette page permet au fournisseur (ou admin des fournisseur) d'accéder a sa page de fonctionnalité de l'agent après avoir être identifié avec login et son mot de passe

➤ Page des opérations de fournisseur



Figure 4. Page des opérations de fournisseur.

➤ Page ajouter aliment



Figure 5. Page ajouter aliment

Conclusion

Nous avons vu à travers ce chapitre l'implémentation de notre application. Nous avons tout d'abord cité les différents outils utilisés lors du développement de notre système, nous avons ensuite détaillé les fonctionnalités de notre prototype de services web dans le cadre de notre application.

Conclusion Générale

Dans ce projet de fin d'études, nous avons appliqué la technologie des web services dans le domaine de la nutrition, l'alimentation et la santé. Ces services sont dédiés aux régimes alimentaires et respectent plusieurs critères qui nous en font nécessiter une recherche bibliographique profonde pour essayer de les comprendre et de déterminer une classification logique des régimes alimentaires.

Bien que le domaine des régimes alimentaires soit vaste et dynamique, nous disons que nous avons atteint les buts que nous avons fixés au début, du fait que nous avons pu répondre à la demande de l'utilisateur avec n'importe quel critère qu'il possède et même s'il souffre de plusieurs maladies. Il ne reste que le cas d'utilisation « Rechercher » qui nécessite soit la publication des services web dans un UDDI publique et l'hébergement du site, ou bien la création (la programmation) d'un UDDI et ainsi travailler dans le local host. La première solution n'est pas assez facile à réaliser, tandis que la deuxième nécessite plus de temps pour sa réalisation.

Ce projet s'est avéré bénéfique sur plusieurs plans. Il nous a permis d'acquérir de nouvelles connaissances et de renforcer d'autres. Nous citons particulièrement les services web, UML, 2TUP, Java, Oracle, MySql, Eclipse et Php... Et il nous a permis aussi de voir et de réaliser l'invocation et l'envoi des services web entre deux systèmes hétérogènes.

Toutefois, des perspectives pour améliorer notre application restent envisageables telles que concevoir un système expert d'aide à la décision des maladies liées à l'alimentation via une approche basée sur les systèmes multi agents. Un tel système permettrait de donner des conseils qualitatifs et quantitatifs à propos des aliments et des repas avec un suivi des patients. La proposition de l'application en version mobile serait un important atout afin de mieux exploiter les services que nous offrons.

Bibliographie

Chapitre 1

[1]: Thierry MATHÉ- Thomas PILORIN- Pascale HÉBEL, DU DISCOURS NUTRITIONNEL AUX REPRÉSENTATIONS DE L'ALIMENTATION, DÉCEMBRE 2008.

[2] : service-educatif-la-hunaudaye.com

[3] : http://sante-medecine.commentcamarche.net/faq/20871-nutriments-definition#simili_main

[4] : nutrition et alimentation, Sup de Cours - Etablissement d'enseignement privé RNE 0333 119 L - 73, rue de Marseille - 33000 Bordeaux

[5] : http://www.cap-sciences.net/upload/equilibre_alim.pdf

[6] : Dr. Me Mohammed RACHIDI, Les apports des biotechnologies et la sécurité alimentaire, 2014

[7] : <http://www.retourvital.com/Equilibre-Alimentaire.php>

[8] : Thierry Mathe -Aurée Francou- Justine Colin -Pascale Hebel, comparaison des modèles alimentaires français et Etats-Uniens, décembre 2011.

[9] : <http://www.i-dietetique.com/articles/maigrir-une-histoire-de-regimes/8813.html>

[10] : <http://regime.comprendrechoisir.com/comprendre/regime-definition>

[11] : article_rga_0035-1121_1957_num_45_2_1898

[12] : Alimentation et environnement

[13] : <http://blog.biomouv.com/2011/02/25/fausses-promesses-et-vrais-dangers-des-regimes-alimentaires/>

Chapitre 2

[1] : Julien DUCRET, Digital Museum, Promotion 2002-2003.

[2] : Ahmed LAFTIMI, SOA (Service Oriented Architecture) Architectures Orientées Services, CNAM 2008-2009.

[3] : CHOUIREF Zahira, Un système de programmation fonctionnelle pour la composition de services Web, Année Universitaire : 2007/2008.

[4]: Issam RABHI, Testabilité des services Web, Soutenue publiquement le 09 janvier 2012.

[5]: Mohamed GHARZOULI, Composition des Web Services Sémantiques dans les systèmes Peer-to-Peer, Soutenue le : 25/09/2011 .

[8]: SoftDeath, Les services Web.

[9] : DALI YAHIA Mohammed, Sélection de service web à base QoWS, 2010/2011.

[10] : http://fr.wikipedia.org/wiki/World_Wide_Web

[11]: Francisco José MOO MENA, Application à la gestion de la qualité de service des applications à base de services Web, Soutenue le 4 Avril 2007.

[12]: MAHFOUD Sami, Interopérabilité sémantique des données échangées entre les services Web, engagés dans une composition (Vers une approche de médiation de données basée sur le contexte), 2010/2011.

[13]: Djelmodi Abdelhamid Belaredj Mohamed Amin, La sélection des services web à base des systèmes immunitaires artificiels (CLONAG), 2012/2013.

[14]: <http://www.tetraedre.com/advanced/xml.php>

[15]: CHAIB AOuatef, Composition automatique et semi- automatique des services Web sémantique par l'utilisation des techniques de planification distribuées «Les Systèmes Multi Agents », 18/06/2011.

[16] : Melle CHEMIDI Zoulikha, Sélection de services Web à base de colonies de fourmis, 2012.

Chapitre 3

[1] DI GALLO Frédéric , Méthodologie des systèmes d'information – UML, 2000-2001.

Annexe

Microsoft : est une entreprise d'informatique et de micro-informatique multinationale américaine, fondée par Bill Gates et Paul Allen

IBM : International Business Machines Corporation, connue sous l'abréviation IBM, est une société multinationale américaine présente dans les domaines du matériel informatique, du logiciel et des services informatiques.

XML-RPC : est un protocole RPC (Remote procedure call), une spécification simple et un ensemble de codes qui permettent à des processus s'exécutant dans des environnements différents de faire des appels de méthodes à travers un réseau.

W3C : Le World Wide Web Consortium, abrégé par le sigle W3C, est un organisme de normalisation à but non lucratif, fondé en octobre 1994 chargé de promouvoir la compatibilité des technologies du World Wide Web telles que HTML5, HTML, XHTML, XML, RDF, SPARQL, CSS, XSL, PNG, SVG et SOP.

Ariba : est une société de logiciels et services de technologie de l'information située à Sunnyvale, en Californie.

API En informatique, une interface de programmation (souvent désignée par le terme API pour *Application Programming Interface*) est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels. Elle est offerte par une bibliothèque logicielle ou un service web, le plus souvent accompagnée d'une description qui spécifie comment des programmes *consommateurs* peuvent se servir des fonctionnalités du programme *fournisseur*.

CORBA : Le Common Object Request Broker Architecture (CORBA) est une norme définie par l' Object Management Group (OMG) conçu pour faciliter la communication des systèmes qui sont déployés sur diverses plates-formes. CORBA permet la collaboration entre les systèmes sur différents systèmes d'exploitation, langages de programmation, et du matériel informatique.

DCOM : Distributed Component Object Model, une technique propriétaire de Microsoft qui permet la communication entre des composants logiciels distribués au sein d'un réseau informatique.

http : L'*HyperText Transfer Protocol*, plus connu sous l'abréviation HTTP — littéralement « protocole de transfert hypertexte » — est un protocole de communication client-serveur développé pour le World Wide Web.