

PROJET DE FIN D'ÉTUDES

**POUR L'OBTENTION DE LA LICENCE FONDAMENTALE EN
INFORMATIQUE DE GESTION**

Conception et développement d'un comparateur de tarifications bancaires tunisiennes

Elaborés par:

Montassar BEN DHIFALLAH
Mohamed Aziz CHAKER

Entreprise d'accueil



Banque Internationale
Arabe de Tunisie

Encadrement

Responsable ISG : Yamen MKADEM
Responsable entreprise : Ferjani RIAHI

Année universitaire 2016/2017

Remerciements

Comme le veut la tradition, nous allons tenter de nous soumettre au dur exercice des remerciements.

En premier lieu, nous tenons à exprimer nos plus vifs remerciements à M. Ferjani Riahi, Directeur Développement SI de nous avoir accueilli au sein de la BIAT et de nous permettre de mener à bien la réalisation de notre projet de fin d'études.

Nos plus vifs remerciements s'adressent à toute l'équipe du Département des Systèmes d'Information pour les moments sympathiques vécus.

Au cours de ce stage notre profonde gratitude s'adresse à notre encadrant M.Ferjani Riahi pour sa disponibilité et pour tout le savoir qu'il nous a inculqué durant notre projet, l'assistance qu'il n'a cessé de nous apporter et son bénéfique encadrement qui nous ont permis de mener à bien le présent travail. Veuillez trouver dans ce travail l'expression de notre profonde reconnaissance pour votre disponibilité et notre sincère admiration pour votre compétence.

Nous sommes particulièrement redevables à notre encadrant universitaire M. Yamen Mkaadem pour l'honneur que vous nous faites en acceptant de diriger ce travail.

Enfin nos meilleurs remerciements s'adressent à tous nos professeurs pour les connaissances qu'ils nous ont acquis durant nos années d'études à ISG SOUSSE.

Contents

Introduction générale	1
1 Cadre général du projet	2
1.1 Présentation de l'organisme d'accueil	2
1.2 Description du contexte du projet	3
1.2.1 Etude de l'Existant	4
1.2.2 Objectif du projet	4
1.2.3 Méthode et langage de modélisation utilisés	4
2 La phase de création	6
2.1 Capture des besoins	6
2.1.1 Identification des besoins fonctionnels	6
2.1.2 Identification des besoins non fonctionnels	7
2.1.3 Identification des acteurs et des cas d'utilisation	8
2.1.4 Affectation des priorités aux cas d'utilisation	10
2.1.5 Diagramme de cas d'utilisation initial	11
2.1.6 Raffinement des cas d'utilisation de priorité 1	12
2.2 Conception	21
2.2.1 La conception des cas d'utilisation	21
2.3 Implémentation	33
2.3.1 Implémentation du cas d'utilisation «S_authentifier»	33
2.3.2 Implémentation du cas d'utilisation «GererBanque»	35
2.3.3 Implémentation du cas d'utilisation «GererCategorieTarif»	36
2.3.4 Implémentation du cas d'utilisation «GererTarif»	37
2.3.5 Implémentation du cas d'utilisation «GererTarification»	38
2.3.6 Implémentation du cas d'utilisation «ComparerTarification»	39
3 La phase d'élaboration	40
3.1 Capture des besoins	40
3.1.1 Raffinement des cas d'utilisation de priorité 2	40
3.2 Conception	53
3.2.1 Conception des cas d'utilisation de priorité 2	53
3.3 Implémentation	66
3.3.1 Implémentation du cas d'utilisation «GererAgence»	66
3.3.2 Implémentation du cas d'utilisation «GererDAB»	67
3.3.3 Implémentation du cas d'utilisation «GererGouvernorat»	68
3.3.4 Implémentation du cas d'utilisation «GererDelegation»	69
3.3.5 Implémentation du cas d'utilisation «GererVille»	70
3.3.6 Implémentation du cas d'utilisation «GererUtilisateur»	71
3.3.7 Implémentation du cas d'utilisation «GererAdherent»	72

4 La phase de construction	73
4.1 Capture des besoins	73
4.1.1 Raffinement des cas d'utilisation de priorité 3	73
4.2 Conception	74
4.2.1 Conception des cas d'utilisation de priorité 3	74
4.2.2 Conception des classes entités	78
4.2.3 Structure de la base de données	79
4.3 Implémentation	85
4.3.1 Implémentation du cas d'utilisation «ConsulterTrafic»	85
4.3.2 Implémentation du cas d'utilisation «ConsulterBanque»	86
4.3.3 Implémentation du cas d'utilisation «ConsulterAgence»	87
4.3.4 Implémentation du cas d'utilisation «ConsulterDAB»	88
4.4 Diagramme de déploiement	89
5 La phase de transition	90
5.1 Architecture de l'application	90
5.2 Environnement de développement	90
5.2.1 Les outils matériels	91
5.2.2 Les outils logiciels	91
5.3 Les interfaces graphiques de l'application	92
5.3.1 S'authentifier admin	92
5.3.2 Réinitialiser mot de passe	95
5.3.3 Ajout Banque	101
5.3.4 Ajout Tarif	104
5.3.5 Modifier tarification	108
5.3.6 Comparaison des tarifications	111
Conclusion générale	116
Bibliographie	117
Annexe	117

List of Figures

1.1	Logo de la société BIAT	2
1.2	Les filiales du groupe bancaire BIAT	3
2.1	Diagramme du cas d'utilisation initial	11
2.2	Diagramme du cas d'utilisation initial partie 2	12
2.3	Raffinement du cas d'utilisation «GererBanque»	13
2.4	Raffinement du cas d'utilisation «GererCategorieTarif»	15
2.5	Raffinement du cas d'utilisation «GererTarif»	17
2.6	Raffinement du cas d'utilisation «GererTarification»	19
2.7	Traçabilité MCU/MC du cas d'utilisation «S_authentifier»	21
2.8	Traçabilité MCU/MC du cas d'utilisation «Sauthentifier» pour l'adherent	22
2.9	Diagramme de séquence du cas d'utilisation «S_authentifier»	22
2.10	Diagramme de séquence du cas d'utilisation «Sauthentifier» pour l'adherent	23
2.11	Traçabilité MCU/MC du cas d'utilisation «GererBanque»	23
2.12	Diagramme de séquence du cas d'utilisation «AjouterBanque»	24
2.13	Diagramme de séquence du cas d'utilisation «ModifierBanque»	25
2.14	Diagramme de séquence du cas d'utilisation «SupprimerBanque»	25
2.15	Traçabilité MCU/MC du cas d'utilisation «GererCategorieTarif»	26
2.16	Diagramme de séquence du cas d'utilisation «AjouterCategorieTarif»	26
2.17	Diagramme de séquence du cas d'utilisation «ModifierCategorieTarif»	27
2.18	Diagramme de séquence du cas d'utilisation «SupprimerCategorieTarif»	27
2.19	Traçabilité MCU/MC du cas d'utilisation «GererTarif»	28
2.20	Diagramme de séquence du cas d'utilisation «AjouterTarif»	28
2.21	Diagramme de séquence du cas d'utilisation «ModifierTarif»	29
2.22	Diagramme de séquence du cas d'utilisation «SupprimerTarif»	29
2.23	Diagramme de séquence du cas d'utilisation «GererTarification»	30
2.24	Diagramme de séquence du cas d'utilisation «AjouterTarification»	30
2.25	Diagramme de séquence du cas d'utilisation «ModifierTarification»	31
2.26	Diagramme de séquence du cas d'utilisation «SupprimerTarification»	31
2.27	Traçabilité MCU/MC du cas d'utilisation «ComparerTarification»	32
2.28	Diagramme de séquence du cas d'utilisation «ComparerTarification»	32
2.29	Traçabilité MC/MI du cas d'utilisation «Sauthentifier»	33
2.30	Traçabilité MC/MI du cas d'utilisation «Sauthentifier» pour l'adhérent	33
2.31	Diagramme de composant du cas d'utilisation «Sauthentifier»	34
2.32	Diagramme de composant du cas d'utilisation «Sauthentifier» pour l'adhérent	34
2.33	Traçabilité MC/MI du cas d'utilisation «GererBanque»	35
2.34	Diagramme de composant du cas d'utilisation «GererBanque»	35
2.35	Traçabilité MC/MI du cas d'utilisation «GererCategorieTarif»	36
2.36	Diagramme de composant du cas d'utilisation «GererCategorieTarif»	36
2.37	Traçabilité MC/MI du cas d'utilisation «GererTarif»	37

2.38	Diagramme de composant du cas d'utilisation «GererTarif»	37
2.39	Traçabilité MC/MI du cas d'utilisation «GererTarification»	38
2.40	Diagramme de composant du cas d'utilisation «GererTarification»	38
2.41	Traçabilité MC/MI du cas d'utilisation «ComparerTarification»	39
2.42	Diagramme de composant du cas d'utilisation «ComparerTarification»	39
3.1	Raffinement du cas d'utilisation «GererAgence»	40
3.2	Raffinement du cas d'utilisation «GererDAB»	42
3.3	Raffinement du cas d'utilisation «GererGouvernorat»	44
3.4	Raffinement du cas d'utilisation «GererDelegation»	46
3.5	Raffinement du cas d'utilisation «GererVille»	48
3.6	Raffinement du cas d'utilisation «GererUtilisateur»	50
3.7	Raffinement du cas d'utilisation «GererAdherent»	51
3.8	Traçabilité MCU/MC du cas d'utilisation «GererAgence»	53
3.9	Diagramme de séquence du cas d'utilisation «AjouterAgence»	53
3.10	Diagramme de séquence du cas d'utilisation «ModifierAgence»	54
3.11	Diagramme de séquence du cas d'utilisation «SupprimerAgence»	54
3.12	Traçabilité MCU/MC du cas d'utilisation «GererDAB»	55
3.13	Diagramme de séquence du cas d'utilisation «AjouterDAB»	55
3.14	Diagramme de séquence du cas d'utilisation «ModifierDAB»	56
3.15	Diagramme de séquence du cas d'utilisation «SupprimerDAB»	56
3.16	Traçabilité MCU/MC du cas d'utilisation «GererGouvernorat»	57
3.17	Diagramme de séquence du cas d'utilisation «AjouterGouvernorat»	57
3.18	Diagramme de séquence du cas d'utilisation «ModifierGouvernorat»	58
3.19	Diagramme de séquence du cas d'utilisation «SupprimerGouvernorat»	58
3.20	Traçabilité MCU/MC du cas d'utilisation «GererDelegation»	59
3.21	Diagramme de séquence du cas d'utilisation «AjouterDelegation»	59
3.22	Diagramme de séquence du cas d'utilisation «ModifierDelegation»	60
3.23	Diagramme de séquence du cas d'utilisation «SupprimerDelegation»	60
3.24	Traçabilité MCU/MC du cas d'utilisation «GererVille»	61
3.25	Diagramme de séquence du cas d'utilisation «AjouterVille»	61
3.26	Diagramme de séquence du cas d'utilisation «ModifierVille»	62
3.27	Diagramme de séquence du cas d'utilisation «SupprimerVille»	62
3.28	Traçabilité MCU/MC du cas d'utilisation «GererUtilisateur»	63
3.29	Diagramme de séquence du cas d'utilisation «AjouterUtilisateur»	63
3.30	Diagramme de séquence du cas d'utilisation «SupprimerUtilisateur»	64
3.31	Traçabilité MCU/MC du cas d'utilisation «GererAdherent»	64
3.32	Diagramme de séquence du cas d'utilisation «SupprimerAdherent»	65
3.33	Traçabilité MC/MI du cas d'utilisation «GererAgence»	66
3.34	Diagramme de composant du cas d'utilisation «GererAgence»	66
3.35	Traçabilité MC/MI du cas d'utilisation «GererDAB»	67
3.36	Diagramme de composant du cas d'utilisation «GererDAB»	67
3.37	Traçabilité MC/MI du cas d'utilisation «GererGouvernorat»	68
3.38	Diagramme de composant du cas d'utilisation «GererGouvernorat»	68
3.39	Traçabilité MC/MI du cas d'utilisation «GererDelegation»	69
3.40	Diagramme de composant du cas d'utilisation «GererDelegation»	69

3.41 Traçabilité MC/MI du cas d'utilisation «GererVille»	70
3.42 Diagramme de composant du cas d'utilisation «GererVille»	70
3.43 Traçabilité MC/MI du cas d'utilisation «GererUtilisateur»	71
3.44 Diagramme de composant du cas d'utilisation «GererUtilisateur»	71
3.45 Traçabilité MC/MI du cas d'utilisation «GererAdherent»	72
3.46 Diagramme de composant du cas d'utilisation «GererAdherent»	72
4.1 Traçabilité MCU/MC du cas d'utilisation «ConsulterTrafic»	74
4.2 Diagramme de séquence du cas d'utilisation «ConsulterTrafic»	75
4.3 Traçabilité MCU/MC du cas d'utilisation «ConsulterBanque»	75
4.4 Diagramme de séquence du cas d'utilisation «ConsulterBanque»	75
4.5 Traçabilité MCU/MC du cas d'utilisation «ConsulterAgence»	76
4.6 Diagramme de séquence du cas d'utilisation «ConsulterAgence»	76
4.7 Traçabilité MCU/MC du cas d'utilisation «ConsulterDAB»	77
4.8 Diagramme de séquence du cas d'utilisation «ConsultDAB»	77
4.9 Diagramme de classes entités de l'application	78
4.10 Table Banque	79
4.11 Table Agence	79
4.12 Table DAB	80
4.13 Table Catégorie	80
4.14 Table Tarif	81
4.15 Table Tarification	81
4.16 Table Gouvernorat	82
4.17 Table Délégation	82
4.18 Table Ville	83
4.19 Table User	83
4.20 Table Adherent	84
4.21 Table Traffic	84
4.22 Traçabilité MC/MI du cas d'utilisation «ConsulterTrafic»	85
4.23 Diagramme de composant du cas d'utilisation «ConsulterTrafic»	85
4.24 Traçabilité MC/MI du cas d'utilisation «ConsulterBanque»	86
4.25 Diagramme de composant du cas d'utilisation «ConsulterBanque»	86
4.26 Traçabilité MC/MI du cas d'utilisation «ConsulterAgence»	87
4.27 Diagramme de composant du cas d'utilisation «ConsulterBanque»	87
4.28 Traçabilité MC/MI du cas d'utilisation «ConsulterDAB»	88
4.29 Diagramme de composant du cas d'utilisation «ConsulterDAB»	88
4.30 Diagramme de déploiement du système	89
5.1 Architecture du système (3-tiers)	90
5.2 Interface «Accueil»	92
5.3 Interface «Connexion Administrateur»	93
5.4 Interface «Erreur connexion administrateur»	93
5.5 Interface «Connexion avec mot de passe correct»	94
5.6 Interface «Connexion avec succès»	94
5.7 Interface «Tableau de bord»	95
5.8 Interface «Mot de passe oublié»	96
5.9 Interface «Réinstaller mot de passe»	96

5.10 Interface «Vérifier email»	97
5.11 Interface «Succès d'envoi»	97
5.12 Interface «Réception de l'email»	98
5.13 Interface «Reconnexion»	98
5.14 Interface «Changer mot de passe»	99
5.15 Interface «Avertissement ancien mot de passe»	99
5.16 Interface «Reconnexion»	100
5.17 Interface «Reconnexion avec succès»	100
5.18 Interface «Consulter banque»	101
5.19 Interface «Ajouter banque»	102
5.20 Interface «Erreur identifiant banque existant»	102
5.21 Interface «Erreur vérifier les champs»	103
5.22 Interface «Informations banque saisies»	103
5.23 Interface «Ajout banque avec succès»	104
5.24 Interface «Consulter tarif»	105
5.25 Interface «Ajout tarif»	105
5.26 Interface «Erreur ID categorie existant»	106
5.27 Interface «Informations tarif saisies»	106
5.28 Interface «Ajout tarif avec succès»	107
5.29 Interface «Nouvelle tarif ajoutée»	107
5.30 Interface «Consulter tarification»	108
5.31 Interface «Modifier tarification»	109
5.32 Interface «Modification des informations»	109
5.33 Interface «Succès de modification»	110
5.34 Interface «Consultation mis à jour»	110
5.35 Interface «Accueil»	111
5.36 Interface «Comparateur»	111
5.37 Interface «Sélectionner catégorie»	112
5.38 Interface «Tarifs comparées d'une catégorie»	112
5.39 Interface «Tarifications comparées d'une tarif»	113
5.40 Interface «Sélectionner plusieurs tarifs»	113
5.41 Interface «Choisir les différentes tarifs»	114
5.42 Interface «Résultat du comparaison»	114

List of Tables

2.1	Affectation des cas d'utilisation aux acteurs	10
2.2	Affectation des priorités aux cas d'utilisation	10
2.3	Spécification du cas d'utilisation «S'authentifier»	12
2.4	Spécification du cas d'utilisation «Ajouter une banque»	13
2.5	Spécification du cas d'utilisation «Modifier une banque»	14
2.6	Spécification du cas d'utilisation «Supprimer une banque»	14
2.7	Spécification du cas d'utilisation «Ajouter une catégorie de tarif»	15
2.8	Spécification du cas d'utilisation «Modifier une catégorie de tarif»	16
2.9	Spécification du cas d'utilisation «Supprimer une catégorie de tarif»	16
2.10	Spécification du cas d'utilisation «Ajouter une tarif»	17
2.11	Spécification du cas d'utilisation «Modifier une tarif»	18
2.12	Spécification du cas d'utilisation «Supprimer une tarif»	18
2.13	Spécification du cas d'utilisation «Ajouter une tarification»	19
2.14	Spécification du cas d'utilisation «Modifier une tarification»	20
2.15	Spécification du cas d'utilisation «Supprimer une tarification»	20
2.16	Spécification du cas d'utilisation «Comparer les tarifications»	21
3.1	Spécification du cas d'utilisation «Ajouter une agence»	41
3.2	Spécification du cas d'utilisation «Modifier une agence»	41
3.3	Spécification du cas d'utilisation «Supprimer une agence»	42
3.4	Spécification du cas d'utilisation «AjouterDAB»	43
3.5	Spécification du cas d'utilisation «ModifierDAB»	43
3.6	Spécification du cas d'utilisation «SupprimerDAB»	44
3.7	Spécification du cas d'utilisation «AjouterGouvernorat»	45
3.8	Spécification du cas d'utilisation «ModifierGouvernorat»	45
3.9	Spécification du cas d'utilisation «SupprimerGouvernorat»	46
3.10	Spécification du cas d'utilisation «AjouterDelegation»	47
3.11	Spécification du cas d'utilisation «ModifierDelegation»	47
3.12	Spécification du cas d'utilisation «SupprimerDelegation»	48
3.13	Spécification du cas d'utilisation «AjouterVille»	49
3.14	Spécification du cas d'utilisation «ModifierVille»	49
3.15	Spécification du cas d'utilisation «SupprimerVille»	50
3.16	Spécification du cas d'utilisation «AjouterUtilisateur»	51
3.17	Spécification du cas d'utilisation «SupprimerUtilisateur»	51
3.18	Spécification du cas d'utilisation «SupprimerAdherent»	52
4.1	Spécification du cas d'utilisation «ConsulterTrafic»	73
4.2	Spécification du cas d'utilisation «ConsulterBanque»	73
4.3	Spécification du cas d'utilisation «ConsulterAgence»	74
4.4	Spécification du cas d'utilisation «ConsulterDAB»	74

5.1 Matériels de base	91
---------------------------------	----

Glossaire

UML	Unified Modeling Language
DAB	Distributeur Automatique de Billets
IP	Internet Protocol
MAC	Media Access Control
SQL	Structured Query Language
AGL	Atelier de Génie Logiciel
HFSQL	Hyper File Structured Query Language
SGBDR	Système de Gestion de Base de Données Relationnelle

Introduction générale

De nos jours, les systèmes d'information deviennent une nécessité fondamentale, surtout pour les grandes entreprises, bien évidemment, les banques. En effet, ces dernières répondent rapidement aux changements des besoins du marché et à augmenter leurs bénéfices en offrant des produits, des biens ou des services afin de satisfaire le besoin de la clientèle.

De ce fait, la satisfaction du client est au centre des préoccupations des banques et se concrétise par une gestion personnalisée de la relation client : comprendre les attentes des clients, les fidéliser et les inciter à consommer davantage.

C'est dans ce contexte que s'inscrit le présent projet réalisé dans le cadre d'un stage de fin d'études effectué au sein du siège social de la Banque Internationale Arabe de Tunisie (BIAT) pour l'obtention de la Licence Fondamentale en Informatique appliquée à la gestion.

Notre projet consiste à développer une application web, intitulée "Comparateur des Tarifications Bancaires Tunisiennes", destinée à la clientèle. Pour la modélisation de cette dernière, nous avons appliqué les méthodologies et les notions enseignées cette année en choisissant le processus unifié basé sur le langage de modélisation UML. En effet, ce processus permet de faciliter les tâches de la conception en les divisant sur quatre phases (création, élaboration, construction, transition).

Selon ces différentes phases, notre rapport est structuré en cinq grands chapitres.

D'abord, le premier chapitre décrit le contexte général du projet, la présentation de l'organisme d'accueil et la définition des grandes lignes du travail demandé.

Ensuite, le deuxième chapitre, «la phase de création», illustre le lancement du projet dans lequel nous déterminons les besoins fonctionnels et non fonctionnels puis l'identification des acteurs mais aussi des cas d'utilisation.

Puis, le troisième chapitre, «la phase d'élaboration», comporte une conception détaillée des cas d'utilisation de priorité 2, les diagrammes de séquences, le diagramme de déploiement ainsi que le diagramme de classe.

Quant au quatrième chapitre, «la phase de construction», il contient la conception des cas d'utilisation de priorité 3 et récupère les résultats des deux chapitres précédents pour compléter le diagramme de classe général.

Enfin, la dernière partie ou «la phase de transition» comprend le dernier cycle du processus unifié dédié à la confirmation et la mise en œuvre de l'application réalisée.

1 Cadre général du projet

Introduction

Au cours de ce premier chapitre, nous nous intéressons à décrire l'organisme d'accueil, dans lequel s'est déroulé notre projet de fin d'études. Ensuite nous passons à une description du contexte du projet qui comprend la problématique et la solution envisagée sans oublier la méthodologie de travail adoptée.

1.1 Présentation de l'organisme d'accueil



Figure 1.1: Logo de la société BIAT

La **Banque Internationale Arabe De Tunisie (BIAT)** est une société bancaire créée en 1976. Elle est aujourd’hui la première banque du pays et se classe au premier rang sur de nombreux indicateurs. Elle représente environ 17.4% de part de marché, en moyenne, sur le panel des dix premières banques de la place en termes de dépôts.

La BIAT a développé toutes les activités de banque et constitue un groupe bancaire avec ses filiales dans les domaines de l’assurance, de la gestion d’actifs, du capital-investissement ou de l’intermédiation boursière.

Appuyant son développement sur la proximité, l’engagement sociétal, la BIAT met son expertise et sa solidité au profit de ses clients et de l’économie de la Tunisie.

Son organisation, qui reflète sa stratégie de diversification, offre une lecture claire de ses principaux métiers. La complétude des métiers du Groupe BIAT et son organisation permettent de développer des synergies efficaces et une offre pertinente et complète pour l'ensemble de ses clientèles : particuliers, dont les Tunisiens résidant à l'étranger, professionnels, PME, grandes entreprises et institutionnels. Implantée sur tout le territoire, la BIAT compte 200 agences à fin 2015, à travers toute la Tunisie. Cotée à la Bourse de Tunis, la BIAT est une entreprise à capitaux très majoritairement tunisiens. [1]

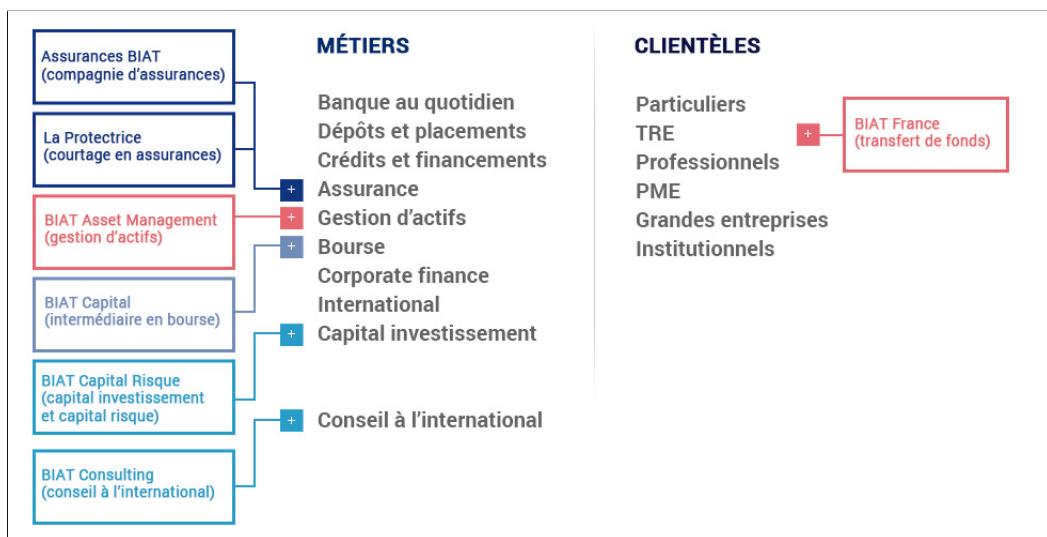


Figure 1.2: Les filiales du groupe bancaire BIAT

Fiche technique :

- Dénomination : Banque Internationale Arabe De Tunisie
- Nationalité : Tunisienne
- Directeur général : M. Mohamed Agrebi
- Siège social : 70-72 Avenue Habib Bourguiba – Tunis
- Téléphone : 71 13 10 00 - 31 31 10 00
- Site Web : www.biat.com.tn

1.2 Description du contexte du projet

Définition Tarification bancaire selon Mataf¹

La tarification bancaire est l'ensemble des frais qui sont pris par une banque envers ses clients lors de la vie d'un compte bancaire. Elle portera sur les conditions pratiquées par la banque pour les différentes opérations et prestations bancaires proposées à sa clientèle. Celle-ci donne aussi à la banque la possibilité de développer sa stratégie en fonction de ses

¹C'est un site financier qui propose des outils et des informations en ligne pour les cambistes

cibles clientèles. Elle peut par exemple dissuader un segment de clientèle par une tarification non compétitive.

Elle peut à l'inverse proposer des taux de crédit In Fine² attractif si elle recherche une clientèle haut de gamme. La tarification n'est donc pas simplement un enjeu de rentabilité pour la banque mais aussi un instrument de sa stratégie. [3]

1.2.1 Etude de l'Existant

Pour choisir les meilleurs produits bancaires au meilleur prix, il y'a lieu de passer par un Benchmark (voir annexe 1) et/ou comparatif des prix des différents produits et services bancaires, notamment pour les crédits (meilleurs taux et meilleure commission de mise en place). Certains frais sont connus à l'avance comme le coût d'une carte bancaire ou les frais de tenue de compte, mais les autres frais sont difficiles à prévoir, particulièrement ceux liés aux opérations de virements, de remises chèque, les incidents de paiement. De ce fait, cette application va permettre aux clients désireux de pouvoir faire une comparaison entre les taux des différents banques par conséquent cette application offre aux clients un gain de temps.

1.2.2 Objectif du projet

Pour pouvoir se situer et choisir sa banque idéale par le biais des tarifs les plus compétitifs, l'outil développé par nos soins est un comparateur de tarifs, qui consiste à choisir les produits et les services appropriés par le client et de faire sortir les tarifs appliqués en standard par les banques. Cet outil, permettrait aux clients particuliers ou professionnels de choisir la banque la plus compétitive dans un premier temps, il permet également aux clients déjà bancarisés de négocier les tarifs appliqués par sa banque sur la base du comparatif des tarifs de toutes les banques qui résultent de cet outil, et il peut obtenir des tarifs préférentiels sur la base d'une bonne négociation.

1.2.3 Méthode et langage de modélisation utilisés

Pour définir l'architecture de notre projet et afin de le modéliser d'une manière claire et précise, nous avons utilisé le processus uniifié.

Le Processus Unifié (PU) de développement logiciel est une méthode générique qui repose sur l'approche orientée-objet et utilise UML comme langage de modélisation visuelle. Son but est d'assurer la production d'un logiciel de bonne qualité, qui satisfait les besoins de ses utilisateurs et qui est produit et livré dans le respect des délais et des coûts prévus. Il est piloté par les cas d'utilisation, centré sur l'architecture, itératif et incrémental. [2]

- ✓ Piloté par les cas d'utilisation: le projet est mené en tenant compte des besoins et des exigences des utilisateurs qui sont modélisés en cas d'utilisation décrits avec précision.
- ✓ Centré sur l'architecture : L'architecture décrit le comment le faire et doit prévoir la réalisation de tous les cas d'utilisation, en traitant en premier les cas

²Ce type de crédit consiste à rembourser de la totalité du capital ne s'effectue que sur la dernière échéance. Toutes les autres échéances ne servent qu'à payer des intérêts (et les assurances).

d'utilisation les plus risqués / critiques, les plus importants pour le client, les plus représentatifs du système. L'architecture se dévoile peu à peu, au rythme de la spécification et de la maturation des cas d'utilisation, qui favorisent, à leur tour, le développement d'un nombre croissant de cas d'utilisation. Ce processus se poursuit jusqu'à ce que l'architecture soit jugée stable.

- ✓ Itératif et incrémental: Il se déroule en quatre phases (incubation, élaboration, construction et transition). Chaque phase répète un nombre de fois une série d'itérations. Et chaque itération est composée de cinq activités à savoir la capture des besoins, l'analyse, la conception, l'implémentation et l'activité de test. Après chaque itération on a un nouvel incrément.

Unified Modeling Language (UML) : est un langage de modélisation normalisé basé sur les concepts orientés objets. Il offre un standard de modélisation afin spécifier, visualiser, modifier et construire les documents nécessaires au bon développement orienté objet d'un logiciel. [2]

Conclusion :

Dans ce chapitre, nous avons dévoilé l'idée générale et les objectifs de notre sujet et présenté le choix de la méthodologie adoptée ainsi nous avons bien compris les tâches à réaliser par la suite. Dans le chapitre suivant, nous allons entamer la création qui est la première phase du processus unifié.

2 La phase de création

Introduction

Ce chapitre présente la première phase du processus unifié et le point de départ de toute application à développer permettant d'étudier le contexte du système.

Nous allons mettre en évidence les différentes fonctionnalités du système qui doit satisfaire les besoins fonctionnels de tous les utilisateurs, et doit aussi respecter une certaine exigence non fonctionnelle.

2.1 Capture des besoins

Au cours de cette activité, nous allons illustrer les besoins fonctionnels ainsi que les besoins non fonctionnels, en premier lieu. En deuxième lieu, nous allons identifier les acteurs principaux et les différents cas d'utilisation. Celles-ci seront classées par priorité. Ensuite nous allons présenter le modèle de cas d'utilisation raffiné.

2.1.1 Identification des besoins fonctionnels

L'ensemble de fonctionnalités offertes par notre future application doit correspondre aux attentes des utilisateurs. Ces attentes sont illustrées par un ensemble de besoins fonctionnels qui sont les suivants:

- L'authentification: l'administrateur de notre application doit saisir son e-mail et son mot de passe pour que le système s'assure de son existence avant de lui donner le droit d'accès.
- Gestion des banques: l'application doit donner la possibilité à l'administrateur de créer, consulter, modifier ou supprimer les informations concernant les banques de toute la Tunisie.
- Gestion des Agences: l'application doit donner la possibilité à l'administrateur de créer, consulter, modifier ou supprimer les informations concernant les agences des banques de toute la Tunisie.
- Gestion des DAB: l'application doit donner la possibilité à l'administrateur de créer, consulter, modifier ou supprimer les informations concernant les DAB des banques de toute la Tunisie.
- Gestion des gouvernorats: l'application doit donner la possibilité à l'administrateur de créer, consulter, modifier ou supprimer les informations concernant les gouvernorats de toute la Tunisie.

- Gestion des délégations: l'application doit donner la possibilité à l'administrateur de créer, consulter, modifier ou supprimer les informations concernant les délégations de toute la Tunisie.
- Gestion des villes: l'application doit donner la possibilité à l'administrateur de créer, consulter, modifier ou supprimer les informations concernant les villes de toute la Tunisie.
- Gestion des tarifs: l'application doit donner la possibilité à l'administrateur de créer, consulter, modifier ou supprimer les informations concernant les tarifs des banques de toute la Tunisie.
- Gestion des catégories des tarifs: l'application doit donner la possibilité à l'administrateur de créer, consulter, modifier ou supprimer les informations concernant les catégories similaires des tarifs des banques de toute la Tunisie.
- Gestion des tarifications des tarifs: l'application doit donner la possibilité à l'administrateur de créer, consulter, modifier ou supprimer les informations concernant les tarifications des tarifs des banques de toute la Tunisie.
- Gestion des administrateurs: l'application doit donner la possibilité à un ou plusieurs administrateurs de créer, consulter ou supprimer les informations concernant les administrateurs du back office.
- Gestion des adhérents: l'application doit donner la possibilité à l'administrateur de consulter, modifier ou supprimer les informations concernant les adhérents inscrits.
- Gestion du trafic: l'application doit donner la possibilité à l'administrateur de consulter les informations concernant l'identité physique (adresse IP, MAC, date du visite) du visiteur du site.
- Comparaison des tarifications: l'application doit donner la possibilité de comparer les tarifications des banques de toute la Tunisie si l'utilisateur (administrateur ou bien adhérent) est authentifié selon la priorité du choix sinon, l'adhérent ne peut comparer qu'une seule tarification entre 2 différentes banques.

2.1.2 Identification des besoins non fonctionnels

Une fois les besoins fonctionnels sont identifiés, les besoins non fonctionnels doivent être pris en compte tout au long du cycle de développement de l'application.

Ces besoins expriment une certaine amélioration des services offerts en décrivant les contraintes techniques, ergonomiques et esthétiques sans aucune intervention sur les apports métiers de l'application. Dans notre cas, le système doit garantir les besoins suivants :

- **La sécurité:** L'application doit assurer que chaque utilisateur ne peut accéder qu'à son espace et qu'après une authentification
- **La performance:** L'application doit répondre aux exigences des usagers d'une manière optimale.
- **La fiabilité:** Le système doit toujours être en mesure de fonctionner correctement sans risque d'erreur et de panne.
- **L'exigence de réalisation:** l'application web doit se faire dans l'environnement WEB-DEV.
- **La convivialité :** L'application doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales, c'est-à-dire, simples, ergonomiques et adaptées à l'utilisateur.
- **L'évolutivité :** Le système doit avoir la capacité de s'adapter aux changements et aux futurs besoins de l'entreprise.

2.1.3 Identification des acteurs et des cas d'utilisation

Cette partie définit les acteurs ainsi que les cas d'utilisation.

Les acteurs : Un acteur est une personne, un matériel ou un logiciel qui interagit avec le système dans le but de réaliser une ou plusieurs tâches.

Les acteurs en interaction avec notre système sont :

- ✓ Un administrateur du back office qui gère les banques, les agences, les DAB, les administrateurs, les gouvernorats, les délégations, les villes, les catégories des tarifs, les tarifs et les tarifications. Aussi, il peut comparer les tarifications pour tester l'affichage.
- ✓ Un Adhérent qui consulte les banques, les agences, les DAB, leurs géolocalisations et compare les tarifications bancaires.

Les cas d'utilisation: un cas d'utilisation est une fonctionnalité du système produisant un résultat satisfaisant pour l'utilisateur. [2]

L'ensemble des acteurs et des cas d'utilisation sont résumés dans le tableau ci-dessous :

Acteurs	Cas d'utilisation
Administrateur	S'authentifier Gérer les banques Gérer les agences Gérer les DAB Gérer les catégories des tarifs Gérer les tarifs Gérer les tarifications des tarifs Comparer les tarifications Gérer les gouvernorats Gérer les délégations

	<p>Gérer les villes Gérer les administrateurs Gérer les adhérents Consulter le trafic</p>
--	---

Adhérent	S'authentifier Comparer les tarifications Consulter les banques Consulter les agences Consulter les DAB
----------	---

Table 2.1: Affectation des cas d'utilisation aux acteurs

2.1.4 Affectation des priorités aux cas d'utilisation

Les différents cas d'utilisation, présentés ci-dessus, sont ordonnés selon la dépendance entre eux. Autrement dit, la réalisation de certains cas d'utilisation nécessite la réalisation d'un ou de plusieurs autres cas d'utilisation. Le tableau suivant représente l'affectation des priorités qui sont relatives à l'ordre de développement des cas d'utilisation.

Cas d'utilisation	Acteurs	Priorité
S'authentifier	Administrateur Adhérent	1
Gérer les banques	Administrateur	1
Gérer les catégories des tarifs	Administrateur	1
Gérer les tarifs	Administrateur	1
Gérer les tarifications des tarifs	Administrateur	1
Comparer les tarifications	Administrateur Adhérent	1
Gérer les agences	Administrateur	2
Gérer les DAB	Administrateur	2
Gérer les gouvernorats	Administrateur	2
Gérer les délégations	Administrateur	2
Gérer les villes	Administrateur	2
Gérer les administrateurs	Administrateur	2
Gérer les adhérents	Administrateur	2
Consulter le trafic	Administrateur	3
Consulter les banques	Adhérent	3
Consulter les agences	Adhérent	3
Consulter les DAB	Adhérent	3

Table 2.2: Affectation des priorités aux cas d'utilisation

2.1.5 Diagramme de cas d'utilisation initial

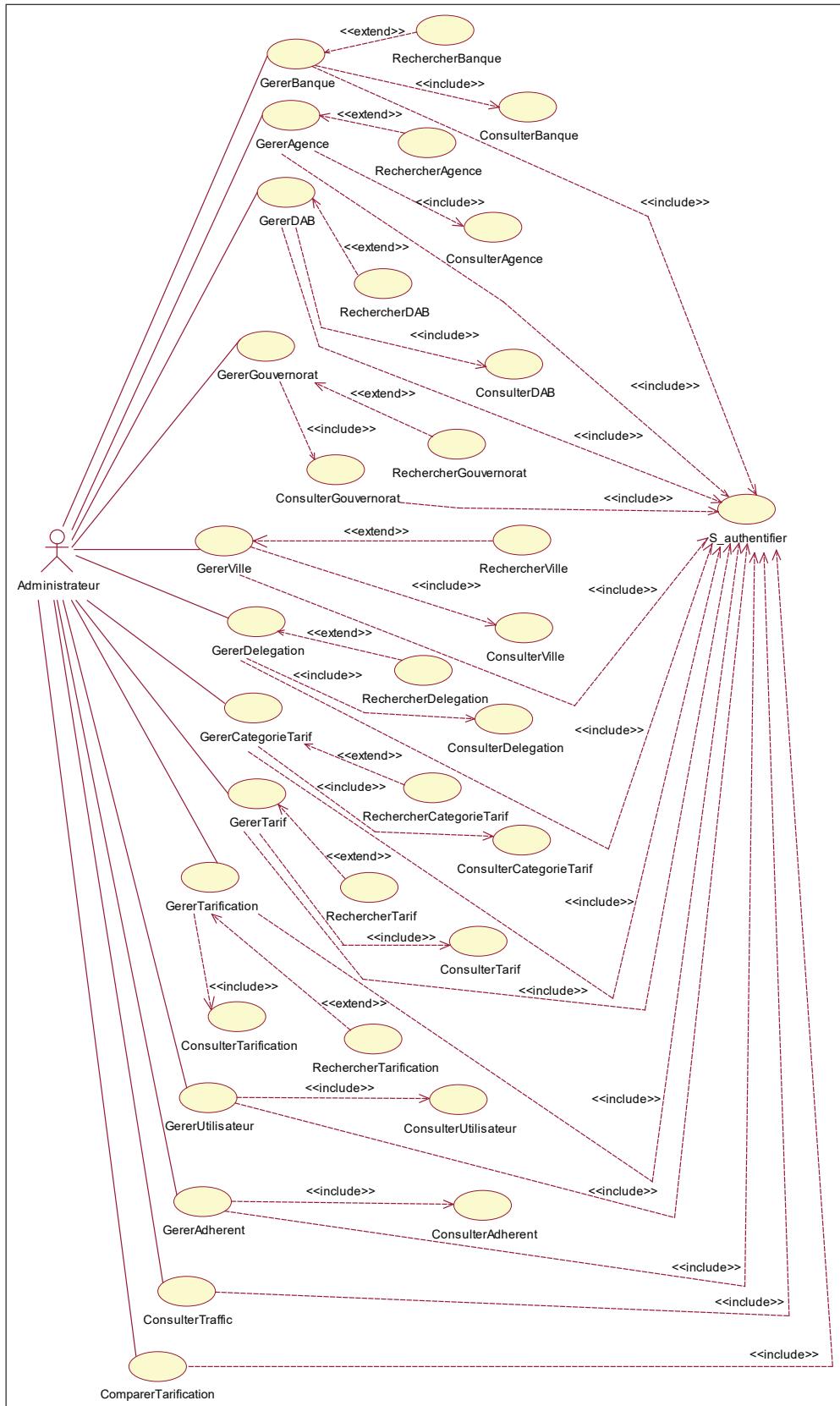


Figure 2.1: Diagramme du cas d'utilisation initial

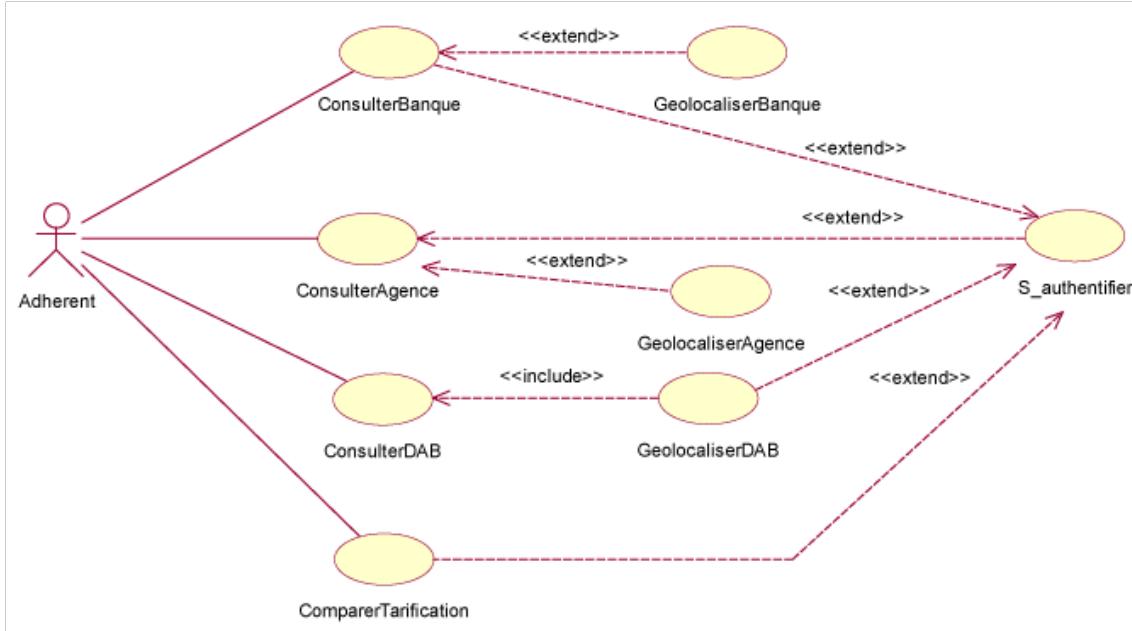


Figure 2.2: Diagramme du cas d'utilisation initial partie 2

2.1.6 Raffinement des cas d'utilisation de priorité 1

Au cours de cette activité nous allons détailler les cas d'utilisation les plus prioritaires en décrivant, dans chaque cas, les pré-conditions et les post-conditions ainsi que le scénario principal, les extensions et éventuellement les exceptions liées à ce cas.

A- Raffinement du cas d'utilisation «S'authentifier» :

niceblueCas d'utilisation	S'authentifier
niceblueActeurs	Administrateur et adhérent
nicebluePré-condition	Email et mot de passe valides
nicebluePost-condition	Administrateur ou adhérent authentifié
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur ou l'adhérent saisit son email et son mot de passe. - Il clique sur le bouton «Connexion». - Le système vérifie la combinaison email et mot de passe. S'ils sont corrects, le système affiche l'interface du menu principal approprié.
niceblueExtensions	L'administrateur ou l'adhérent peut choisir de réinitialiser son mot de passe s'il est oublié.
niceblueExceptions	Si le email et/ou le mot de passe sont erronés, le système affiche un message d'erreur demandant à l'administrateur ou l'adhérent de s'authentifier avec un email et un mot de passe valides.

Table 2.3: Spécification du cas d'utilisation «S'authentifier»

B- Raffinement du cas d'utilisation «Gérer les banques» :

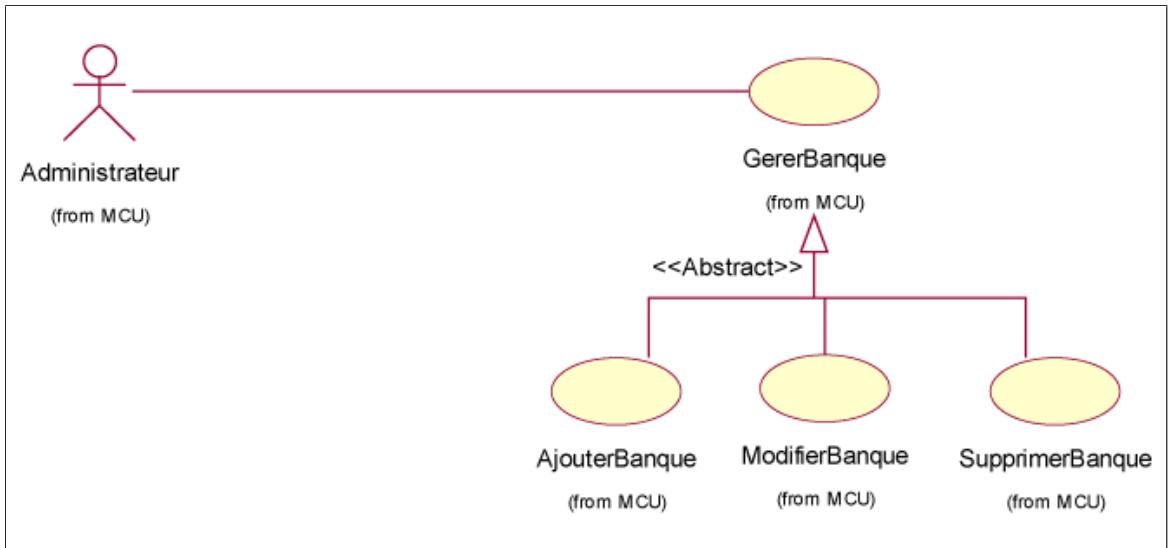


Figure 2.3: Raffinement du cas d'utilisation «GererBanque»

- Raffinement du cas d'utilisation «Ajouter une banque» :

niceblueCas d'utilisation	AjouterBanque
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Banque ajoutée
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Nouvelle banque» de l'interface de consultation. - Il saisit l'identifiant unique de la nouvelle banque. - Le système vérifie l'inexistence de cet identifiant dans la base. - L'administrateur continue à saisir les informations restantes de la banque. - Il clique sur le bouton «valider». - Le système enregistre les données saisies. - Il affiche un message indiquant l'ajout avec succès de la nouvelle banque.
niceblueExceptions	<ul style="list-style-type: none"> - Si un champ obligatoire manque la saisie, le système affiche un message d'erreur. - Si l'identifiant saisi est existant dans la base, le système affiche un message d'erreur. - La taille de l'identifiant est égale à 2, sinon le système affiche un message d'erreur.

Table 2.4: Spécification du cas d'utilisation «Ajouter une banque»

- Raffinement du cas d'utilisation «Modifier une banque» :

niceblueCas d'utilisation	ModifierBanque
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Informations modifiées d'une banque
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit la banque à modifier depuis l'interface de consultation. - Il clique sur l'icône de la modification. - Le système affiche la page de la modification avec les informations de la banque choisie. - L'administrateur modifie les informations désirées. - Il clique sur le bouton «valider». - Le système met à jour la banque avec les nouvelles informations. - Il affiche un message indiquant la modification avec succès de la banque.
niceblueExceptions	Si un champ obligatoire manque la saisie ou bien ne respecte pas la format de l'écriture (pour le site web par exemple), le système affiche un message d'erreur.

Table 2.5: Spécification du cas d'utilisation «Modifier une banque»

- Raffinement du cas d'utilisation «Supprimer une banque» :

niceblueCas d'utilisation	SupprimerBanque
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Banque supprimée
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit la banque à supprimer depuis l'interface de consultation. - Il clique sur l'icône de suppression. - Le système affiche un message d'avertissement indiquant l'accord de suppression. - L'administrateur clique sur le bouton «Oui». - Le système affiche un message indiquant la suppression avec succès de la banque.
niceblueExceptions	Si l'administrateur clique sur le bouton «Non», la banque ne sera pas supprimée.

Table 2.6: Spécification du cas d'utilisation «Supprimer une banque»

C- Raffinement du cas d'utilisation «Gérer les catégories des tarifs» :

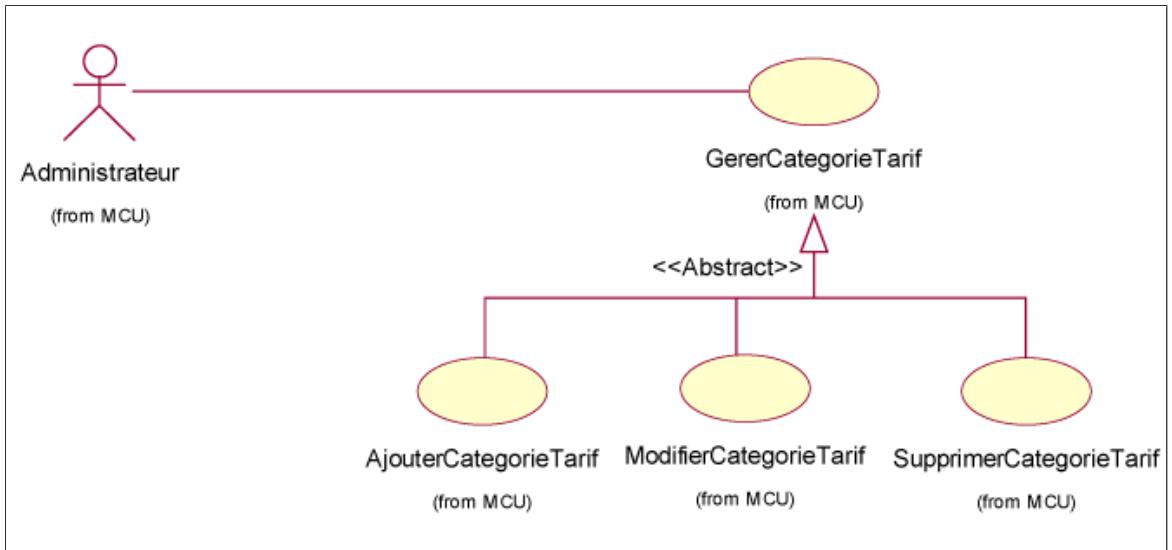


Figure 2.4: Raffinement du cas d'utilisation «GererCategorieTarif»

- Raffinement du cas d'utilisation «Ajouter une Categorie de tarifs» :

niceblueCas d'utilisation	AjouterCategorieTarif
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Catégorie de tarifs ajoutée
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Nouvelle catégorie» de l'interface de consultation. - L'administrateur saisit l'identifiant unique de la nouvelle catégorie. - Le système vérifie l'inexistence de cet identifiant dans la base. - L'administrateur continue à saisir le nom de la catégorie. - Il clique sur le bouton «valider». - Le système enregistre les données saisies. - Il affiche un message indiquant l'ajout avec succès de la nouvelle catégorie.
niceblueExceptions	<ul style="list-style-type: none"> Si un champ obligatoire manque la saisie, le système affiche un message d'erreur. - Si l'identifiant de la nouvelle catégorie est existant, le système affiche un message d'erreur.

Table 2.7: Spécification du cas d'utilisation «Ajouter une catégorie de tarif»

- Raffinement du cas d'utilisation «Modifier une catégorie de tarif» :

niceblueCas d'utilisation	ModifierCategorieTarif
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Informations modifiées d'une catégorie de tarif
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit la catégorie à modifier depuis l'interface de consultation. - Il clique sur l'icône de la modification. - Le système affiche la page de la modification avec les informations de la catégorie choisie. - L'administrateur modifie les informations désirées. - Il clique sur le bouton «valider». - Le système met à jour la catégorie avec les nouvelles informations. - Il affiche un message indiquant la modification avec succès de la catégorie.
niceblueExceptions	Si un champ obligatoire manque la saisie, le système affiche un message d'erreur.

Table 2.8: Spécification du cas d'utilisation «Modifier une catégorie de tarif»

- Raffinement du cas d'utilisation «Supprimer une catégorie de tarif» :

niceblueCas d'utilisation	SupprimerCategorieTarif
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Catégorie supprimée
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit la catégorie à supprimer depuis l'interface de consultation. - Il clique sur l'icône de suppression. - Le système affiche un message d'avertissement indiquant l'accord de suppression. - L'administrateur clique sur le bouton «Oui». - Le système affiche un message indiquant la suppression avec succès de la catégorie.
niceblueExceptions	Si l'administrateur clique sur le bouton «Non», la catégorie ne sera pas supprimée.

Table 2.9: Spécification du cas d'utilisation «Supprimer une catégorie de tarif»

D- Raffinement du cas d'utilisation «Gérer les tarifs» :

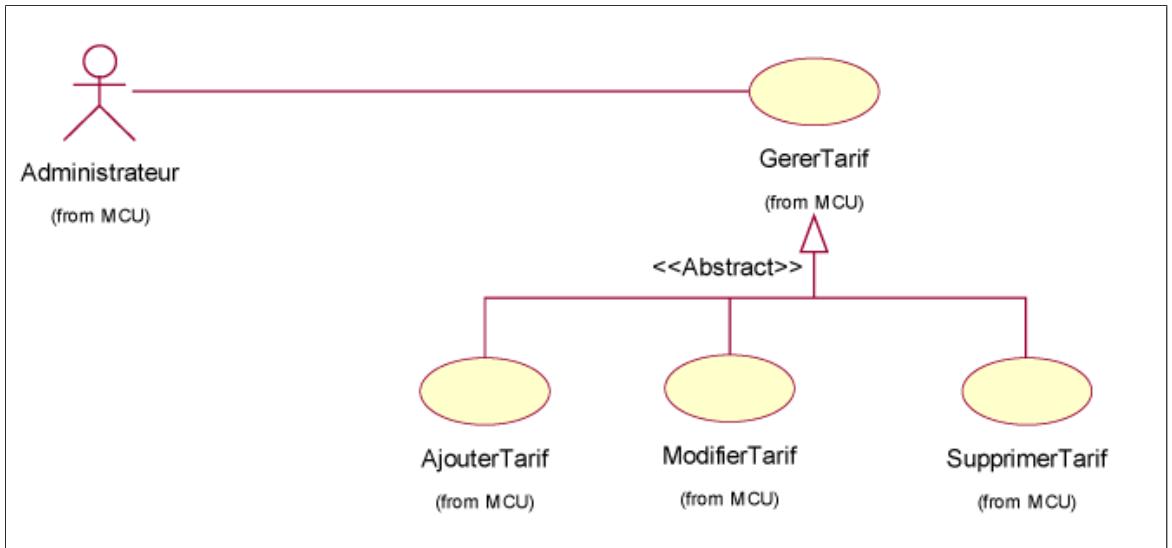


Figure 2.5: Raffinement du cas d'utilisation «Gérer Tarif»

- Raffinement du cas d'utilisation «Ajouter une tarif» :

niceblueCas d'utilisation	AjouterTarif
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Tarif ajoutée
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Nouvelle tarif» de l'interface de consultation. - Il saisit l'identifiant unique de la catégorie correspondante à la nouvelle tarif. - Le système vérifie l'existence de cet identifiant dans la base. - L'administrateur continue à saisir les informations restantes de la tarif. - Il clique sur le bouton «valider». - Le système enregistre les données saisies. - Il affiche un message indiquant l'ajout avec succès de la nouvelle tarif.
niceblueExceptions	<ul style="list-style-type: none"> - Si un champ obligatoire manque la saisie ou bien ne respecte pas la format de l'écriture (pour l'assiette par exemple), le système affiche un message d'erreur. - Si l'identifiant de la nouvelle tarif est existant, le système affiche un message d'erreur.

Table 2.10: Spécification du cas d'utilisation «Ajouter une tarif»

- Raffinement du cas d'utilisation «Modifier une tarif» :

niceblueCas d'utilisation	ModifierTarif
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Informations modifiées d'une tarif
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit la tarif à modifier depuis l'interface de consultation. - Il clique sur l'icône de la modification. - Le système affiche la page de la modification avec les informations de la tarif choisie. - L'administrateur modifie les informations désirées. - Il clique sur le bouton «valider». - Le système met à jour la tarif avec les nouvelles informations. - Il affiche un message indiquant la modification avec succès de la tarif.
niceblueExceptions	Si un champ obligatoire manque la saisie ou bien ne respecte pas la format de l'écriture (pour l'assiette par exemple), le système affiche un message d'erreur.

Table 2.11: Spécification du cas d'utilisation «Modifier une tarif»

- Raffinement du cas d'utilisation «Supprimer une tarif» :

niceblueCas d'utilisation	SupprimerTarif
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Tarif supprimée
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit la catégorie à supprimer depuis l'interface de consultation. - Il clique sur l'icône de suppression. - Le système affiche un message d'avertissement indiquant l'accord de suppression. - L'administrateur clique sur le bouton «Oui». - Le système affiche un message indiquant la suppression avec succès de la tarif.
niceblueExceptions	Si l'administrateur clique sur le bouton «Non», la tarif ne sera pas supprimée.

Table 2.12: Spécification du cas d'utilisation «Supprimer une tarif»

E- Raffinement du cas d'utilisation «Gérer les tarifications» :

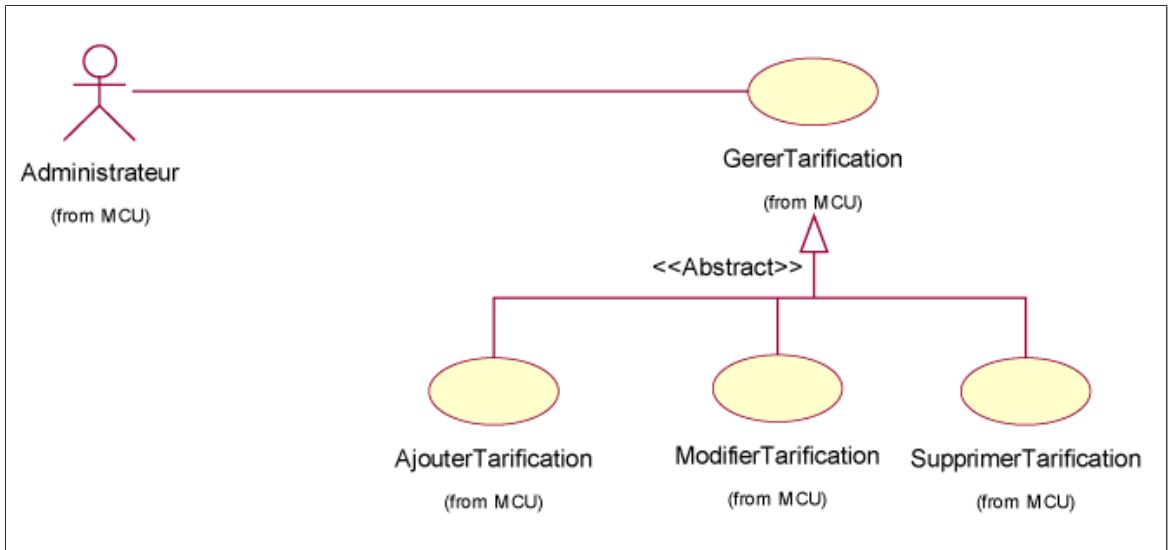


Figure 2.6: Raffinement du cas d'utilisation «Gérer les tarifications»

- Raffinement du cas d'utilisation «Ajouter une tarification» :

niceblueCas d'utilisation	AjouterTarification
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Tarification ajoutée
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Nouvelle tarification» de l'interface de consultation. - L'administrateur saisit l'identifiant unique de la tarif et l'identifiant unique de la banque correspondante à la nouvelle tarification. - Le système vérifie l'existence de la combinaison de ces deux identifiants dans la base. - L'administrateur continue à saisir les informations restantes de la tarification - Il clique sur le bouton «valider». - Le système enregistre les données saisies. - Il affiche un message indiquant l'ajout avec succès de la nouvelle tarification
niceblueExceptions	<ul style="list-style-type: none"> - Si un champ obligatoire manque la saisie ou bien ne respecte pas la format de l'écriture (pour la valeur par exemple), le système affiche un message d'erreur. - Si la combinaison des deux identifiants n'existe pas, le système affiche un message d'erreur.

Table 2.13: Spécification du cas d'utilisation «Ajouter une tarification»

- Raffinement du cas d'utilisation «Modifier une tarification» :

niceblueCas d'utilisation	ModifierTarification
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Informations modifiées d'une tarification
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit la tarification à modifier depuis l'interface de consultation. - Il clique sur l'icône de la modification. - Le système affiche la page de la modification avec les informations de la tarification choisie. - L'administrateur modifie les informations désirées. - Il clique sur le bouton «valider». - Le système met à jour la tarification avec les nouvelles informations. - Il affiche un message indiquant la modification avec succès de la tarification.
niceblueExceptions	Si un champ obligatoire manque la saisie ou bien ne respecte pas la format de l'écriture (pour la valeur par exemple), le système affiche un message d'erreur.

Table 2.14: Spécification du cas d'utilisation «Modifier une tarification»

- Raffinement du cas d'utilisation «Supprimer une tarification» :

niceblueCas d'utilisation	SupprimerTarification
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Tarification supprimée
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit la tarification à supprimer depuis l'interface de consultation. - Il clique sur l'icône de suppression. - Le système affiche un message d'avertissement indiquant l'accord de suppression. - L'administrateur clique sur le bouton «Oui». - Le système affiche un message indiquant la suppression avec succès de la tarification.
niceblueExceptions	Si l'administrateur clique sur le bouton «Non», la tarification ne sera pas supprimée.

Table 2.15: Spécification du cas d'utilisation «Supprimer une tarification»

F- Raffinement du cas d'utilisation «Comparer les tarifications» :

niceblueCas d'utilisation	ComparerTarification
niceblueActeurs	Administrateur et Adhérent
nicebluePré-condition	Administrateur ou adhérent authentifié
nicebluePost-condition	Tarifications des banques sélectionnées comparées
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur ou l'adhérent sélectionne les tarifs ou la catégorie des tarifs à comparer. - Il sélectionne ensuite les banques à comparer. - Le système compare les valeurs des tarifications puis affiche les banques sélectionnées par ordre croissant des valeurs comparées.
niceblueExceptions	<ul style="list-style-type: none"> - Si les tarifs et/ou les banques ne sont pas sélectionnées, le système n'affiche rien. - Si l'adhérent n'est pas authentifié, il peut sélectionner qu'une seule tarif et 2 différentes banques.

Table 2.16: Spécification du cas d'utilisation «Comparer les tarifications»

2.2 Conception

C'est une phase qui façonne le système en lui donnant une forme et une architecture bien précise, pour élaborer cette phase nous allons utiliser :

- Traçabilité entre le modèle de cas d'utilisation (MCU) et le modèle de conception (MC).
- Le diagramme de séquence : représente graphiquement l'interaction entre les acteurs et le système selon un point de vue temporel.

2.2.1 La conception des cas d'utilisation

A- La conception du cas d'utilisation «S_authentifier»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

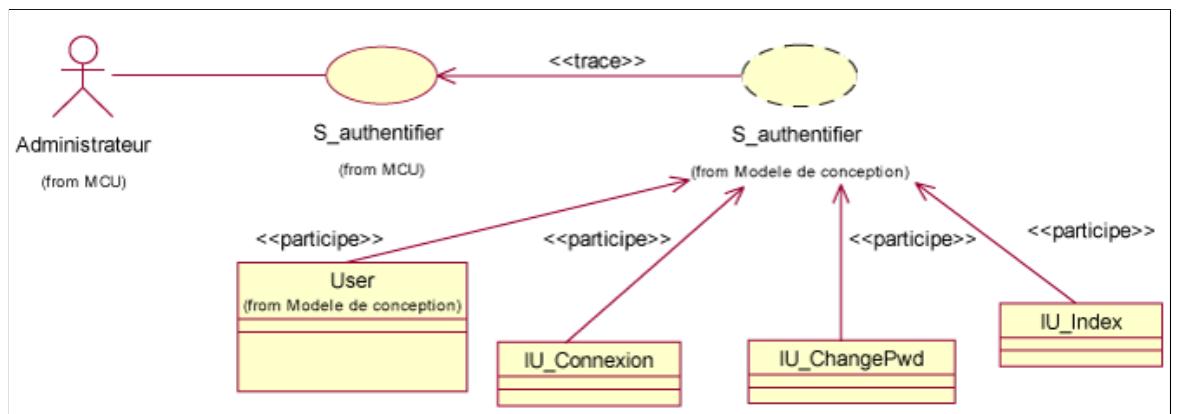


Figure 2.7: Traçabilité MCU/MC du cas d'utilisation «S_authentifier»

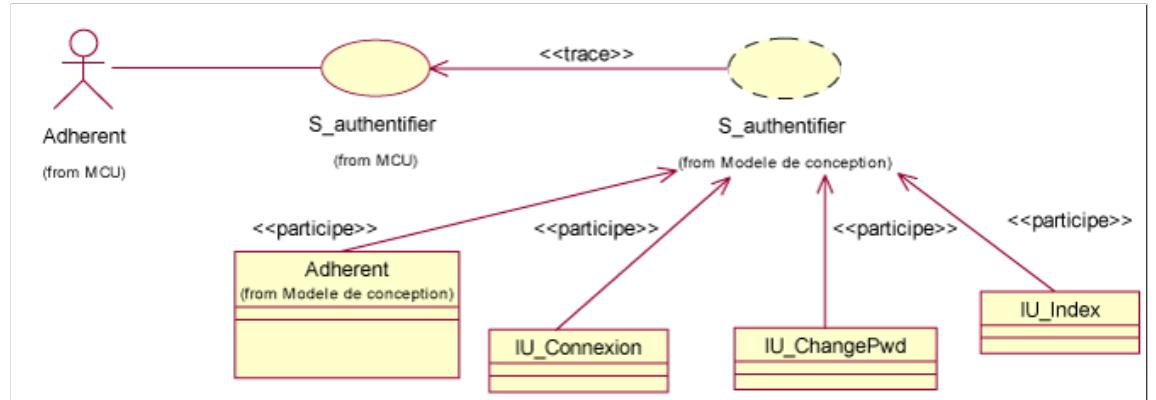


Figure 2.8: Traçabilité MCU/MC du cas d'utilisation «Sauthentifier» pour l'adherent

- Diagramme de séquence :

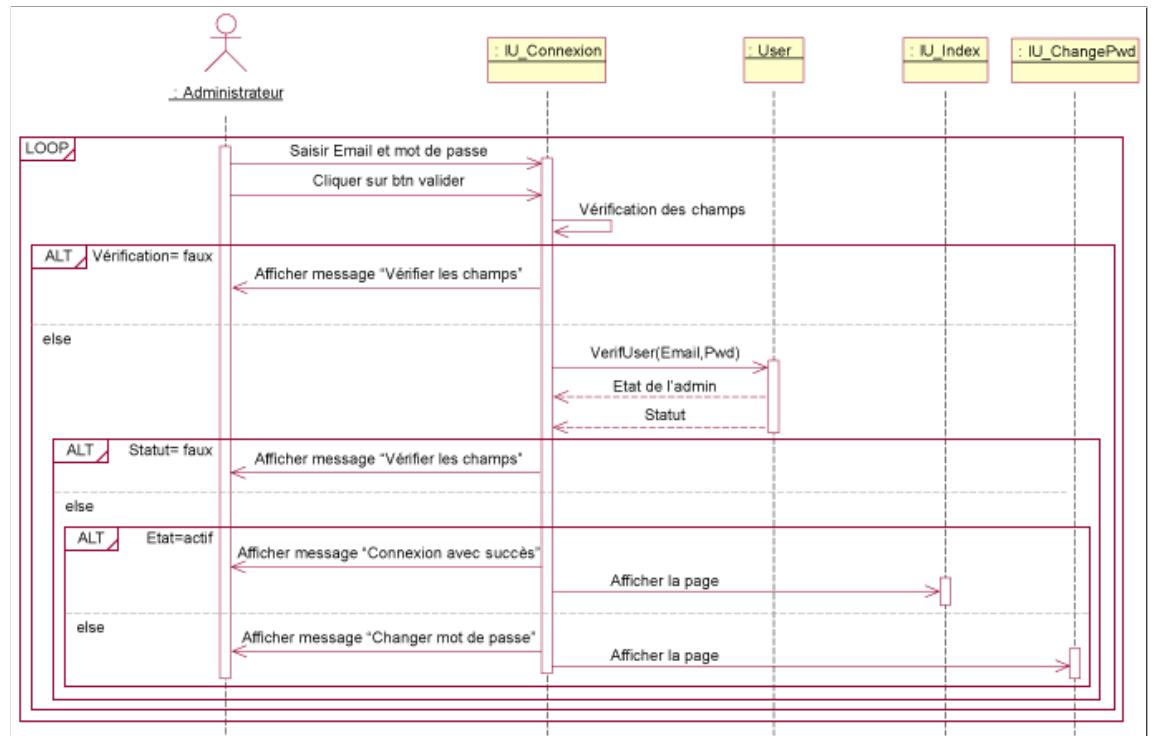


Figure 2.9: Diagramme de séquence du cas d'utilisation «S_authentifier»

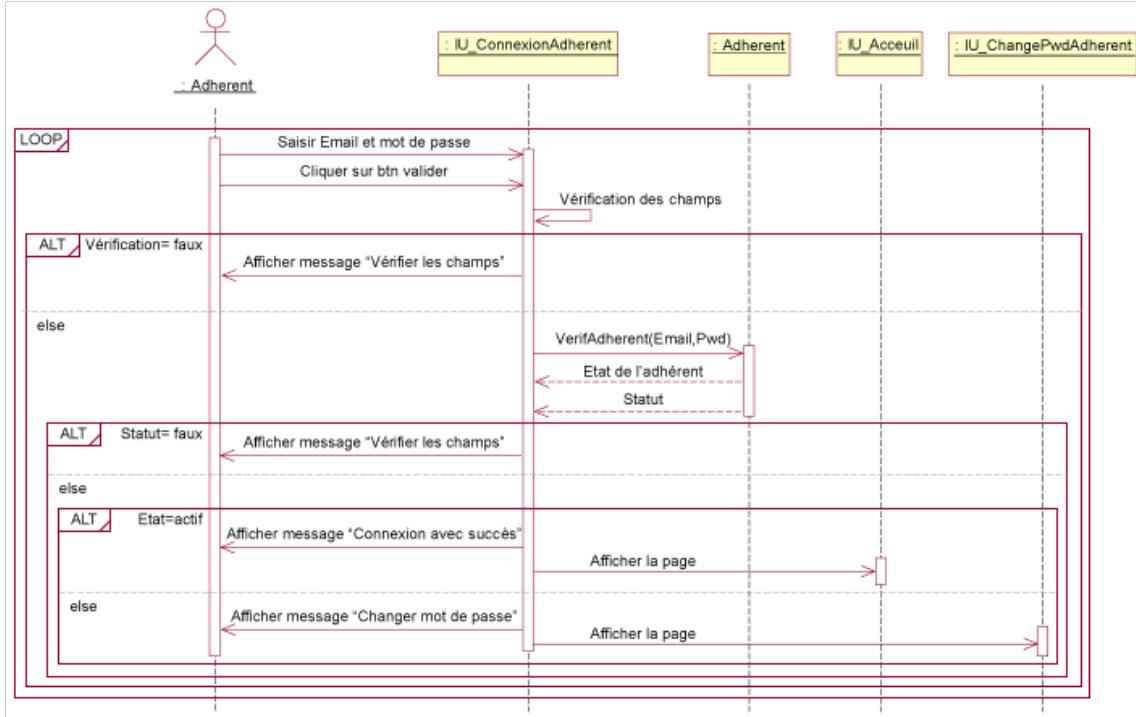


Figure 2.10: Diagramme de séquence du cas d'utilisation «Sauthentifier» pour l'adhérent

B- La conception du cas d'utilisation «GererBanque»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

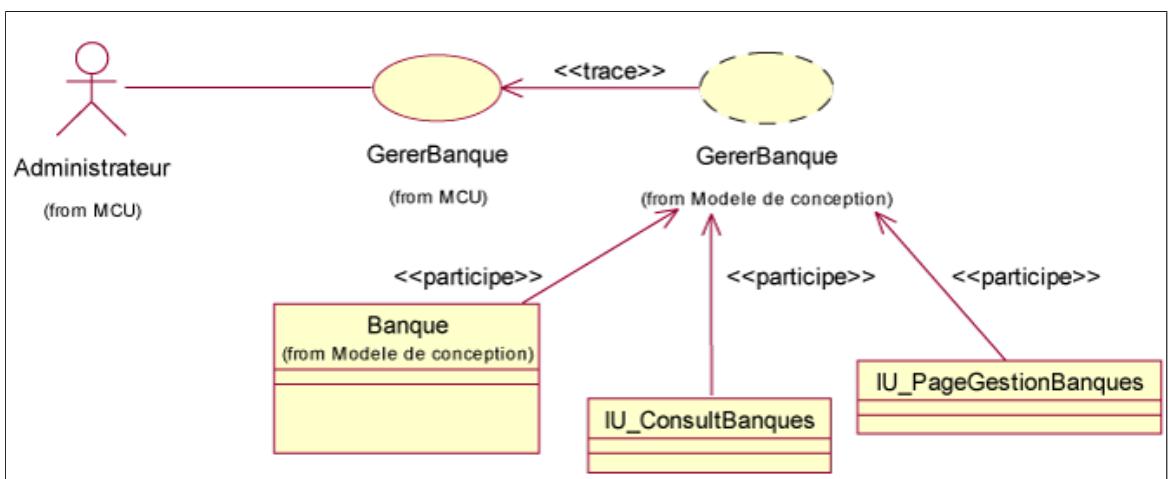


Figure 2.11: Traçabilité MCU/MC du cas d'utilisation «GererBanque»

- Diagrammes de séquence :

- Diagramme de séquence du cas d'utilisation «AjouterBanque»:

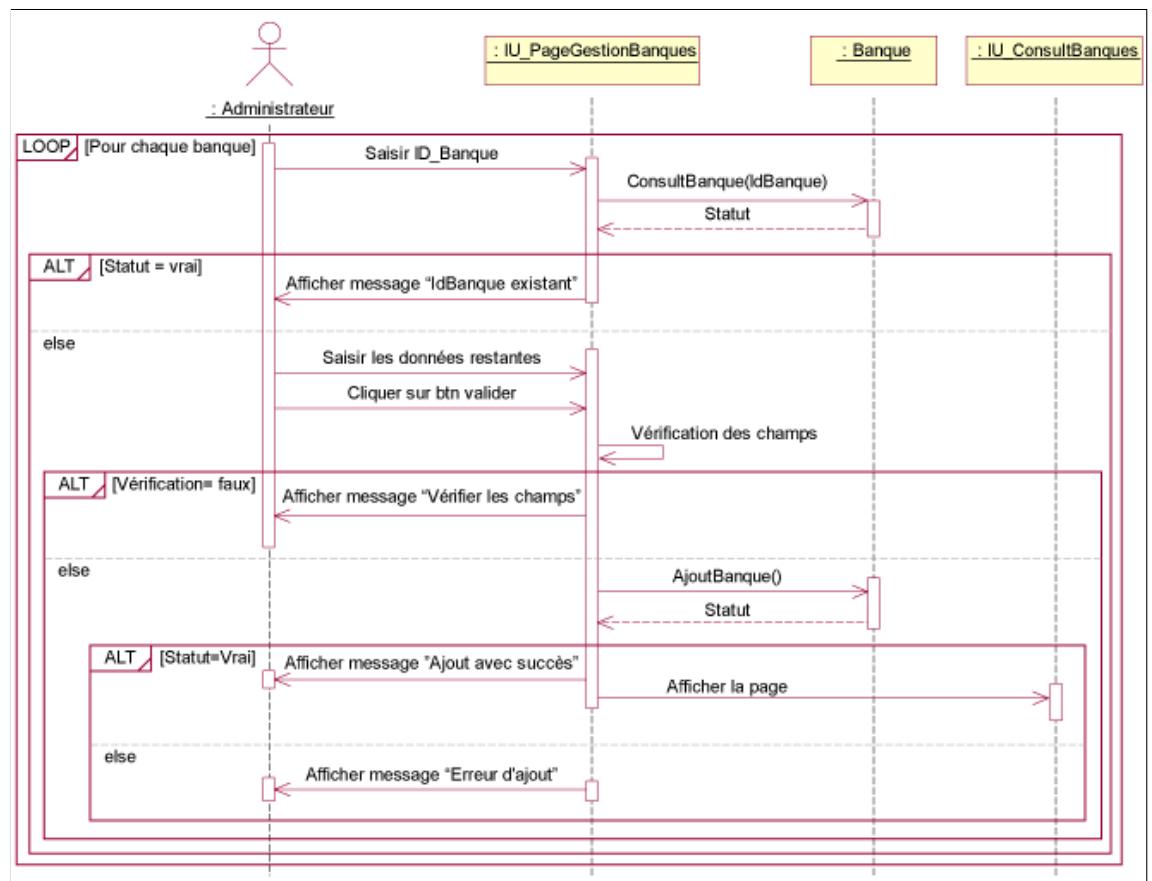


Figure 2.12: Diagramme de séquence du cas d'utilisation
«AjouterBanque»

■ Diagramme de séquence du cas d'utilisation «ModifierBanque»:

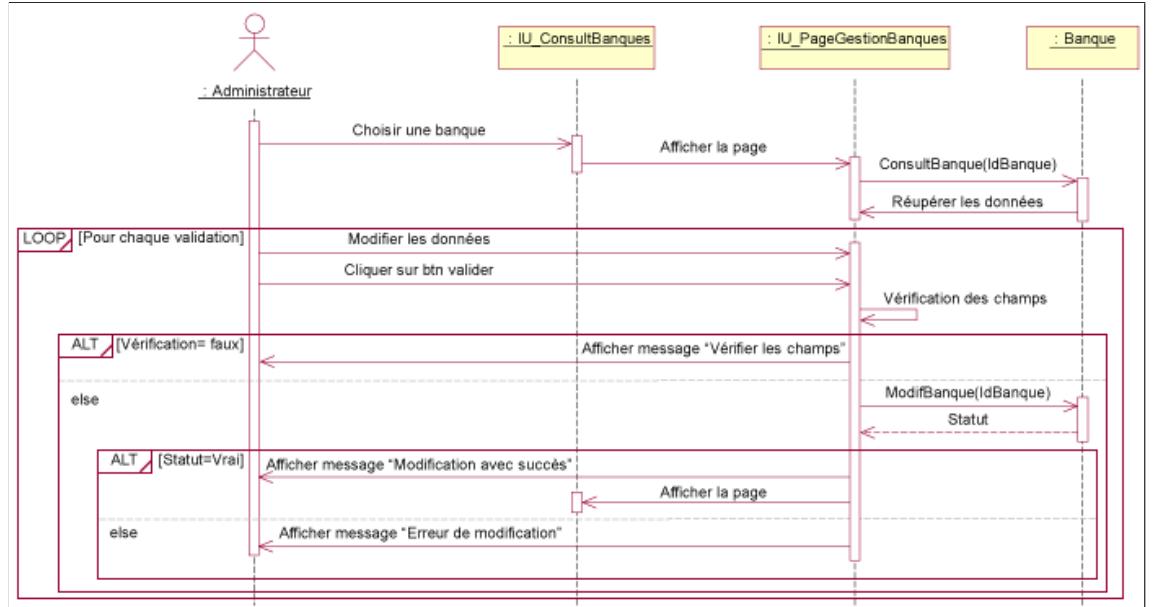


Figure 2.13: Diagramme de séquence du cas d'utilisation
«ModifierBanque»

■ Diagramme de séquence du cas d'utilisation «SupprimerBanque»:

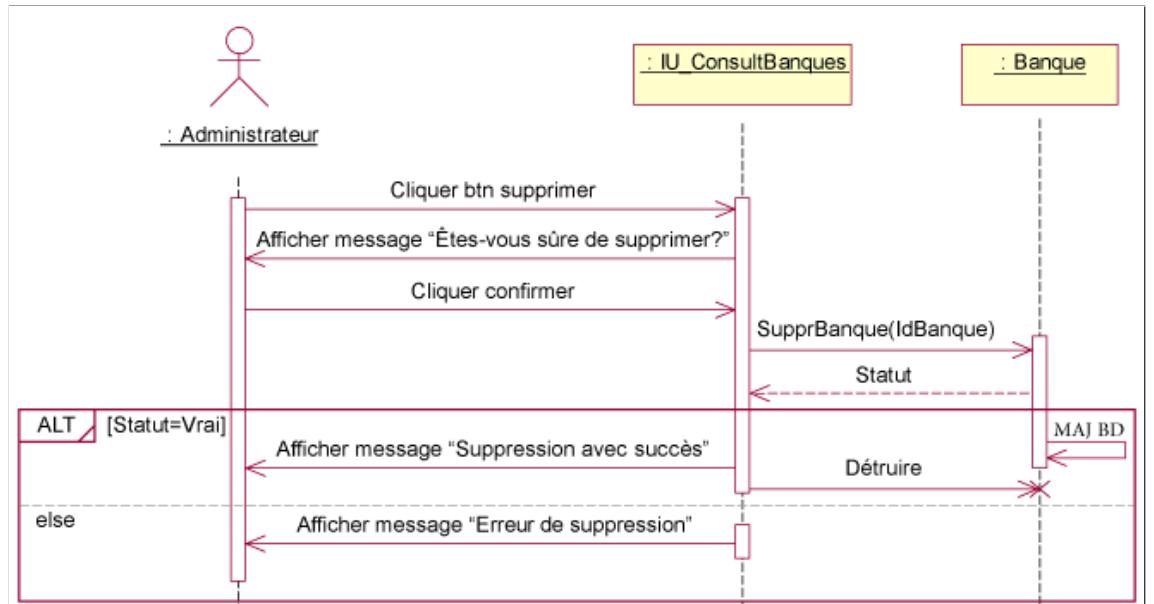


Figure 2.14: Diagramme de séquence du cas d'utilisation
«SupprimerBanque»

C- La conception du cas d'utilisation «GererCategorieTarif»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

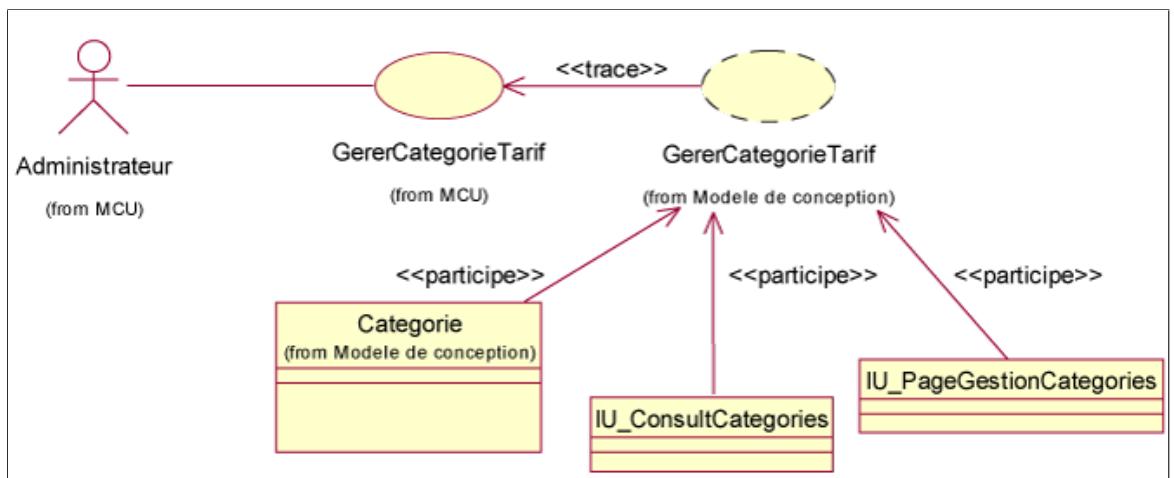


Figure 2.15: Traçabilité MCU/MC du cas d'utilisation
«GererCategorieTarif»

- Diagrammes de séquence :

- Diagramme de séquence du cas d'utilisation «AjouterCategorieTarif»:

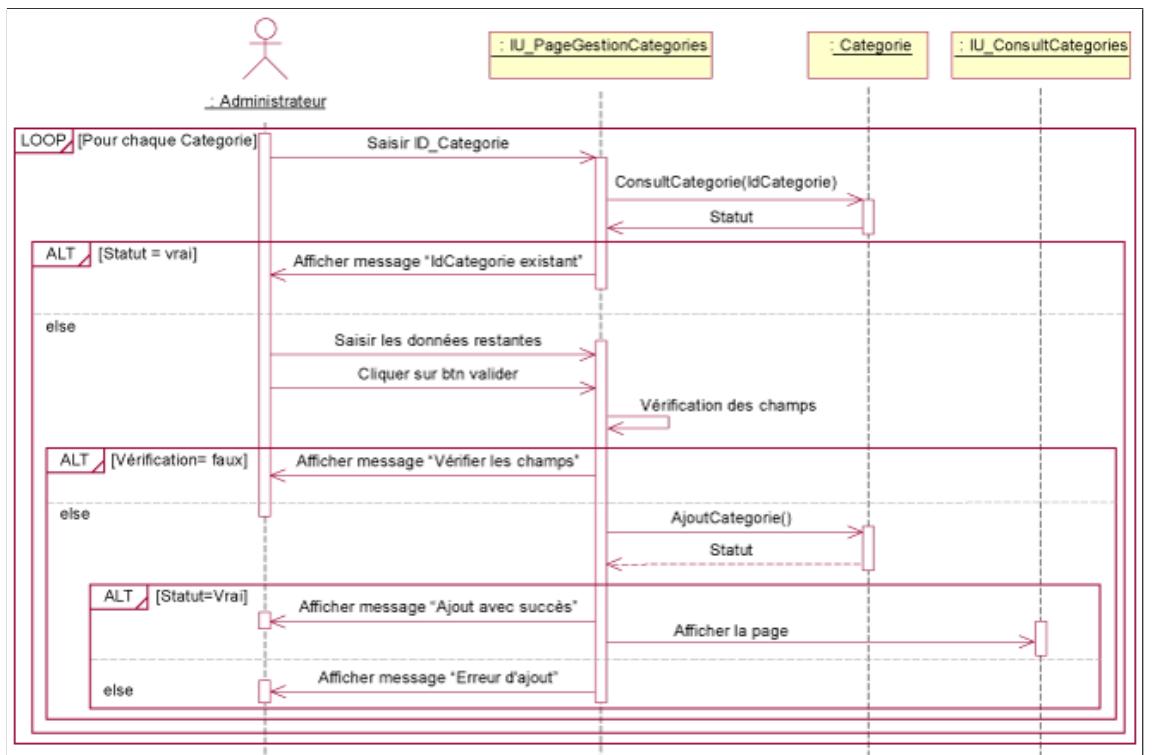


Figure 2.16: Diagramme de séquence du cas d'utilisation
«AjouterCategorieTarif»

■ Diagramme de séquence du cas d'utilisation «ModifierCategorieTarif»:

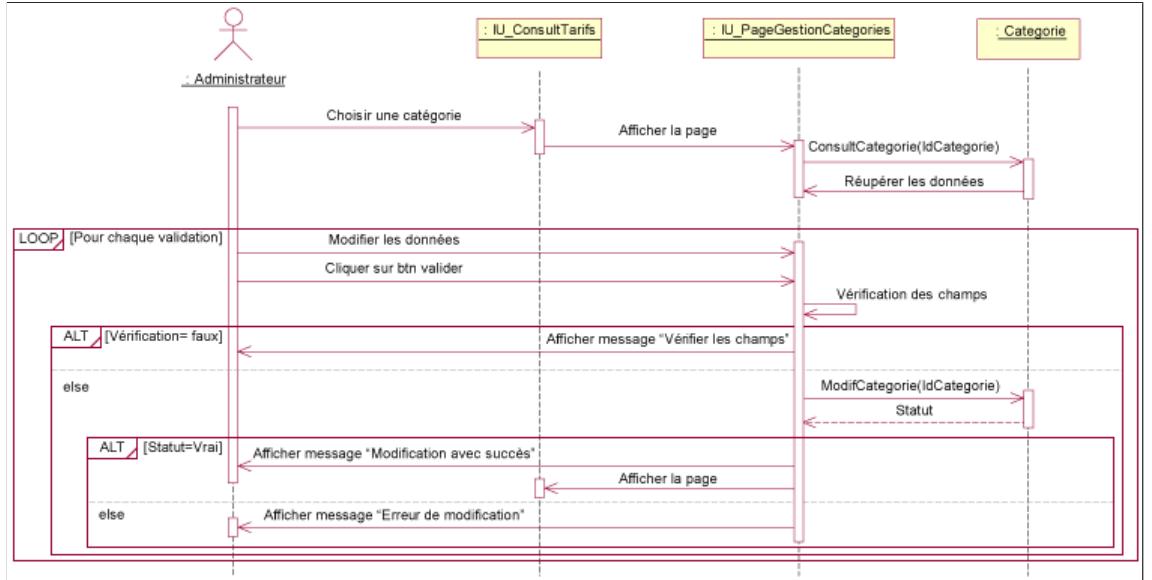


Figure 2.17: Diagramme de séquence du cas d'utilisation
«ModifierCategorieTarif»

■ Diagramme de séquence du cas d'utilisation «SupprimerCategorieTarif»:

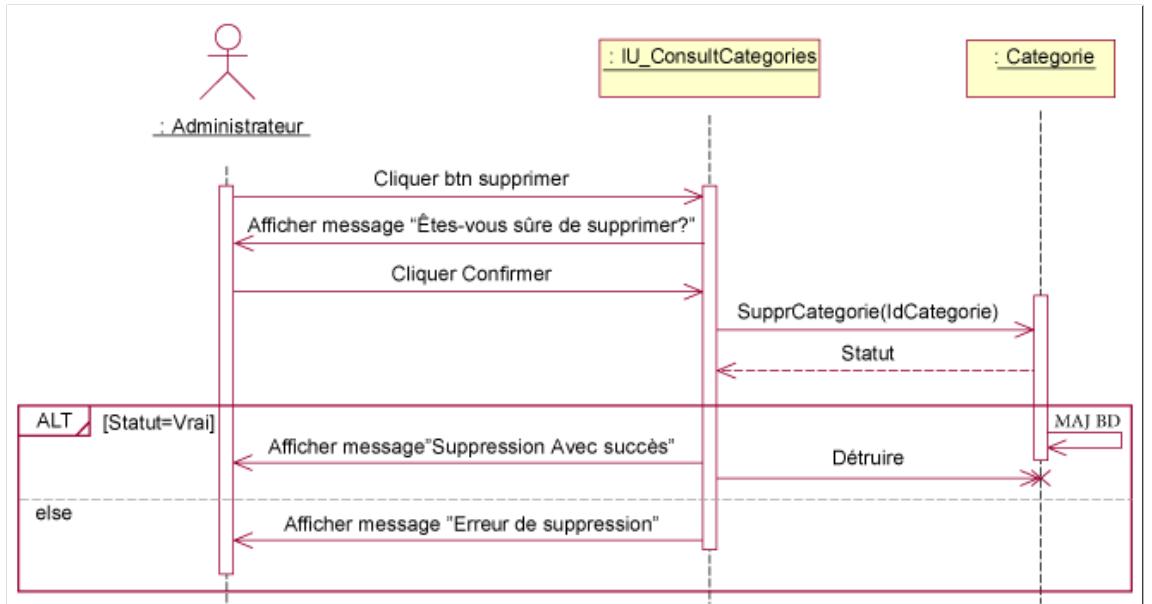


Figure 2.18: Diagramme de séquence du cas d'utilisation «SupprimerCategorieTarif»

D- La conception du cas d'utilisation «GererTarif»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

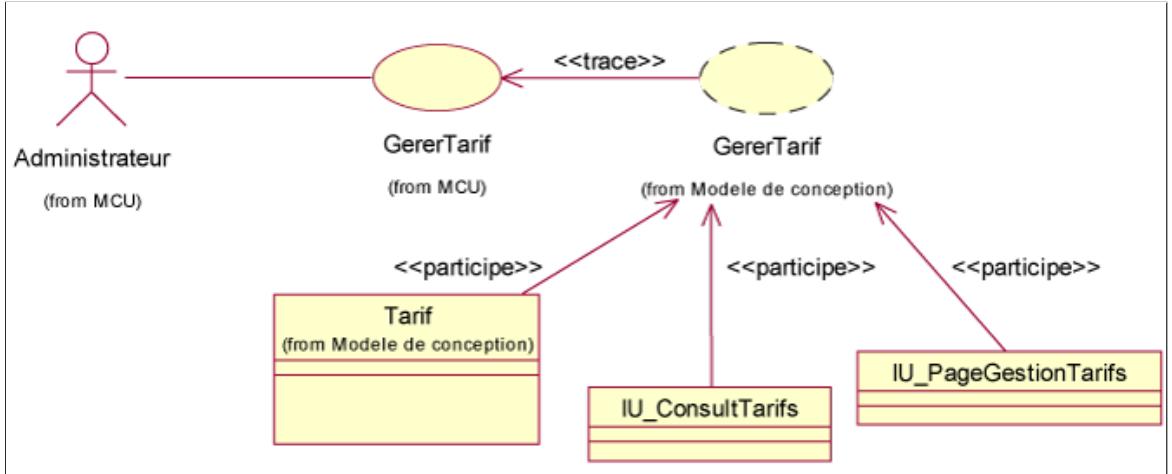


Figure 2.19: Traçabilité MCU/MC du cas d'utilisation «GererTarif»

- Diagrammes de séquence :

- Diagramme de séquence du cas d'utilisation «AjouterTarif»:

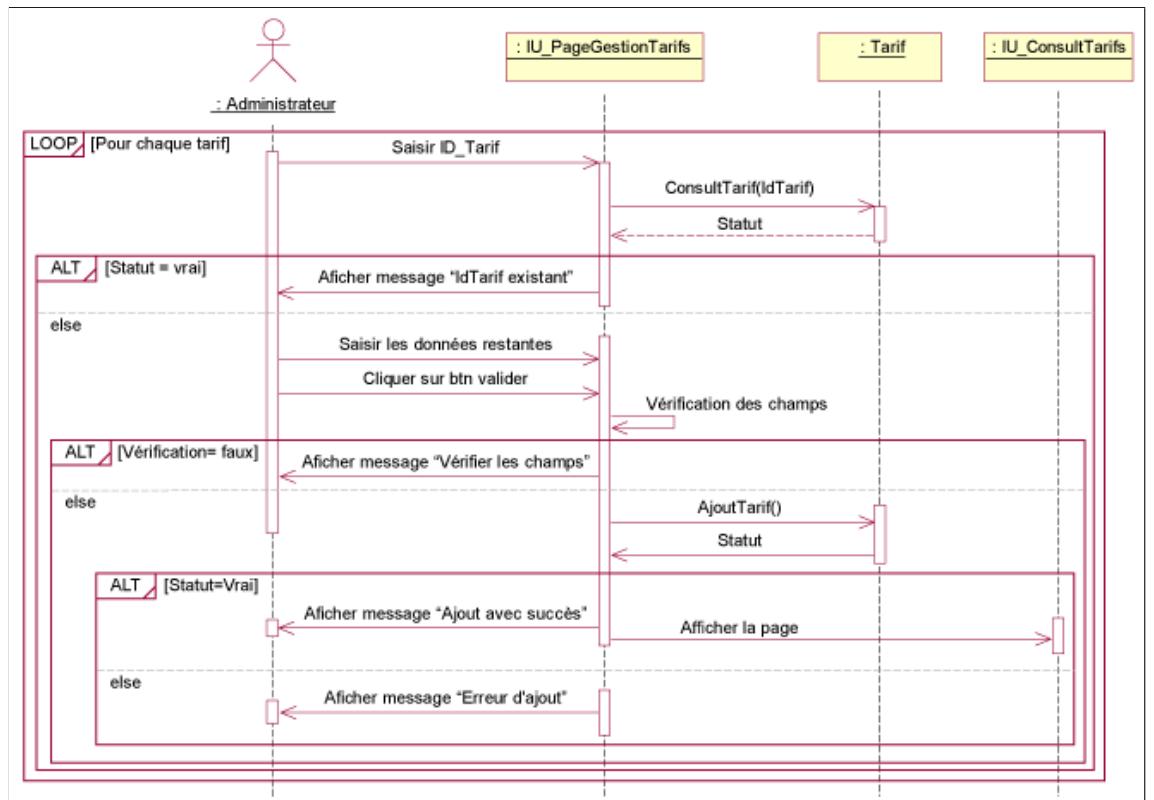


Figure 2.20: Diagramme de séquence du cas d'utilisation «AjouterTarif»

■ Diagramme de séquence du cas d'utilisation «ModifierTarif»:

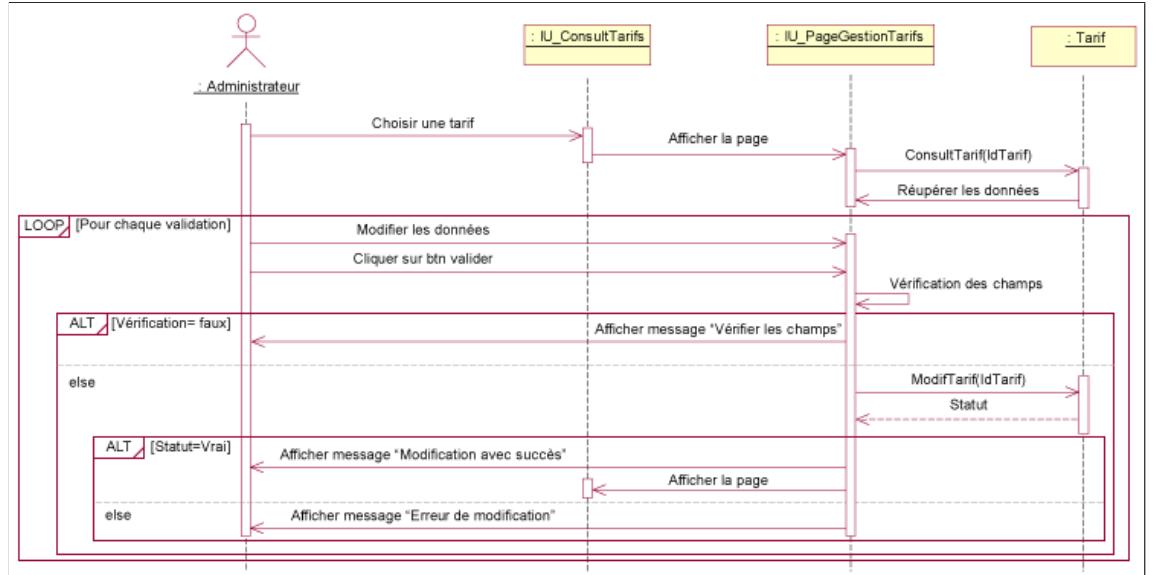


Figure 2.21: Diagramme de séquence du cas d'utilisation «ModifierTarif»

■ Diagramme de séquence du cas d'utilisation «SupprimerTarif»:

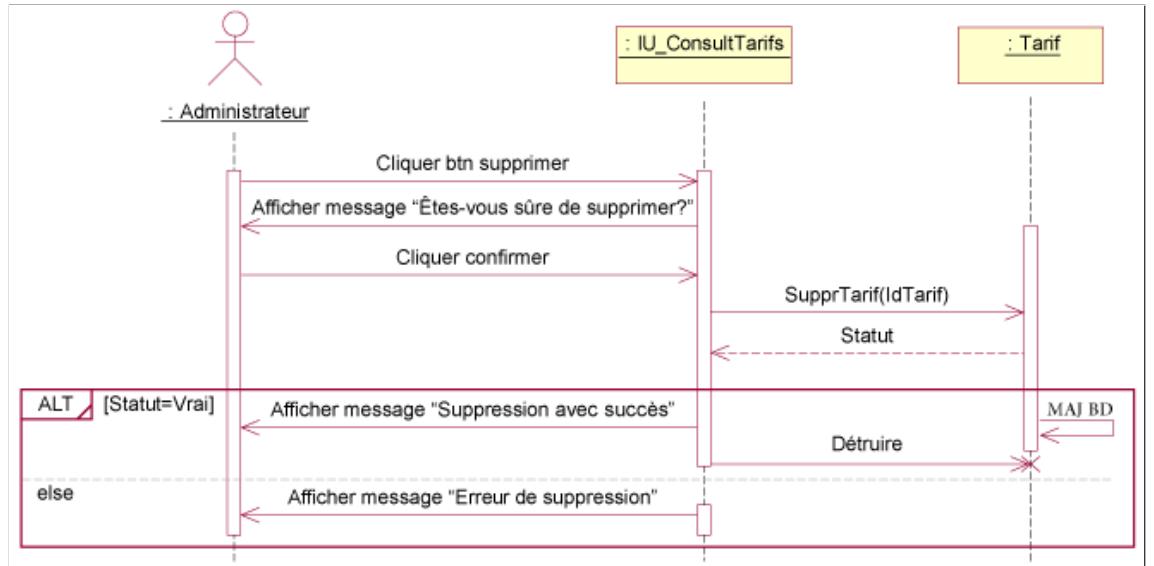


Figure 2.22: Diagramme de séquence du cas d'utilisation «SupprimerTarif»

E- La conception du cas d'utilisation «GererTarification»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

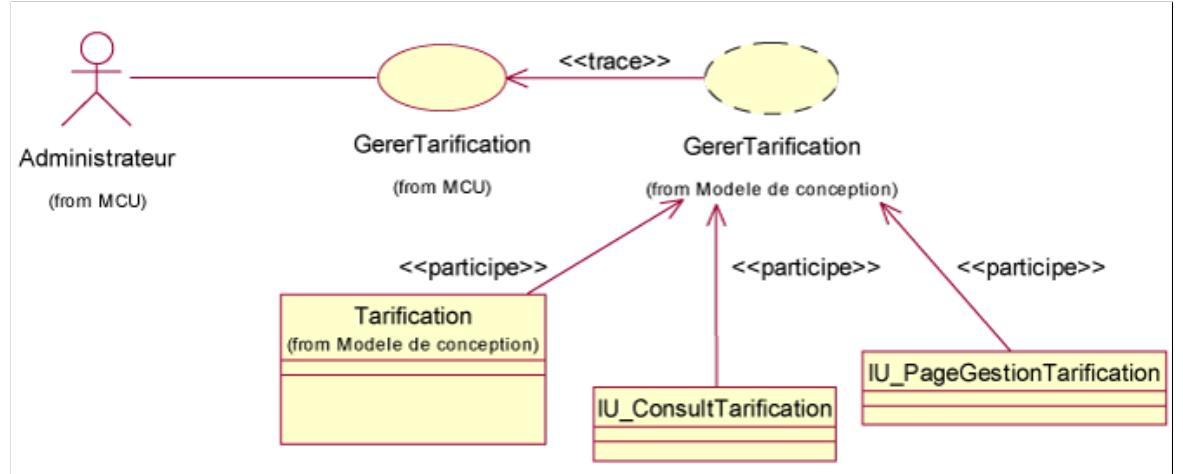


Figure 2.23: Diagramme de séquence du cas d'utilisation «GererTarification»

- Diagrammes de séquence :
 - Diagramme de séquence du cas d'utilisation «AjouterTarification»:

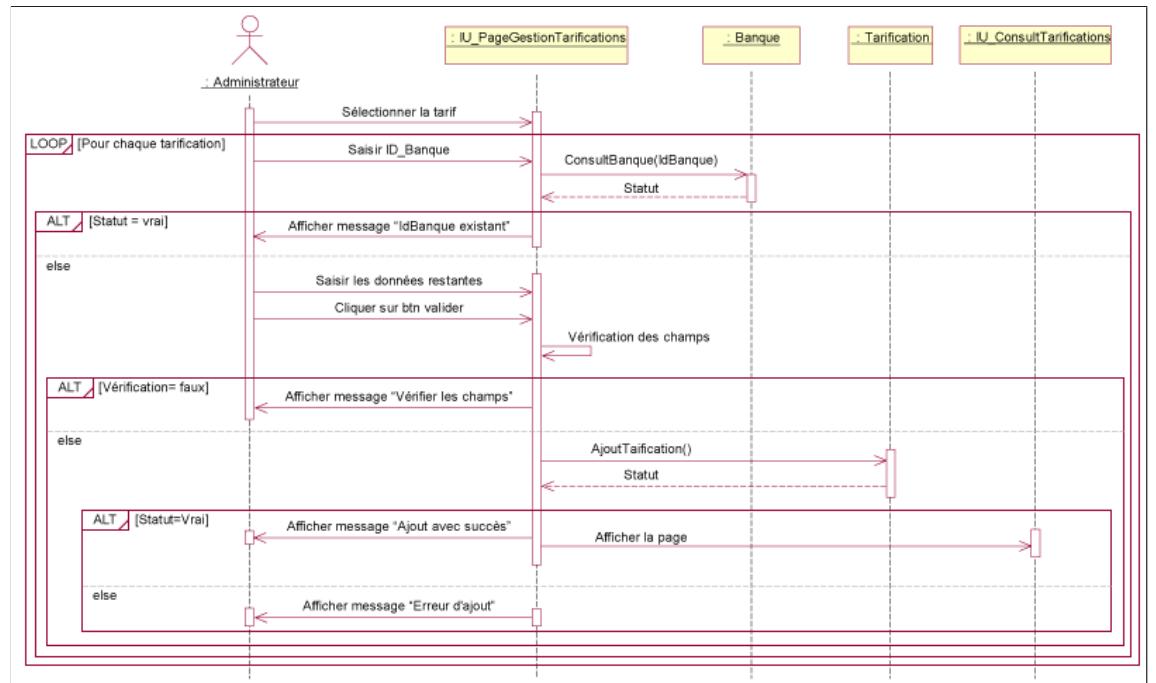


Figure 2.24: Diagramme de séquence du cas d'utilisation «AjouterTarification»

■ Diagramme de séquence du cas d'utilisation «ModifierTarification»:

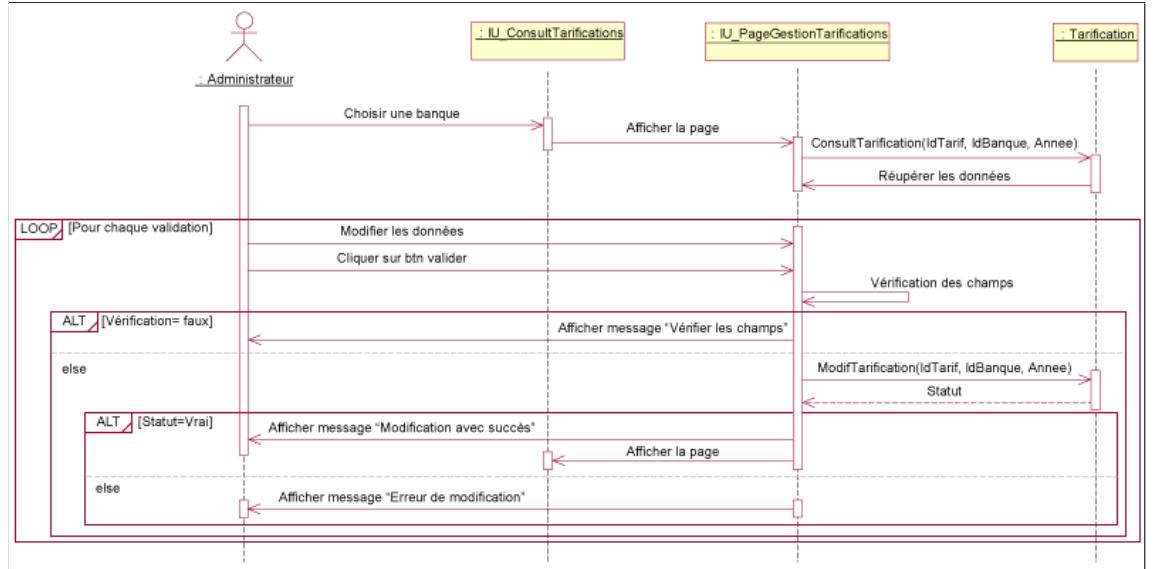


Figure 2.25: Diagramme de séquence du cas d'utilisation «ModifierTarification»

■ Diagramme de séquence du cas d'utilisation «SupprimerTarification»:

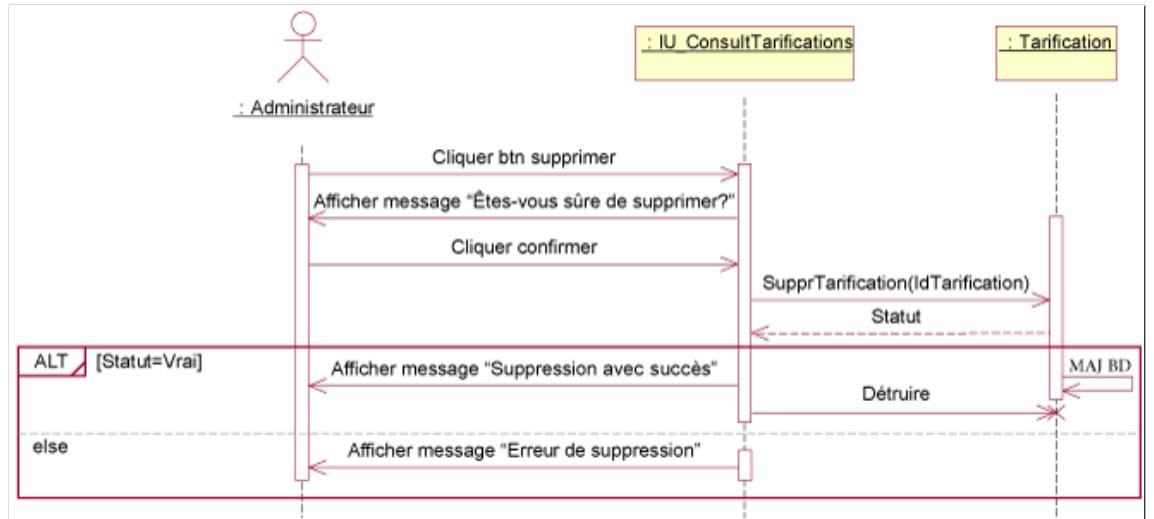


Figure 2.26: Diagramme de séquence du cas d'utilisation «SupprimerTarification»

F- La conception du cas d'utilisation «ComparerTarification»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

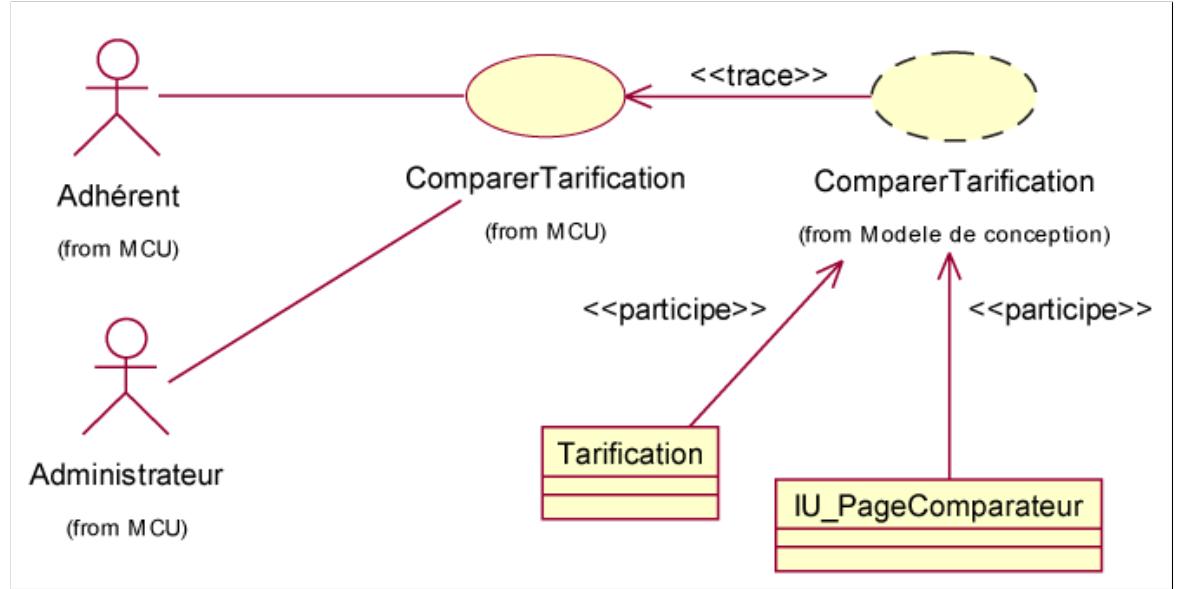


Figure 2.27: Traçabilité MCU/MC du cas d'utilisation «ComparerTarification»

- Diagramme de séquence :

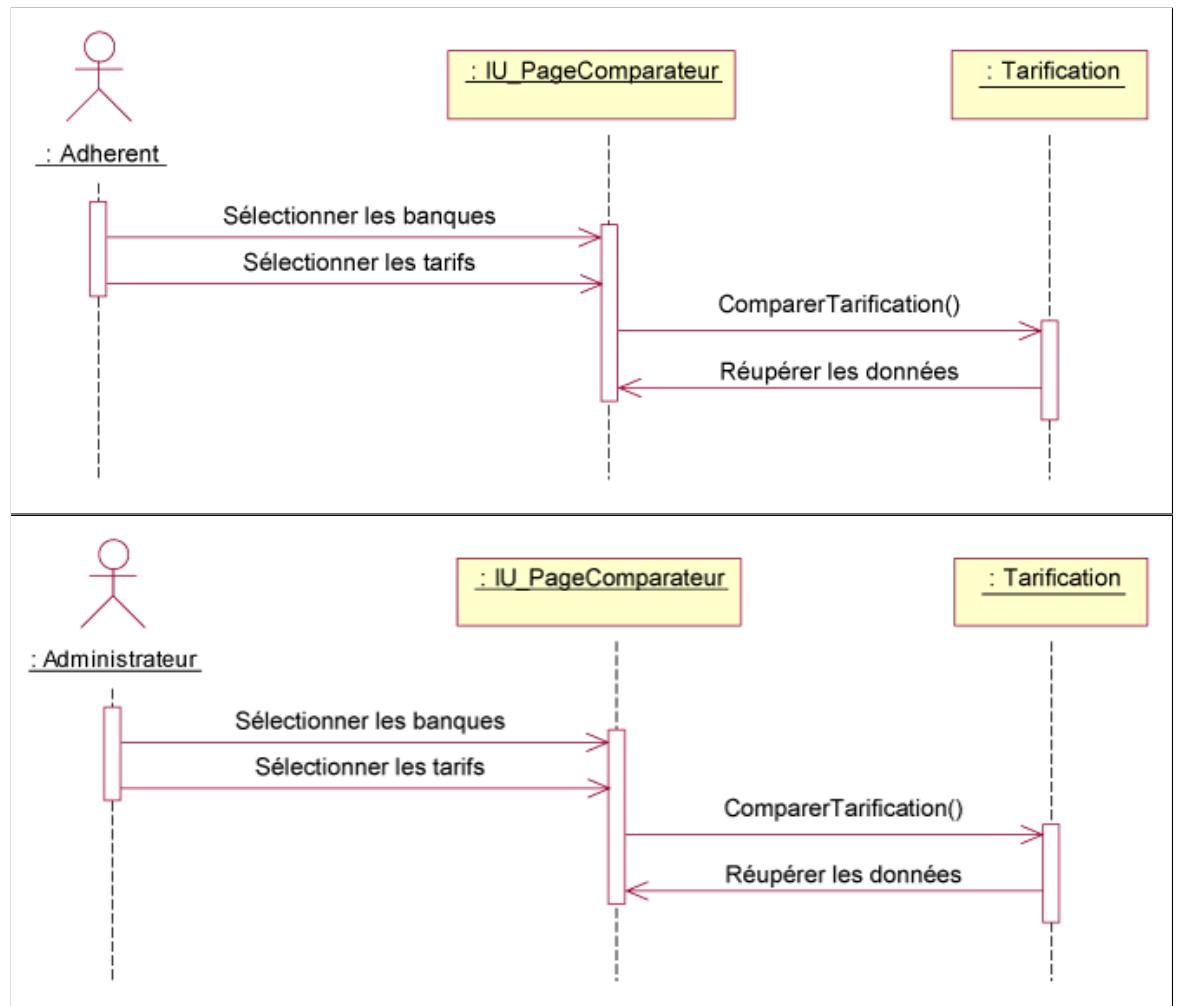


Figure 2.28: Diagramme de séquence du cas d'utilisation «ComparerTarification»

2.3 Implémentation

Au cours de cette activité, nous allons établir les diagrammes de composants pour les cas d'utilisation prioritaires avec leur traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI). En effet, ces diagrammes décrivent le système modélisé sous forme de composants réutilisables et mettent en évidence leurs relations de dépendance.

2.3.1 Implémentation du cas d'utilisation «S_authentifier»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

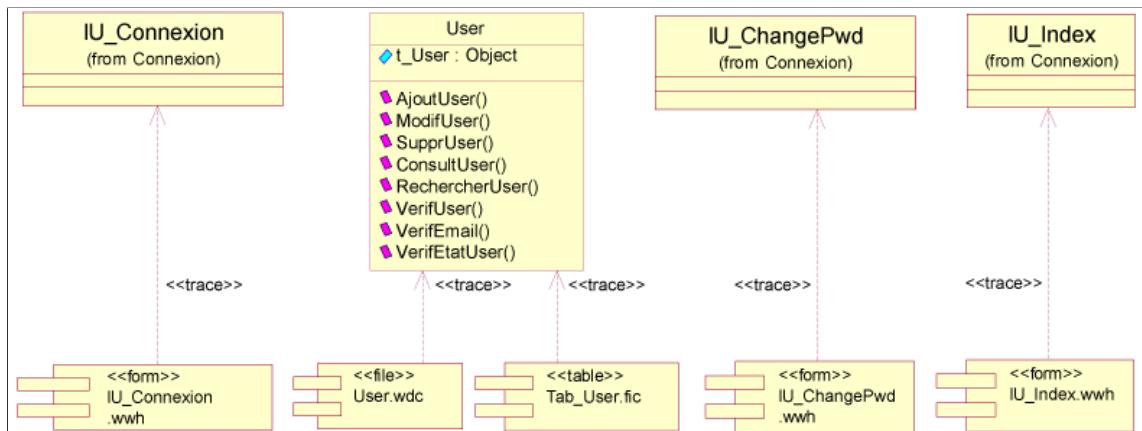


Figure 2.29: Traçabilité MC/MI du cas d'utilisation «Sauthentifier»

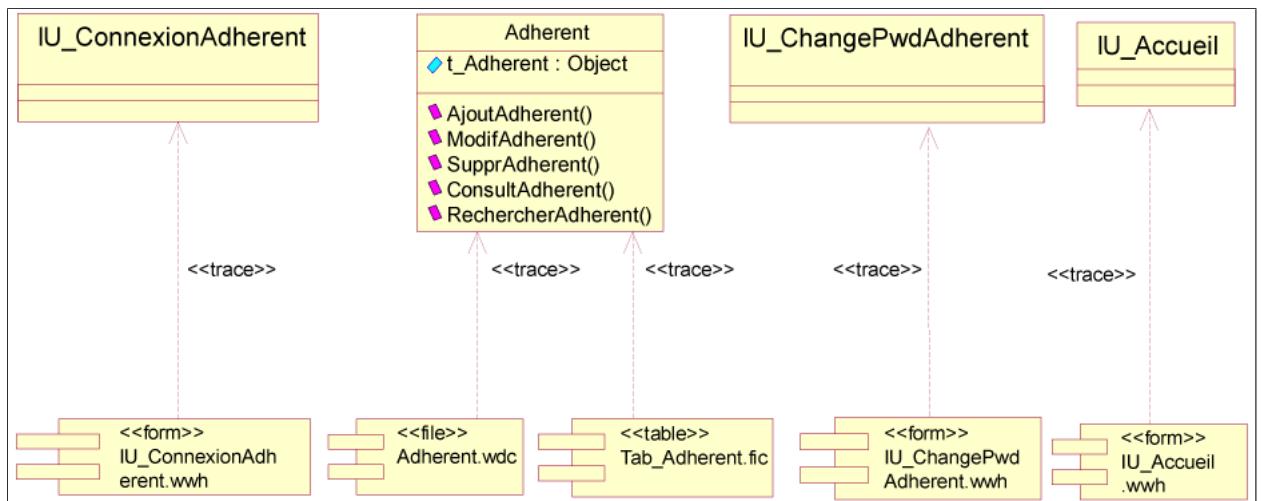


Figure 2.30: Traçabilité MC/MI du cas d'utilisation «Sauthentifier» pour l'adhérent

B- Diagramme de composant

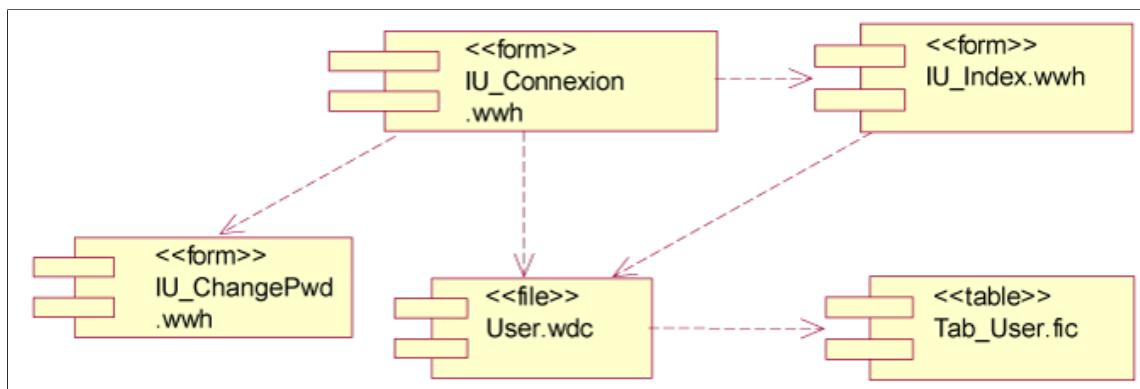


Figure 2.31: Diagramme de composant du cas d'utilisation «Sauthentifier»

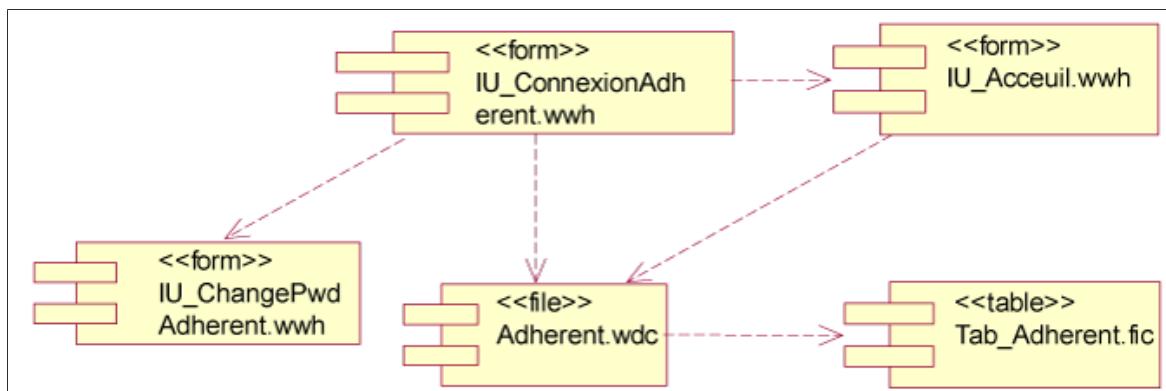


Figure 2.32: Diagramme de composant du cas d'utilisation «Sauthentifier»
pour l'adhérent

2.3.2 Implémentation du cas d'utilisation «GererBanque»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

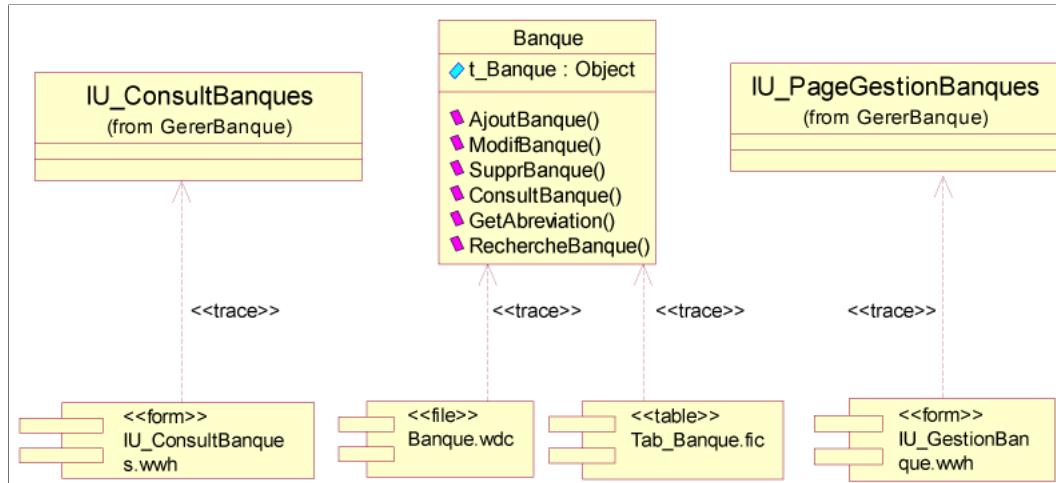


Figure 2.33: Traçabilité MC/MI du cas d'utilisation «GererBanque»

B- Diagramme de composant

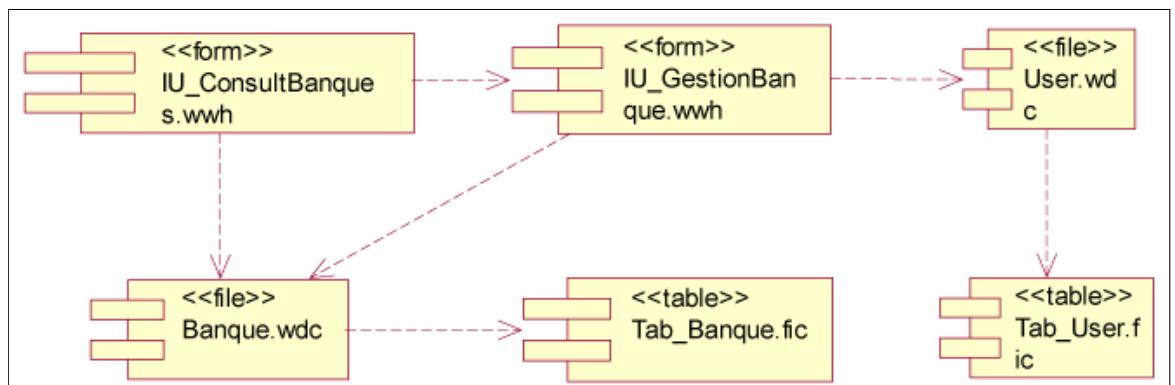


Figure 2.34: Diagramme de composant du cas d'utilisation
«GererBanque»

2.3.3 Implémentation du cas d'utilisation «GererCategorieTarif»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

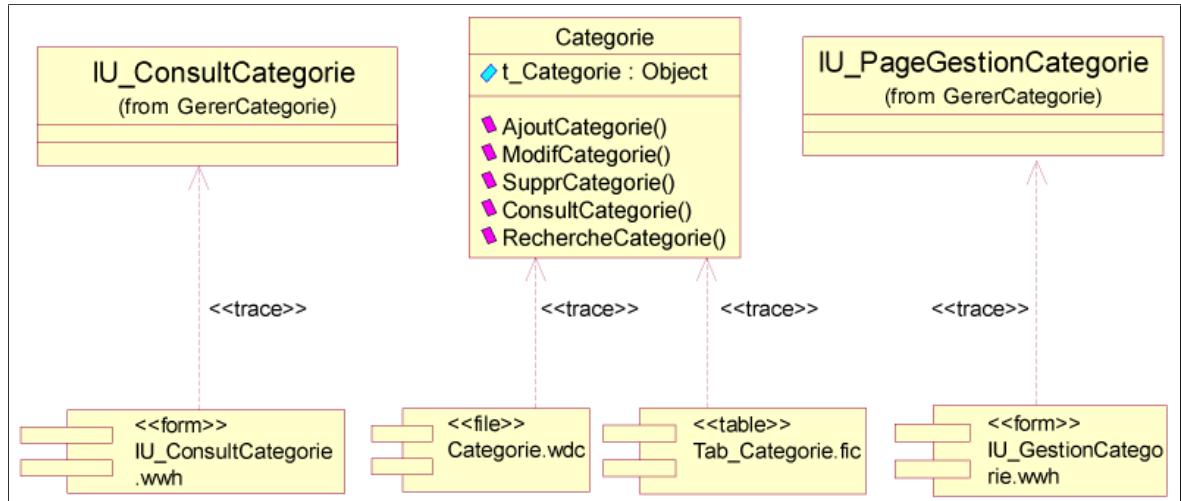


Figure 2.35: Traçabilité MC/MI du cas d'utilisation
«GererCategorieTarif»

B- Diagramme de composant

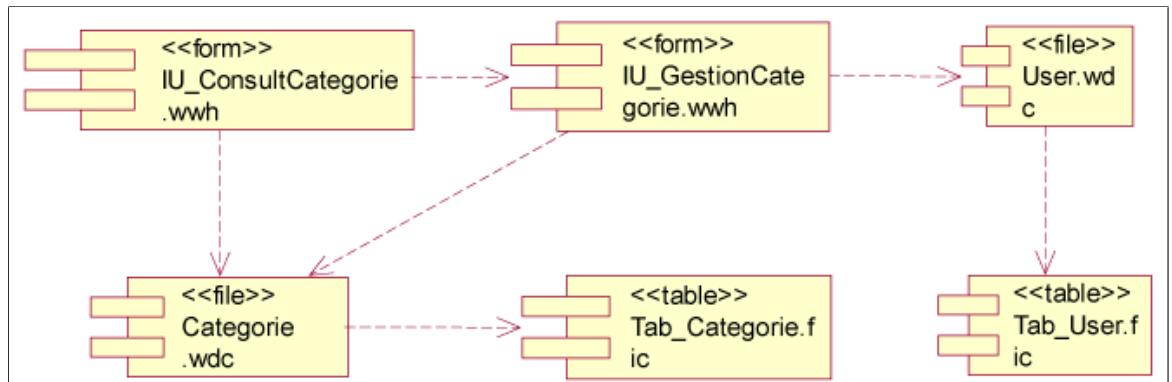


Figure 2.36: Diagramme de composant du cas d'utilisation
«GererCategorieTarif»

2.3.4 Implémentation du cas d'utilisation «GererTarif»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

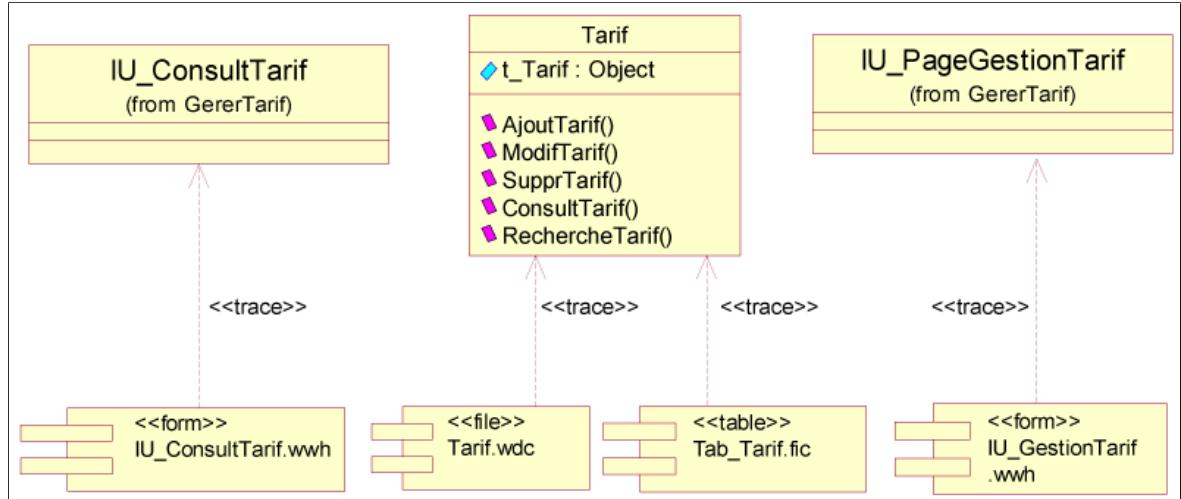


Figure 2.37: Traçabilité MC/MI du cas d'utilisation «GererTarif»

B- Diagramme de composant

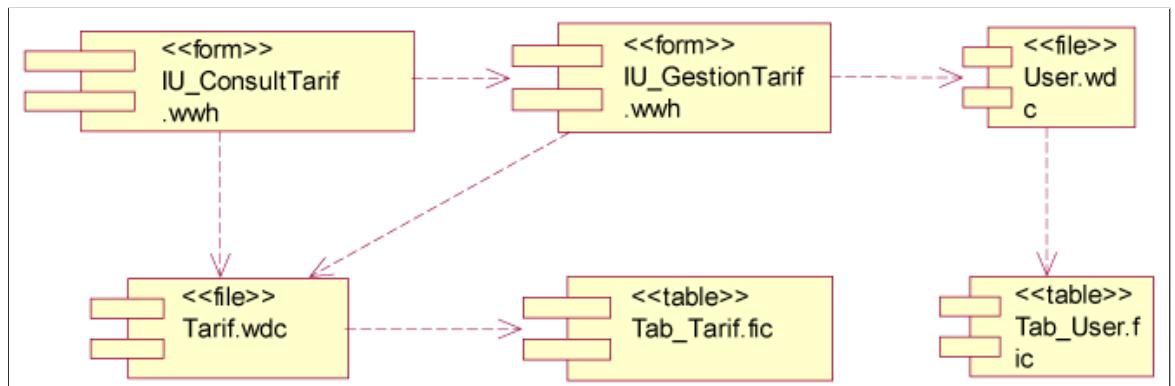


Figure 2.38: Diagramme de composant du cas d'utilisation «GererTarif»

2.3.5 Implémentation du cas d'utilisation «GererTarification»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

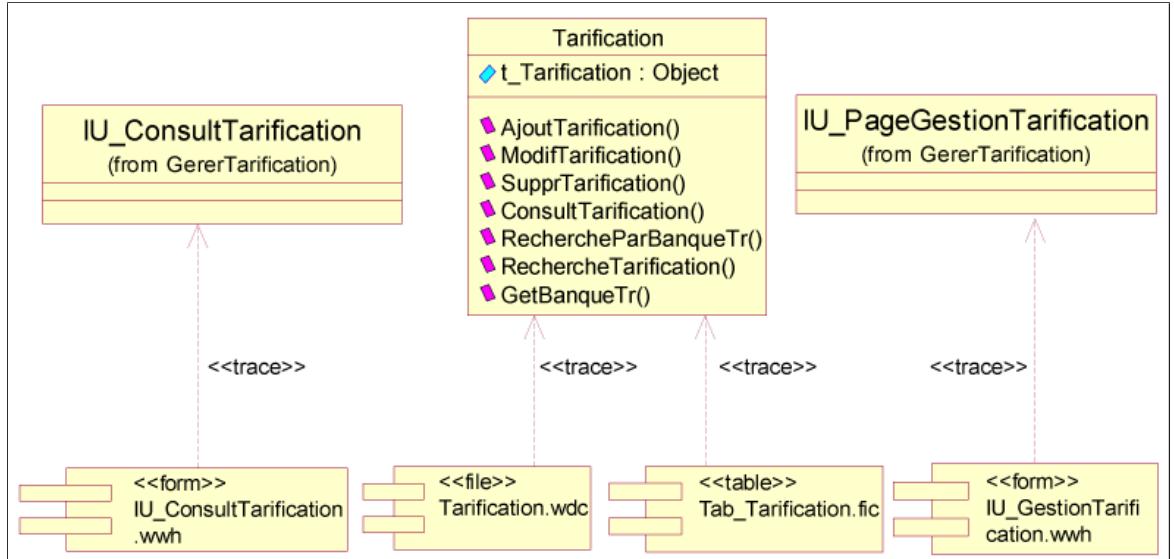


Figure 2.39: Traçabilité MC/MI du cas d'utilisation «GererTarification»

B- Diagramme de composant

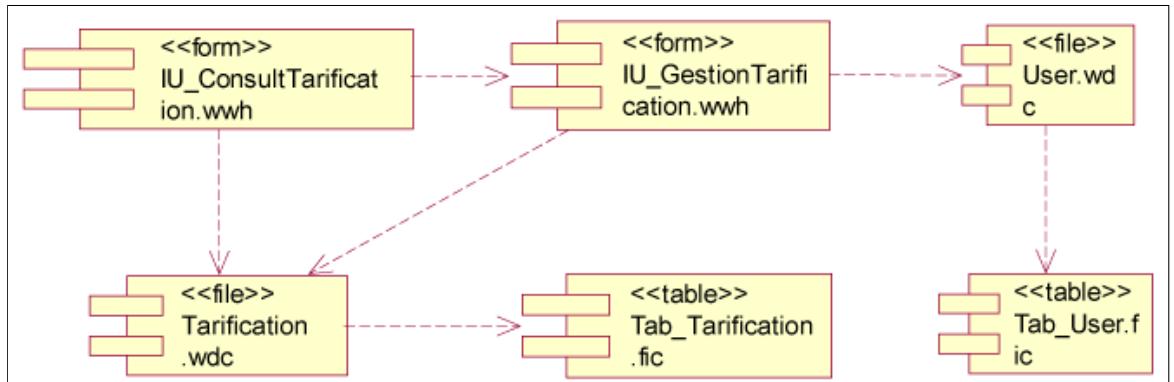


Figure 2.40: Diagramme de composant du cas d'utilisation
«GererTarification»

2.3.6 Implémentation du cas d'utilisation «ComparerTarification»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

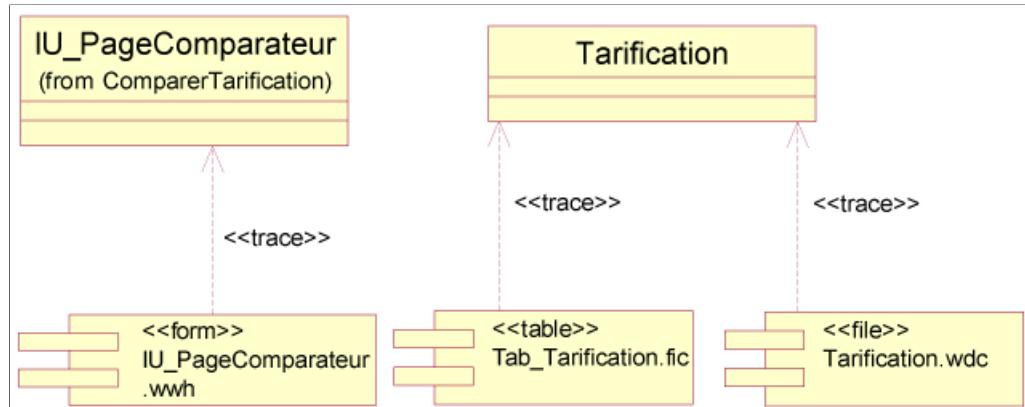


Figure 2.41: Traçabilité MC/MI du cas d'utilisation «ComparerTarification»

B- Diagramme de composant

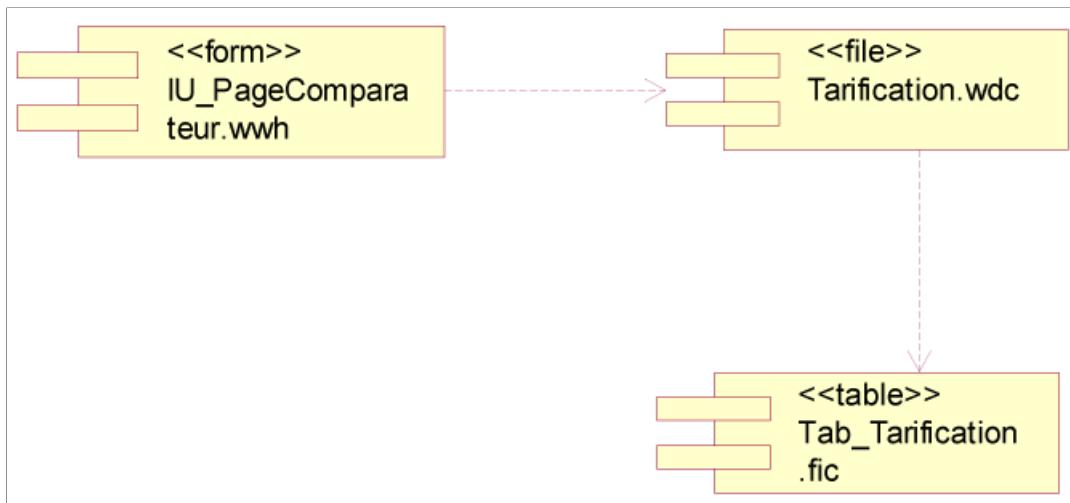


Figure 2.42: Diagramme de composant du cas d'utilisation «ComparerTarification»

Conclusion

Dans ce chapitre nous avons identifié les fonctionnalités du système lors de la capture des besoins. Ensuite, nous avons conçu et implémenté les cas d'utilisations de priorité une. Dans le chapitre suivant, notre objectif est de traiter les autres cas d'utilisation de priorité deux.

3 La phase d'élaboration

Introduction

La phase d'élaboration est la deuxième phase du Processus Unifié, au cours de laquelle nous visons à compléter l'étude des besoins fonctionnels durant la phase de création. L'objectif maintenant est d'approfondir notre compréhension par la conception et l'implémentation des cas d'utilisation de priorité deux.

3.1 Capture des besoins

3.1.1 Raffinement des cas d'utilisation de priorité 2

A- Raffinement du cas d'utilisation «Gérer les agences» :

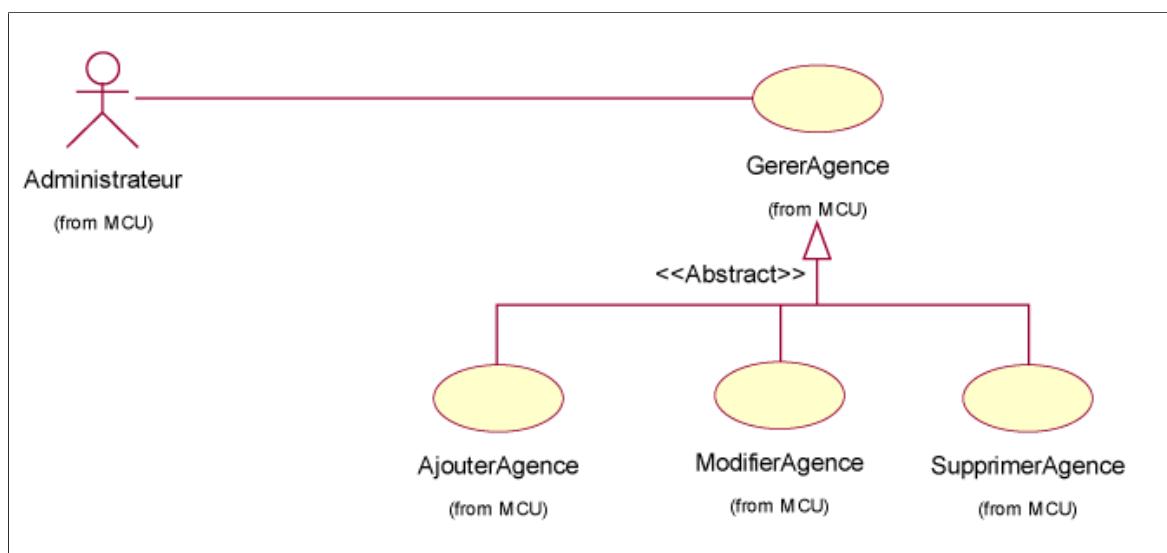


Figure 3.1: Raffinement du cas d'utilisation «GererAgence»

- Raffinement du cas d'utilisation «Ajouter une agence» :

niceblueCas d'utilisation	AjouterAgence
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Agence ajoutée
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Nouvelle agence» de l'interface de consultation. - L'administrateur saisit l'identifiant de l'agence et et de la banque correspondante de la nouvelle agence. - Le système vérifie l'inexistence de la combinaison des deux identifiants dans la base. - L'administrateur continue à saisir les informations restantes de l'agence. - Il clique sur le bouton «valider». - Le système enregistre les données saisies. - Il affiche un message indiquant l'ajout avec succès de la nouvelle agence.
niceblueExceptions	<ul style="list-style-type: none"> - Si un champ obligatoire est vide ou bien ne respecte pas la format de l'écriture (pour le téléphone par exemple), le système affiche un message d'erreur. - Si la combinaison des deux identifiants est existante dans la base, le système affiche un message d'erreur.

Table 3.1: Spécification du cas d'utilisation «Ajouter une agence»

- Raffinement du cas d'utilisation «Modifier une agence» :

niceblueCas d'utilisation	ModifierAgence
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Informations modifiées d'une agence
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit l'agence à modifier depuis l'interface de consultation. - Il clique sur l'icône de la modification. - Le système affiche la page de la modification avec les informations de l'agence choisie. - L'administrateur modifie les informations désirées. - Il clique sur le bouton «valider». - Le système met à jour l'agence avec les nouvelles informations. - Il affiche un message indiquant la modification avec succès de l'agence.
niceblueExceptions	Si un champ obligatoire manque la saisie ou bien ne respecte pas la format de l'écriture (pour le téléphone par exemple), le système affiche un message d'erreur.

Table 3.2: Spécification du cas d'utilisation «Modifier une agence»

- Raffinement du cas d'utilisation «Supprimer une agence» :

niceblueCas d'utilisation	SupprimerAgence
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Agence supprimée
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit la agence à supprimer depuis l'interface de consultation. - Il clique sur l'icône de suppression. <p>Le système affiche un message d'avertissement indiquant l'accord de suppression.</p> <ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Oui». - Le système affiche un message indiquant la suppression avec succès de la agence.
niceblueExceptions	Si l'administrateur clique sur le bouton «Non», l'agence ne sera pas supprimée.

Table 3.3: Spécification du cas d'utilisation «Supprimer une agence»

B- Raffinement du cas d'utilisation «Gérer les DAB» :

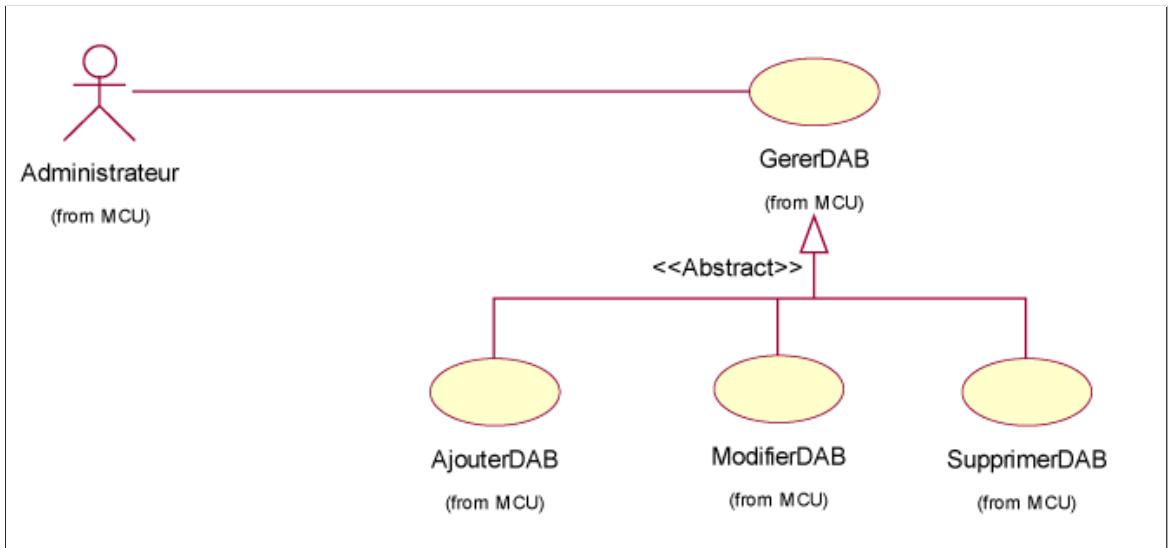


Figure 3.2: Raffinement du cas d'utilisation «GererDAB»

- Raffinement du cas d'utilisation «AjouterDAB»

niceblueCas d'utilisation	AjouterDAB
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	DAB ajouté
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Nouveau DAB» de l'interface de consultation. - L'administrateur saisit l'identifiant de la banque - Le système vérifie l'existence de cet identifiant dans la base. - L'administrateur saisit le code ville du DAB - Le système vérifie l'existence de ce code dans la base. - L'administrateur continue à saisir les informations restantes du DAB. - Il clique sur le bouton «valider». - Le système enregistre les données saisies. - Il affiche un message indiquant l'ajout avec succès du nouveau DAB.
niceblueExceptions	<ul style="list-style-type: none"> - Si un champ obligatoire est vide ou bien ne respecte pas la format de l'écriture (pour le code ville par exemple), le système affiche un message d'erreur. - Si l'identifiant et/ou le code n'existe pas dans la base, le système affiche un message d'erreur.

Table 3.4: Spécification du cas d'utilisation «AjouterDAB»

- Raffinement du cas d'utilisation «ModifierDAB»

niceblueCas d'utilisation	ModifierDAB
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Informations modifiées d'un DAB
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit le DAB à modifier depuis l'interface de consultation. - Il clique sur l'icône de la modification. - Le système affiche la page de la modification avec les informations du DAB choisi. - L'administrateur modifie les informations désirées. - Il clique sur le bouton «Envoyer». - Le système met à jour le DAB avec les nouvelles informations. - Il affiche un message indiquant la modification avec succès du DAB.
niceblueExceptions	Si un champ obligatoire manque la saisie ou bien ne respecte pas la format de l'écriture (pour le code ville par exemple), le système affiche un message d'erreur.

Table 3.5: Spécification du cas d'utilisation «ModifierDAB»

- Raffinement du cas d'utilisation «SupprimerDAB»

niceblueCas d'utilisation	SupprimerDAB
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	DAB supprimé
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit le DAB à supprimer depuis l'interface de consultation. - Il clique sur l'icône de suppression. <p>Le système affiche un message d'avertissement indiquant l'accord de suppression.</p> <ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Oui». - Le système affiche un message indiquant la suppression avec succès du DAB.
niceblueExceptions	Si l'administrateur clique sur le bouton «Non», le DAB ne sera pas supprimé.

Table 3.6: Spécification du cas d'utilisation «SupprimerDAB»

C- Raffinement du cas d'utilisation «Gérer les gouvernorats» :

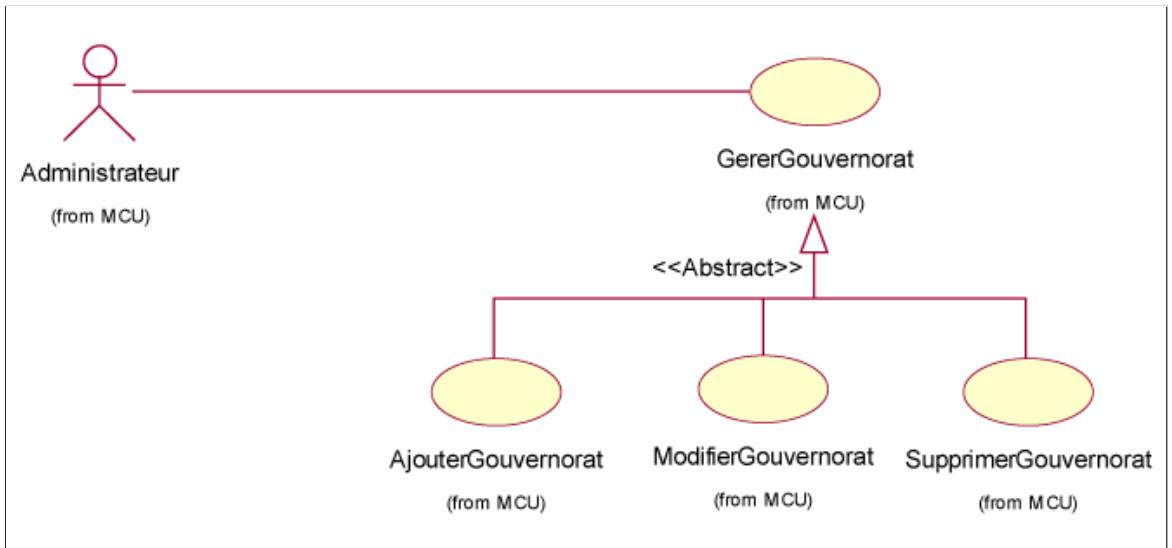


Figure 3.3: Raffinement du cas d'utilisation «GererGouvernorat»

- Raffinement du cas d'utilisation «AjouterGouvernorat»

niceblueCas d'utilisation	AjouterGouvernorat
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Gouvernorat ajouté

Scénario principal	<ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Nouveau gouvernorat» de l'interface de consultation. - L'administrateur saisit l'identifiant unique du nouveau gouvernorat. - Le système vérifie l'inexistence de cet identifiant dans la base. - L'administrateur continue à saisir les informations restantes du gouvernorat. - Il clique sur le bouton «valider». - Le système enregistre les données saisies. - Il affiche un message indiquant l'ajout avec succès du nouveau gouvernorat.
Exceptions	<ul style="list-style-type: none"> - Si un champ obligatoire manque la saisie, le système affiche un message d'erreur. - Si l'identifiant saisi est existant dans la base, le système affiche un message d'erreur.

Table 3.7: Spécification du cas d'utilisation «AjouterGouvernorat»

- Raffinement du cas d'utilisation «ModifierGouvernorat»

niceblueCas d'utilisation	ModifierGouvernorat
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Informations modifiées d'un gouvernorat
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit le gouvernorat à modifier depuis l'interface de consultation. - Il clique sur l'icône de la modification. - Le système affiche la page de la modification avec les informations du gouvernorat choisi. - L'administrateur modifie les informations désirées. - Il clique sur le bouton «valider». - Le système met à jour le gouvernorat avec les nouvelles informations. - Il affiche un message indiquant la modification avec succès du gouvernorat.
niceblueExceptions	Si un champ obligatoire manque la saisie, le système affiche un message d'erreur.

Table 3.8: Spécification du cas d'utilisation «ModifierGouvernorat»

- Raffinement du cas d'utilisation «SupprimerGouvernorat»

niceblueCas d'utilisation	SupprimerGouvernorat
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Gouvernorat supprimé

niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit le gouvernorat à supprimer depuis l'interface de consultation. - Il clique sur l'icône de suppression. <p>Le système affiche un message d'avertissement indiquant l'accord de suppression.</p> <ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Oui». - Le système affiche un message indiquant la suppression avec succès du gouvernorat.
niceblueExceptions	Si l'administrateur clique sur le bouton «Non», le gouvernorat ne sera pas supprimé.

Table 3.9: Spécification du cas d'utilisation «SupprimerGouvernorat»

D- Raffinement du cas d'utilisation «Gérer les delegations» :

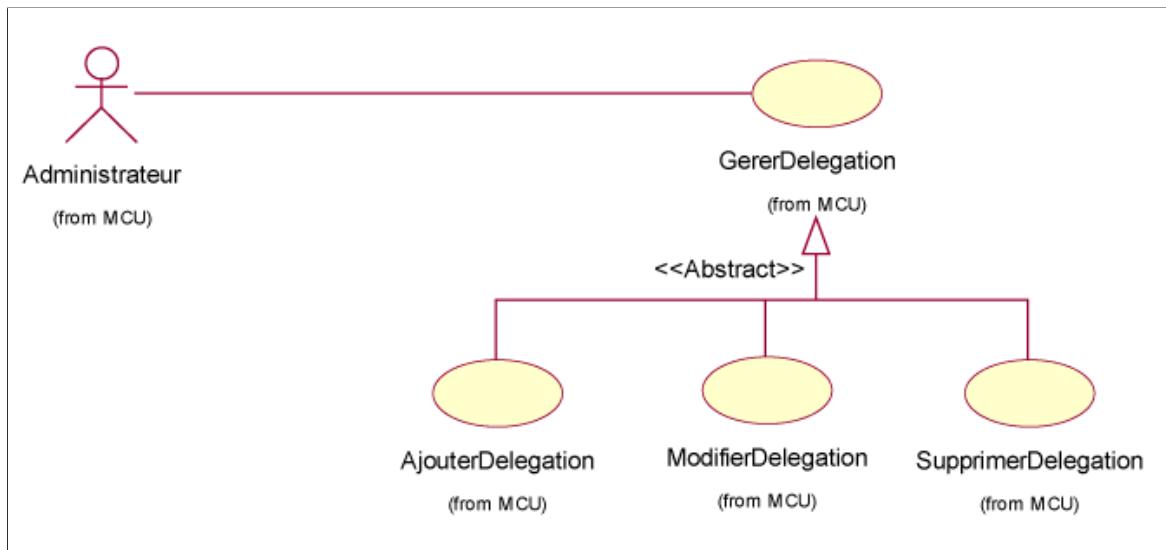


Figure 3.4: Raffinement du cas d'utilisation «GererDelegation»

- Raffinement du cas d'utilisation «AjouterDelegation»

niceblueCas d'utilisation	AjouterDelegation
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Délégation ajoutée

niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Nouvelle délégation» de l'interface de consultation. - L'administrateur saisit le code postal unique de la nouvelle délégation. - Le système vérifie l'existence de l'identifiant du gouvernorat correspondant à la ville à partir des deux premiers chiffres du code postal. - Le système vérifie l'inexistence de la totalité du code postal dans la base. - L'administrateur continue à saisir les informations restantes de la délégation. - Il clique sur le bouton «valider». - Le système enregistre les données saisies. - Il affiche un message indiquant l'ajout avec succès de la nouvelle délégation.
niceblueExceptions	<ul style="list-style-type: none"> - Si un champ obligatoire manque la saisie, le système affiche un message d'erreur. - Si le l'identifiant du gouvernorat formé par les deux premiers chiffres du code postal saisi est inexistant dans la base, le système affiche un message d'erreur. - Si le code postal saisi est existant dans la base, le système affiche un message d'erreur.

Table 3.10: Spécification du cas d'utilisation «AjouterDelegation»

• Raffinement du cas d'utilisation «ModifierDelegation»

niceblueCas d'utilisation	ModifierDelegation
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Informations modifiées d'une délégation
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit la délégation à modifier depuis l'interface de consultation. - Il clique sur l'icône de la modification. - Le système affiche la page de la modification avec les informations de la délégation choisie. - L'administrateur modifie les informations désirées. - Il clique sur le bouton «valider». - Le système met à jour la délégation avec les nouvelles informations. - Il affiche un message indiquant la modification avec succès de la délégation.
niceblueExceptions	Si un champ obligatoire manque la saisie, le système affiche un message d'erreur.

Table 3.11: Spécification du cas d'utilisation «ModifierDelegation»

- Raffinement du cas d'utilisation «SupprimerDelegation»

niceblueCas d'utilisation	SupprimerDelegation
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Délégation supprimée
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit la délégation à supprimer depuis l'interface de consultation. - Il clique sur l'icône de suppression. <p>Le système affiche un message d'avertissement indiquant l'accord de suppression.</p> <ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Oui». - Le système affiche un message indiquant la suppression avec succès de la délégation.
niceblueExceptions	Si l'administrateur clique sur le bouton «Non», la délégation ne sera pas supprimée.

Table 3.12: Spécification du cas d'utilisation «SupprimerDelegation»

E- Raffinement du cas d'utilisation «Gérer les villes» :

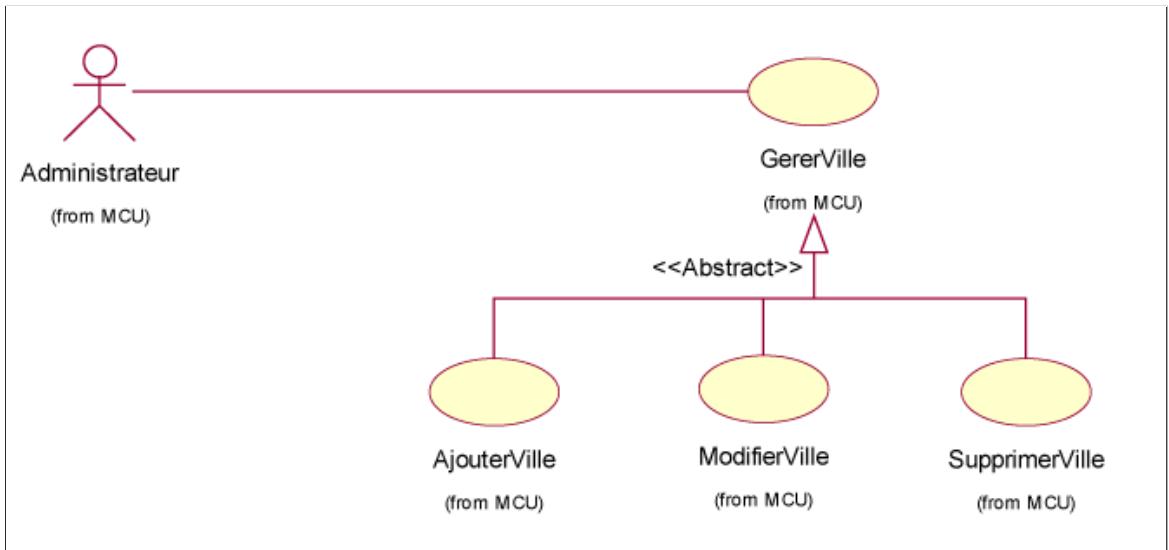


Figure 3.5: Raffinement du cas d'utilisation «GererVille»

- Raffinement du cas d'utilisation «AjouterVille»

niceblueCas d'utilisation	AjouterVille
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Ville ajoutée

niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Nouvelle ville» de l'interface de consultation. - Il saisit le code postal de la nouvelle ville. - Le système vérifie l'existence de l'identifiant du gouvernorat correspondant à la ville à partir des deux premiers chiffres du code postal. - L'administrateur continue à saisir les informations restantes de la nouvelle ville. - Il clique sur le bouton «valider». - Le système enregistre les données saisies. - Il affiche un message indiquant l'ajout avec succès de la nouvelle ville.
niceblueExceptions	<ul style="list-style-type: none"> - Si un champ obligatoire manque la saisie, le système affiche un message d'erreur. - Si le code postal saisi est inexistant dans la base, le système affiche un message d'erreur.

Table 3.13: Spécification du cas d'utilisation «AjouterVille»

• Raffinement du cas d'utilisation «ModifierVille»

niceblueCas d'utilisation	ModifierVille
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Informations modifiées d'une ville
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit la ville à modifier depuis l'interface de consultation. - Il clique sur l'icône de la modification. - Le système affiche la page de la modification avec les informations de la délégation choisie. - L'administrateur modifie les informations désirées. - Il clique sur le bouton «valider». - Le système met à jour la ville avec les nouvelles informations. - Il affiche un message indiquant la modification avec succès de la ville.
niceblueExceptions	Si un champ obligatoire manque la saisie, le système affiche un message d'erreur.

Table 3.14: Spécification du cas d'utilisation «ModifierVille»

• Raffinement du cas d'utilisation «SupprimerVille»

niceblueCas d'utilisation	SupprimerVille
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Ville supprimée

niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit la ville à supprimer depuis l'interface de consultation. - Il clique sur l'icône de suppression. - Le système affiche un message d'avertissement indiquant l'accord de suppression. - L'administrateur clique sur le bouton «Oui». - Le système affiche un message indiquant la suppression avec succès de la ville.
niceblueExceptions	Si l'administrateur clique sur le bouton «Non», la ville ne sera pas supprimée.

Table 3.15: Spécification du cas d'utilisation «SupprimerVille»

F- Raffinement du cas d'utilisation «Gérer les utilisateurs» :

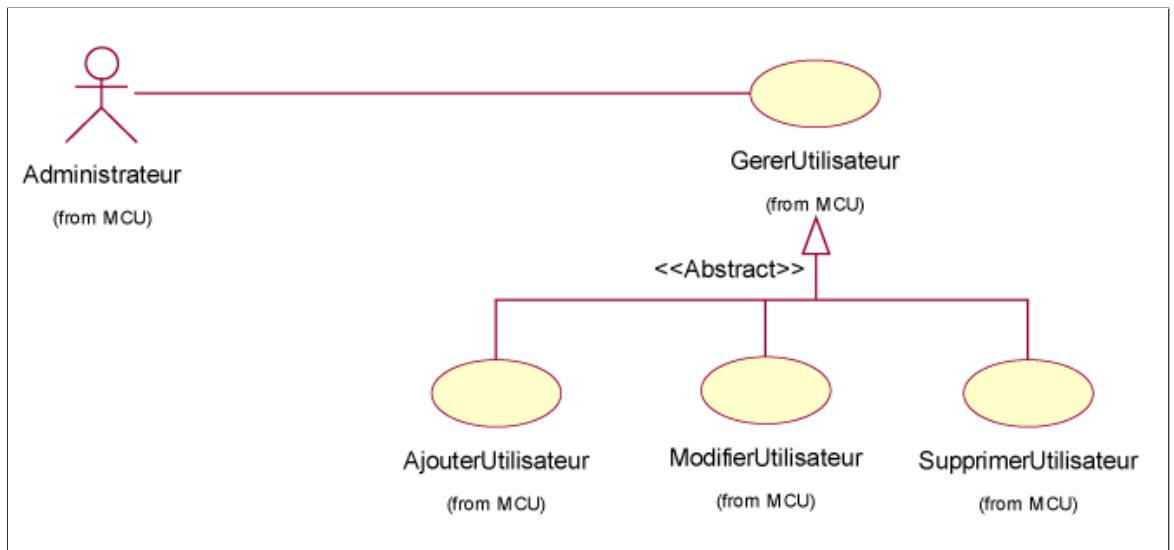


Figure 3.6: Raffinement du cas d'utilisation «GererUtilisateur»

- Raffinement du cas d'utilisation «AjouterUtilisateur»

niceblueCas d'utilisation	AjouterUtilisateur
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Utilisateur ajouté
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Nouveau utilisateur» de l'interface de consultation. - L'administrateur saisit l'email du nouveau utilisateur. - Le système vérifie l'inexistence de cette adresse électronique dans la base. - L'administrateur continue à saisir les informations restantes de l'utilisateur. - Il clique sur le bouton «valider». - Le système enregistre les données saisies. - Il affiche un message indiquant l'ajout avec succès du nouveau utilisateur.

niceblueExceptions	Si un champ obligatoire manque la saisie ou bien ne respecte pas la format (pour email par exemple), le système affiche un message d'erreur. - Si l'adresse électronique saisie est existante dans la base, le système affiche un message d'erreur.
--------------------	--

Table 3.16: Spécification du cas d'utilisation «AjouterUtilisateur»

- Raffinement du cas d'utilisation «SupprimerUtilisateur»

niceblueCas d'utilisation	SupprimerUtilisateur
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Utilisateur supprimé
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit l'utilisateur à supprimer depuis l'interface de consultation. - Il clique sur l'icône de suppression. <p>Le système affiche un message d'avertissement indiquant l'accord de suppression.</p> <ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Oui». - Le système affiche un message indiquant la suppression avec succès de l'utilisateur.
niceblueExceptions	Si l'administrateur clique sur le bouton «Non», l'utilisateur ne sera pas supprimé.

Table 3.17: Spécification du cas d'utilisation «SupprimerUtilisateur»

G- Raffinement du cas d'utilisation «Gérer les adhérents» :

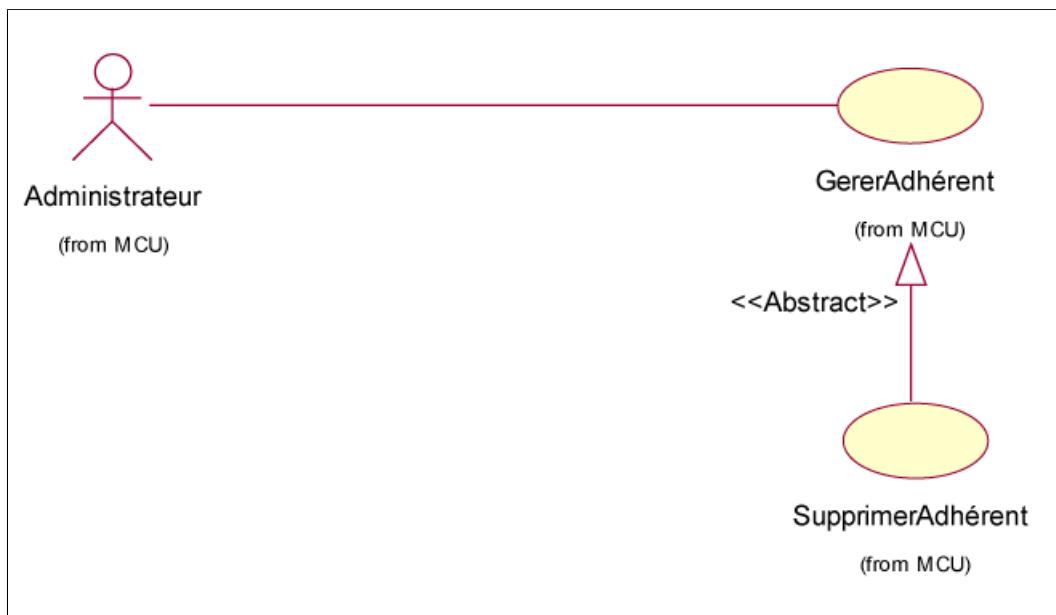


Figure 3.7: Raffinement du cas d'utilisation «GererAdherent»

- Raffinement du cas d'utilisation «SupprimerAdherent»

niceblueCas d'utilisation	SupprimerAdherent
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Adhérent supprimé
niceblueScénario principal	<ul style="list-style-type: none"> - L'administrateur choisit l'adhérent à supprimer depuis l'interface de consultation. - Il clique sur l'icône de suppression. <p>Le système affiche un message d'avertissement indiquant l'accord de suppression.</p> <ul style="list-style-type: none"> - L'administrateur clique sur le bouton «Oui». <p>Le système affiche un message indiquant la suppression avec succès de l'adhérent.</p>
niceblueExceptions	Si l'administrateur clique sur le bouton «Non», l'adhérent ne sera pas supprimé.

Table 3.18: Spécification du cas d'utilisation «SupprimerAdherent»

3.2 Conception

3.2.1 Conception des cas d'utilisation de priorité 2

A- La conception du cas d'utilisation «GererAgence»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

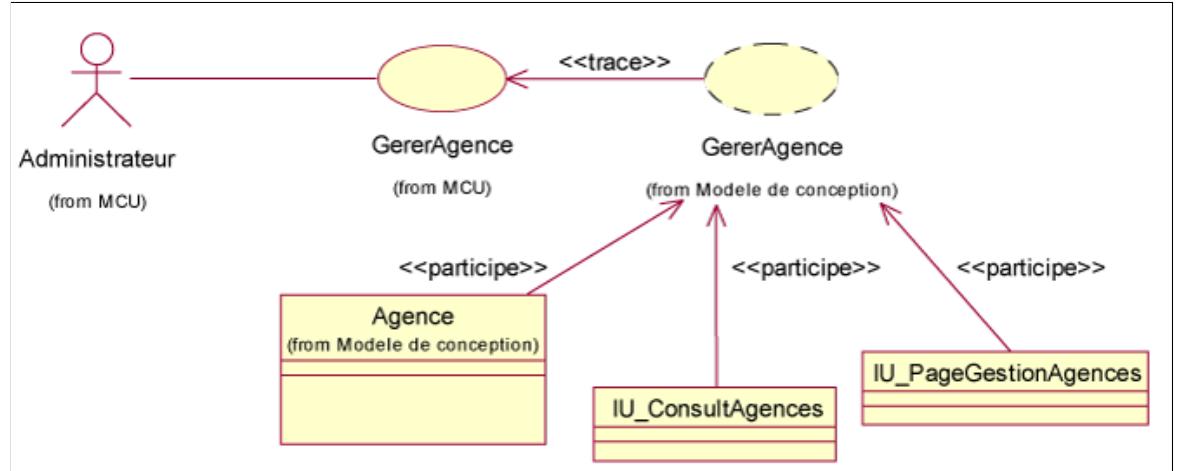


Figure 3.8: Traçabilité MCU/MC du cas d'utilisation «GererAgence»

- Diagrammes de séquence :

■ Diagramme de séquence du cas d'utilisation «AjouterAgence»:

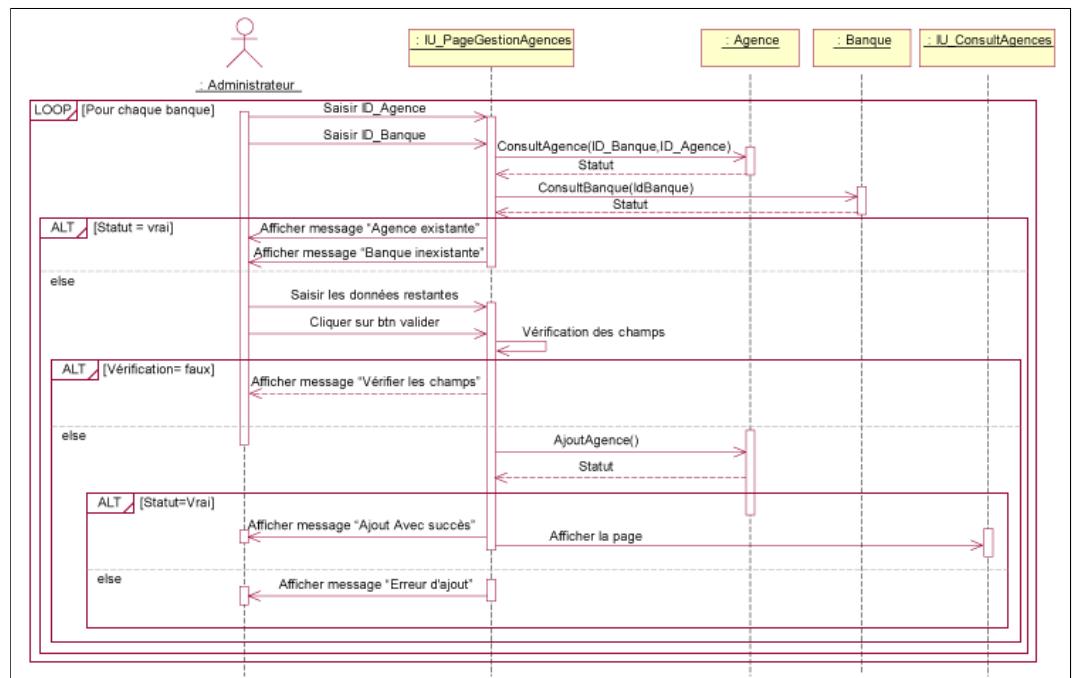


Figure 3.9: Diagramme de séquence du cas d'utilisation «AjouterAgence»

■ Diagramme de séquence du cas d'utilisation «ModifierAgence»:

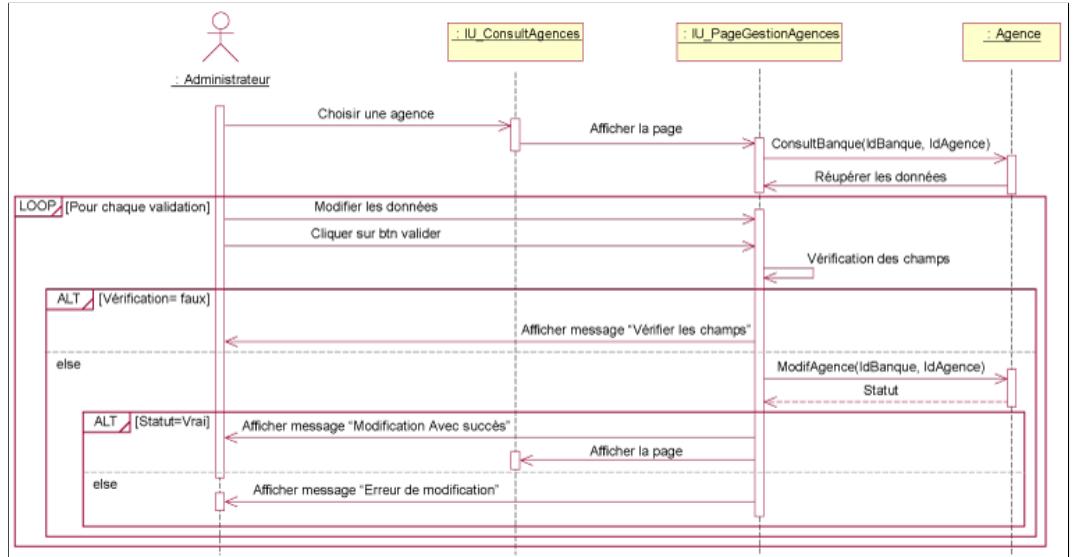


Figure 3.10: Diagramme de séquence du cas d'utilisation
«ModifierAgence»

■ Diagramme de séquence du cas d'utilisation «SupprimerAgence»:

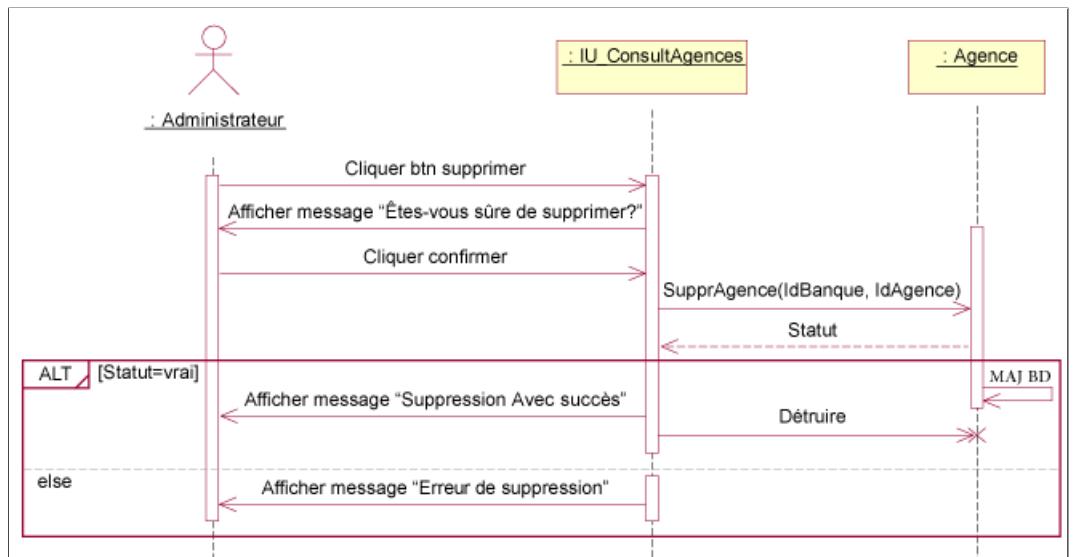


Figure 3.11: Diagramme de séquence du cas d'utilisation
«SupprimerAgence»

B- La conception du cas d'utilisation «GererDAB»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

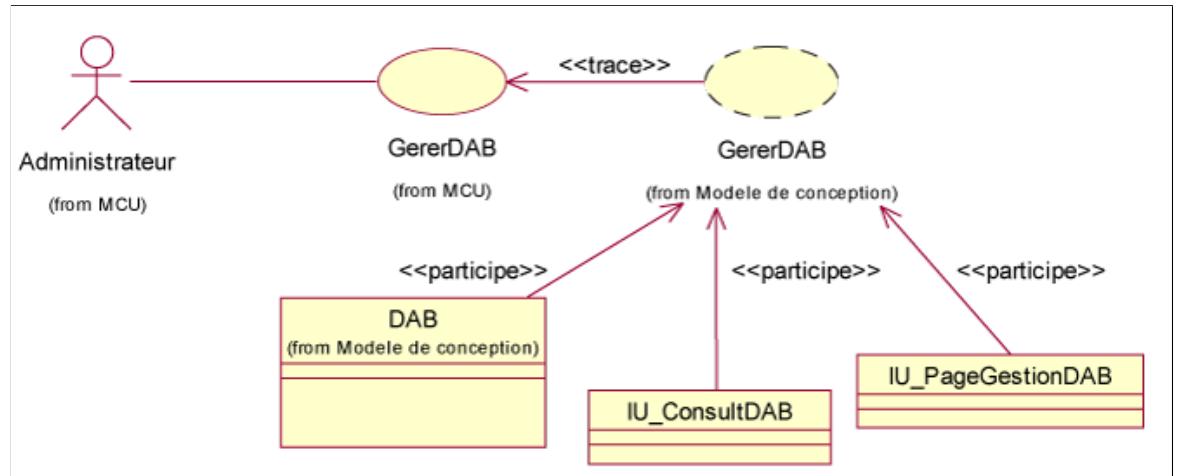


Figure 3.12: Traçabilité MCU/MC du cas d'utilisation «GererDAB»

- Diagrammes de séquence :
- Diagramme de séquence du cas d'utilisation «AjouterDAB»:

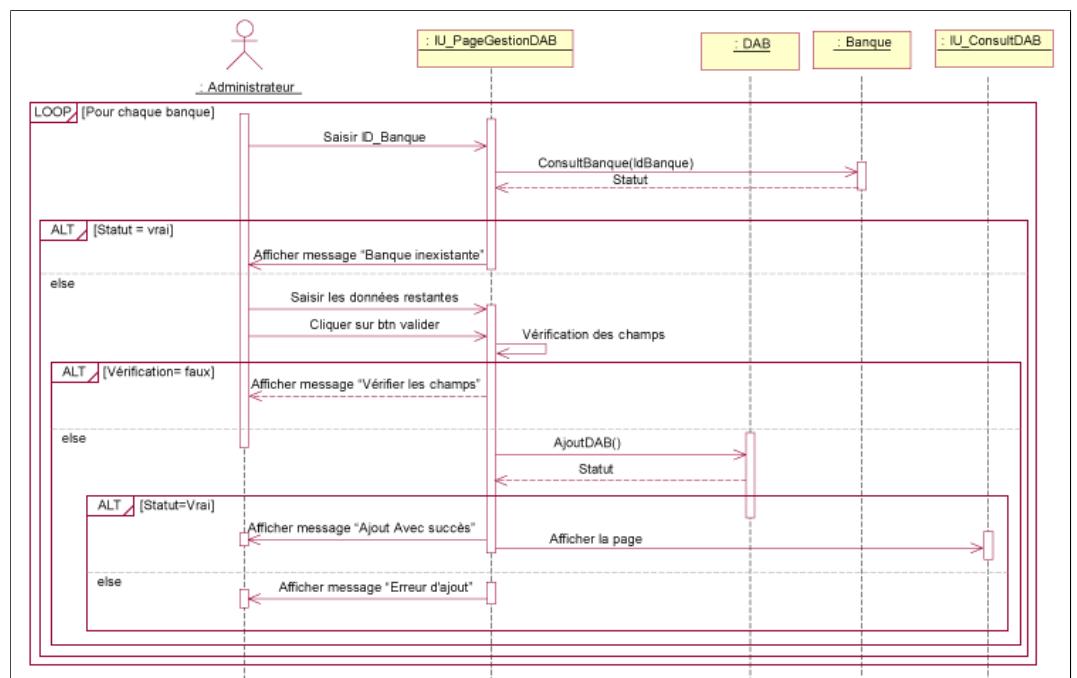


Figure 3.13: Diagramme de séquence du cas d'utilisation «AjouterDAB»

■ Diagramme de séquence du cas d'utilisation «ModifierDAB»:

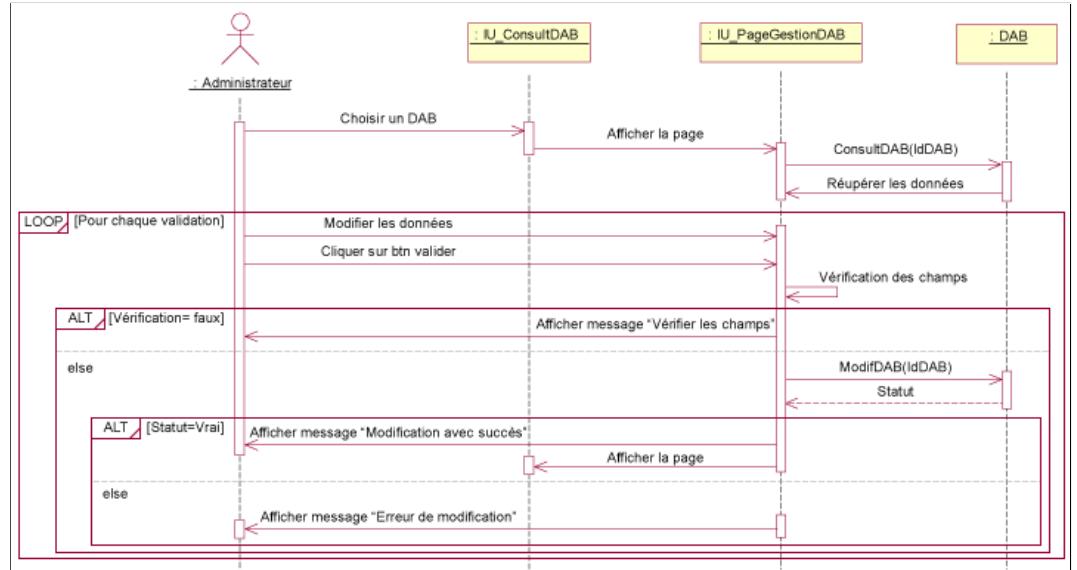


Figure 3.14: Diagramme de séquence du cas d'utilisation «ModifierDAB»

■ Diagramme de séquence du cas d'utilisation «SupprimerDAB»:

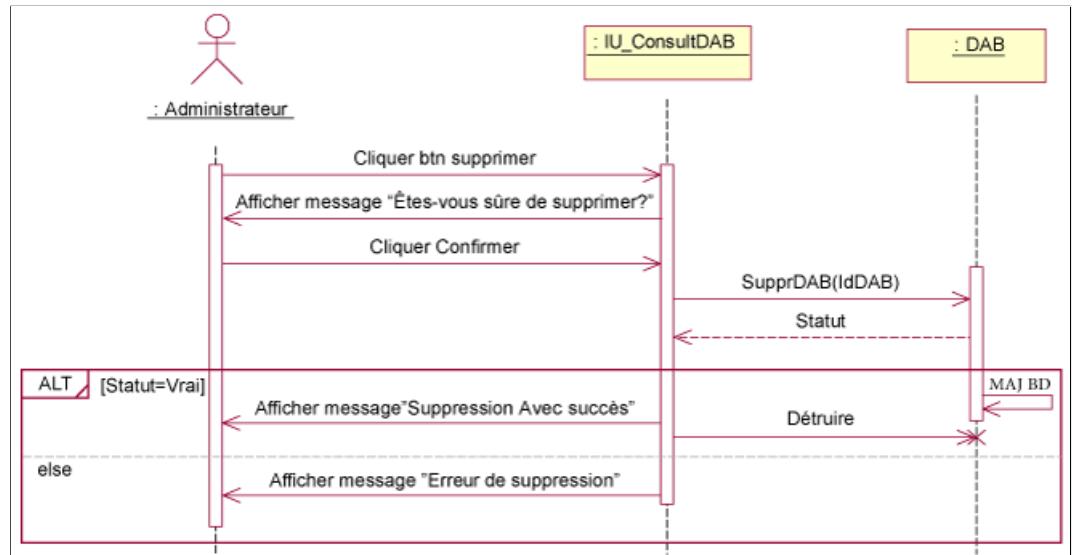


Figure 3.15: Diagramme de séquence du cas d'utilisation «SupprimerDAB»

C- La conception du cas d'utilisation «GererGouvernorat»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

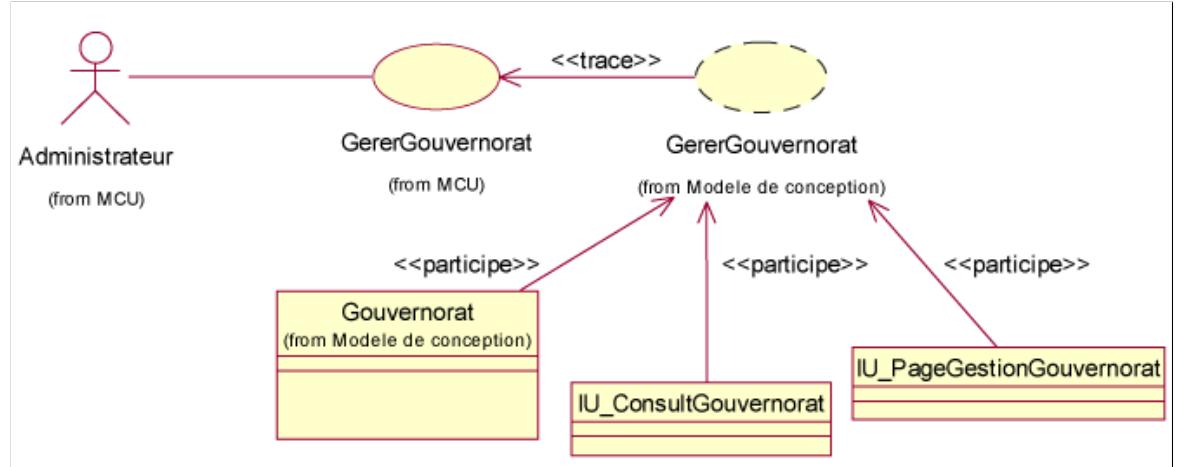


Figure 3.16: Traçabilité MCU/MC du cas d'utilisation
«GererGouvernorat»

- Diagrammes de séquence :
- Diagramme de séquence du cas d'utilisation «AjouterGouvernorat»:

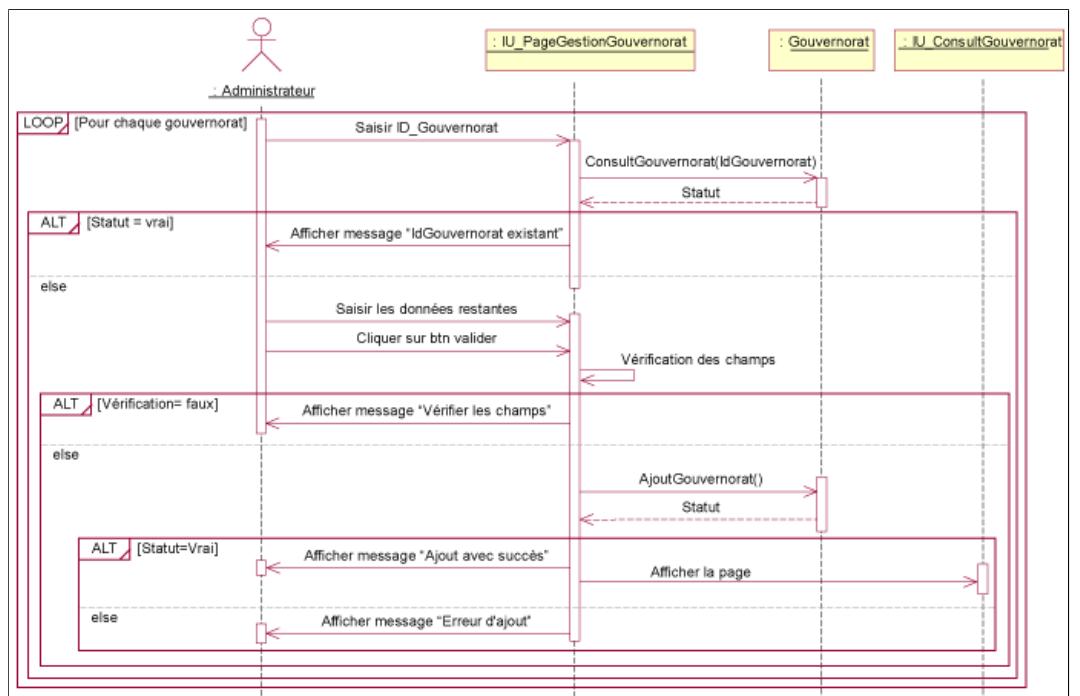


Figure 3.17: Diagramme de séquence du cas d'utilisation
«AjouterGouvernorat»

■ Diagramme de séquence du cas d'utilisation «ModifierGouvernorat»:

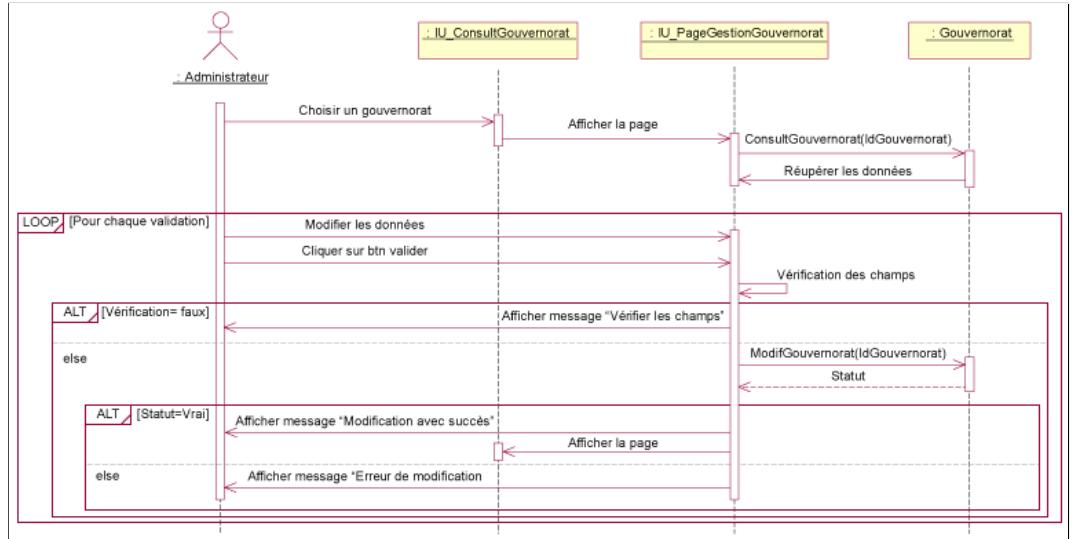


Figure 3.18: Diagramme de séquence du cas d'utilisation
«ModifierGouvernorat»

■ Diagramme de séquence du cas d'utilisation «SupprimerGouvernorat»:

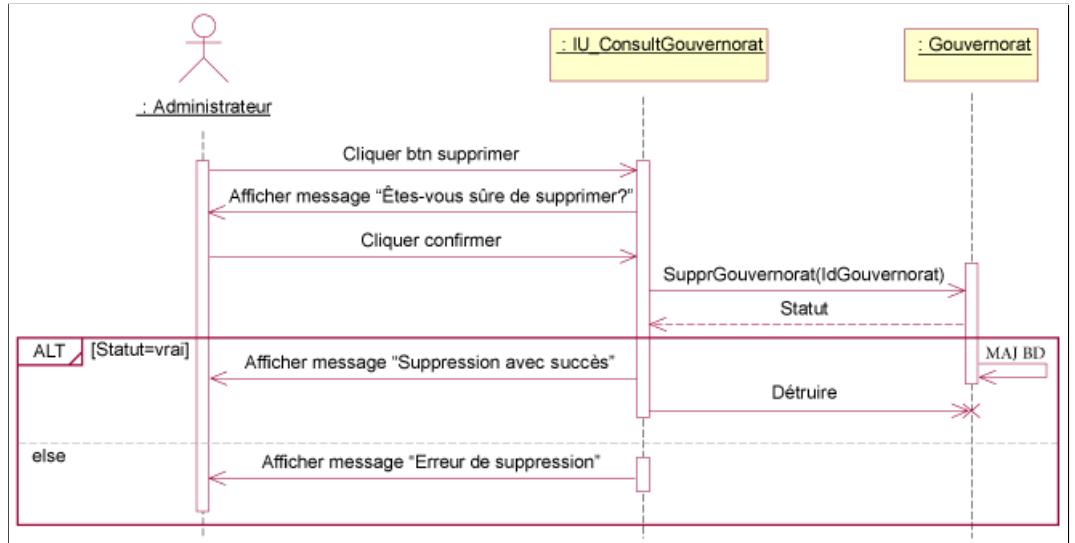


Figure 3.19: Diagramme de séquence du cas d'utilisation
«SupprimerGouvernorat»

D- La conception du cas d'utilisation «GererDelegation»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

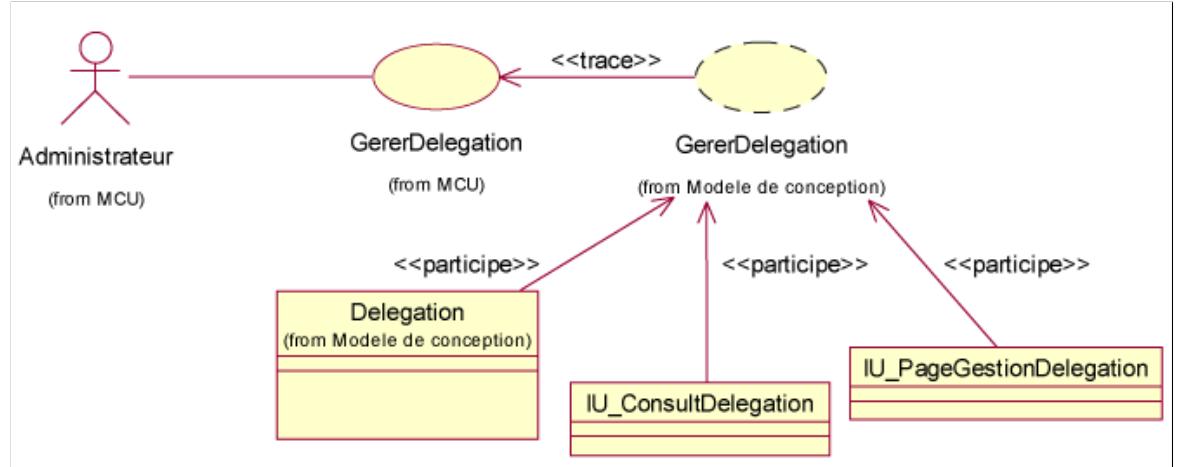


Figure 3.20: Traçabilité MCU/MC du cas d'utilisation
«GererDelegation»

- Diagrammes de séquence :

- Diagramme de séquence du cas d'utilisation «AjouterDelegation»:

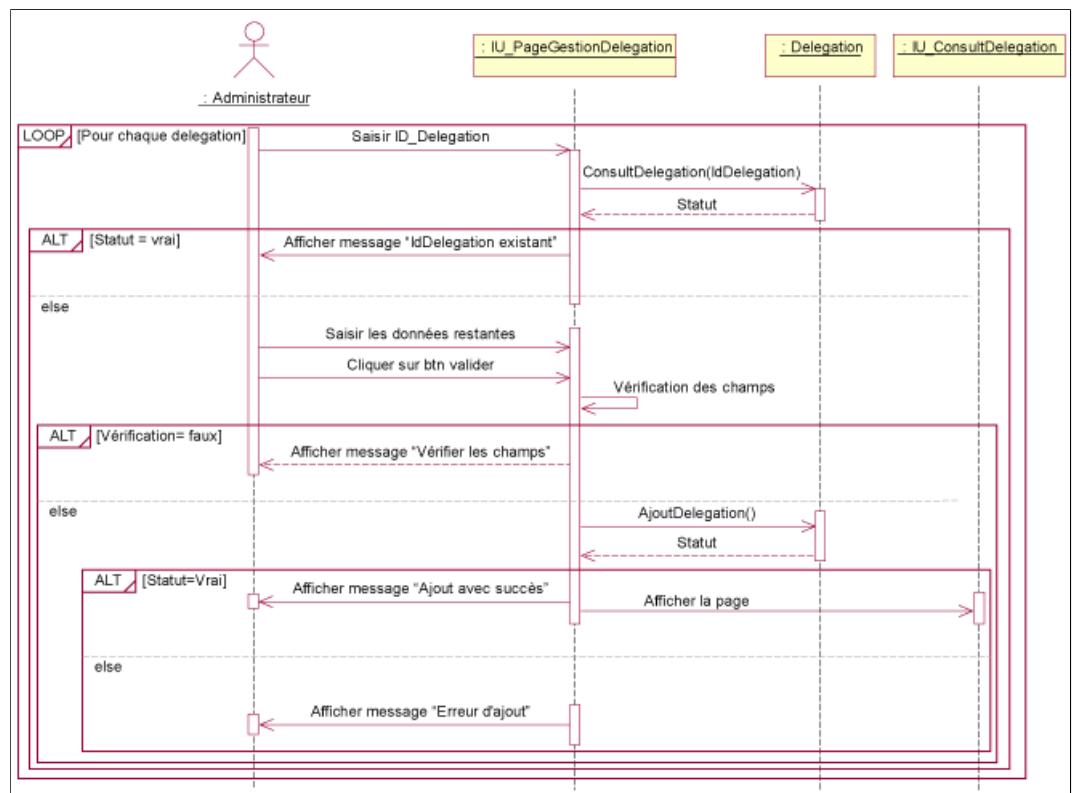


Figure 3.21: Diagramme de séquence du cas d'utilisation
«AjouterDelegation»

■ Diagramme de séquence du cas d'utilisation «ModifierDelegation»:

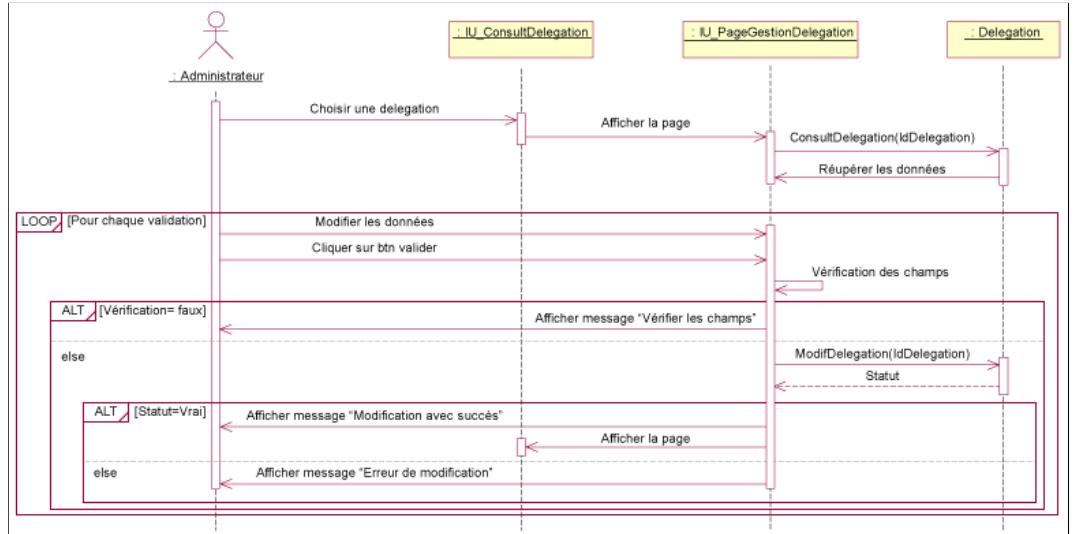


Figure 3.22: Diagramme de séquence du cas d'utilisation
«ModifierDelegation»

■ Diagramme de séquence du cas d'utilisation «SupprimerDelegation»:

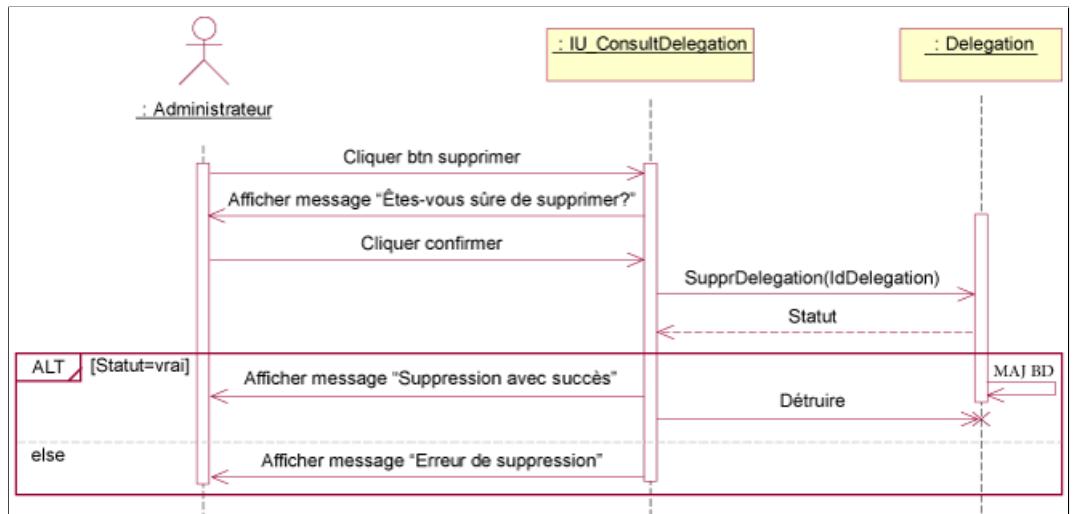


Figure 3.23: Diagramme de séquence du cas d'utilisation
«SupprimerDelegation»

E- La conception du cas d'utilisation «GererVille»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

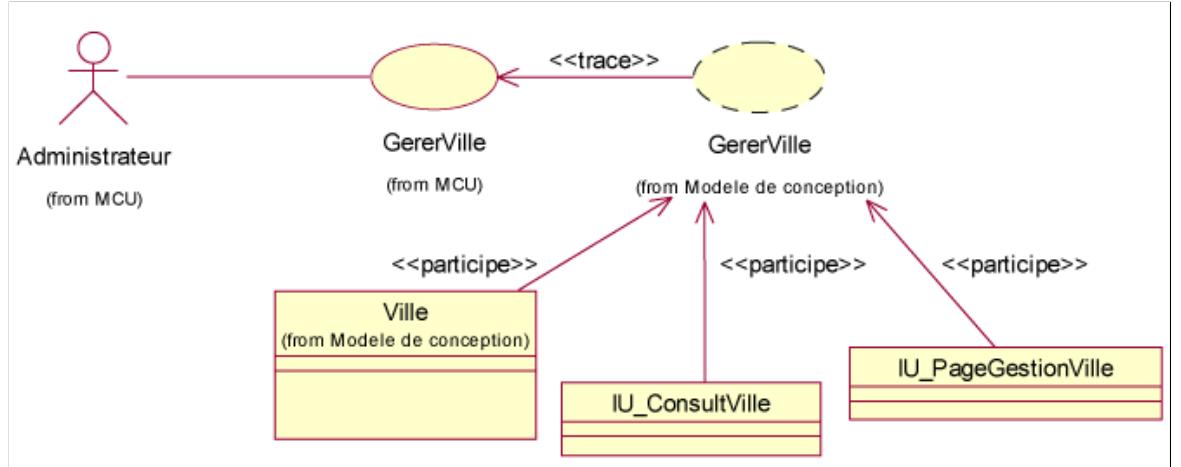


Figure 3.24: Traçabilité MCU/MC du cas d'utilisation «GererVille»

- Diagrammes de séquence :

- Diagramme de séquence du cas d'utilisation «AjouterVille»:

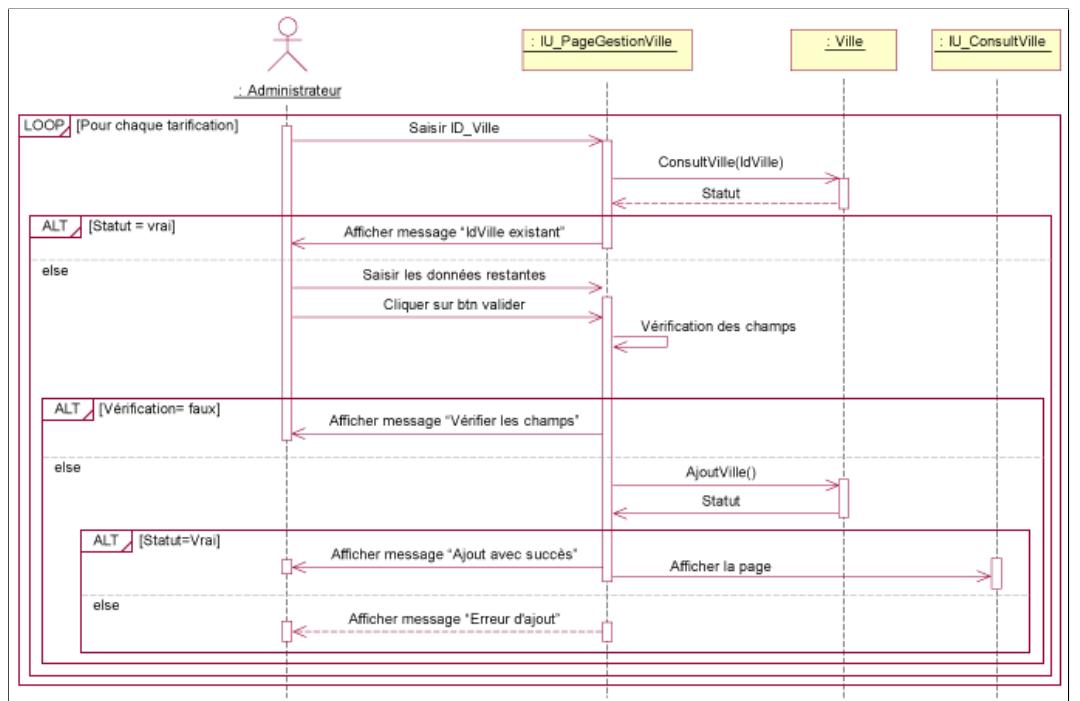


Figure 3.25: Diagramme de séquence du cas d'utilisation «AjouterVille»

■ Diagramme de séquence du cas d'utilisation «ModifierVille»:

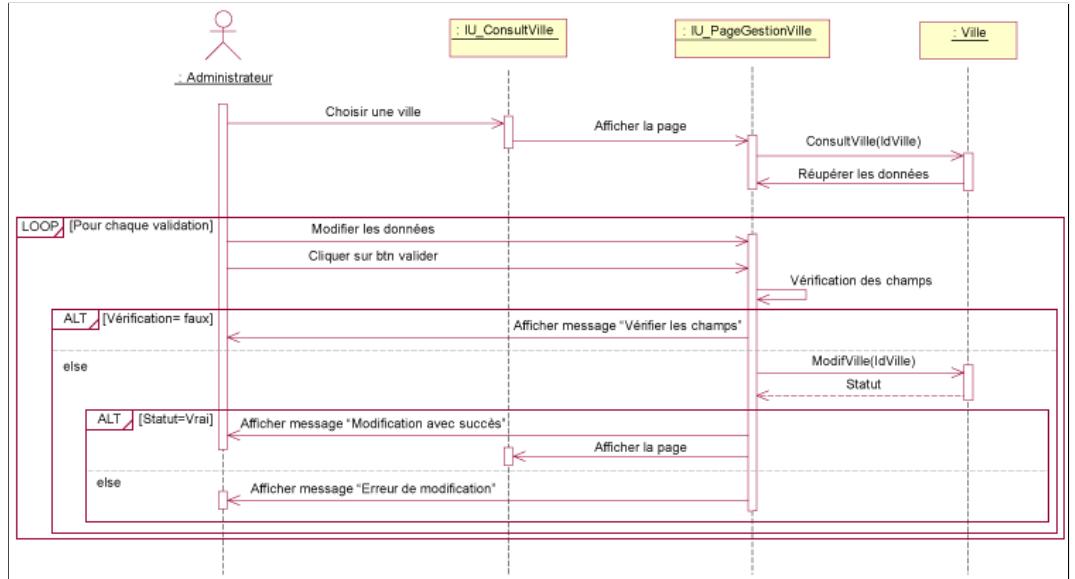


Figure 3.26: Diagramme de séquence du cas d'utilisation «ModifierVille»

■ Diagramme de séquence du cas d'utilisation «SupprimerVille»:

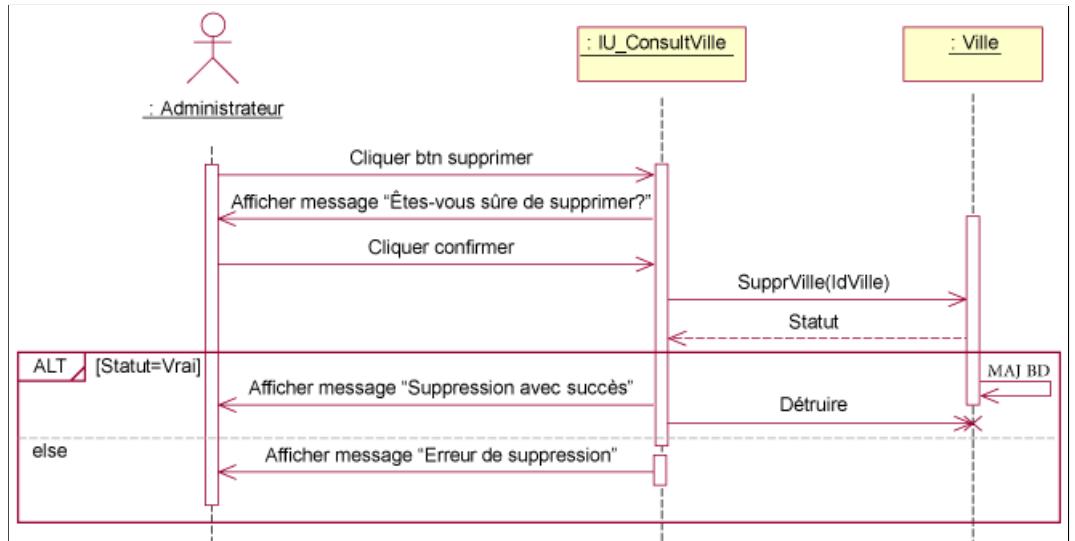


Figure 3.27: Diagramme de séquence du cas d'utilisation «SupprimerVille»

F- La conception du cas d'utilisation «GererUtilisateur»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

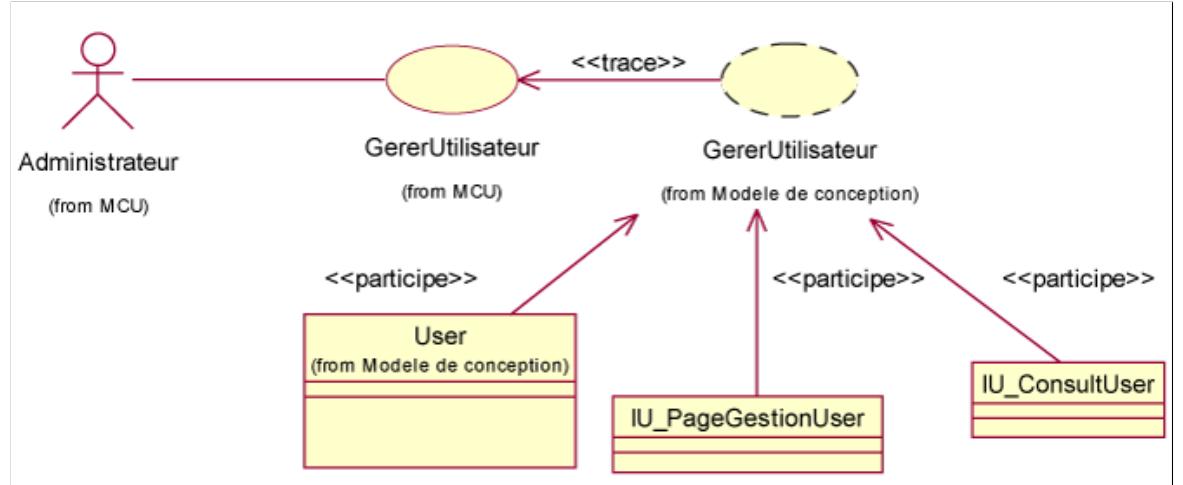


Figure 3.28: Traçabilité MCU/MC du cas d'utilisation «GererUtilisateur»

- Diagrammes de séquence :

- Diagramme de séquence du cas d'utilisation «AjouterUtilisateur»:

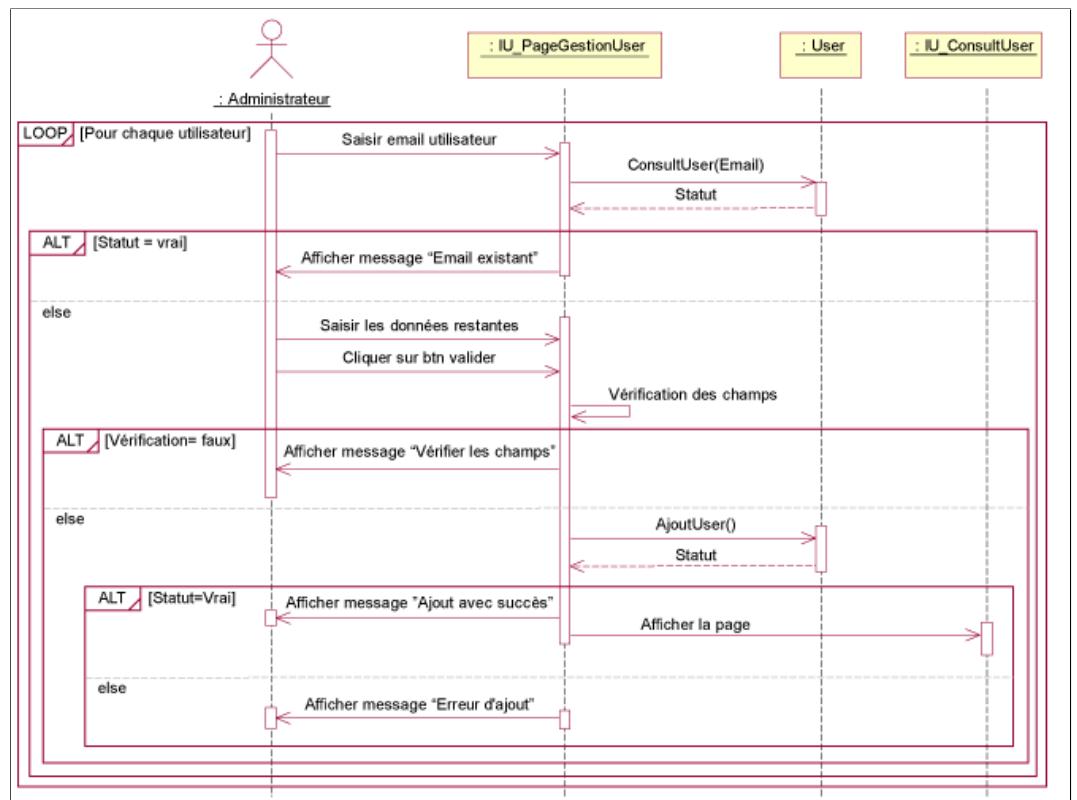


Figure 3.29: Diagramme de séquence du cas d'utilisation «AjouterUtilisateur»

- Diagramme de séquence du cas d'utilisation «SupprimerUtilisateur»:

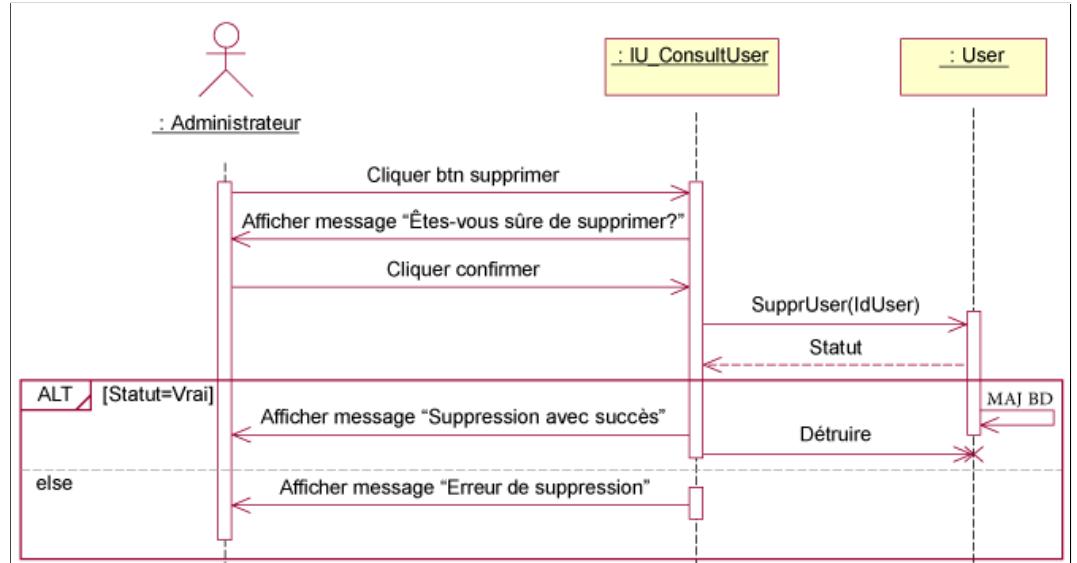


Figure 3.30: Diagramme de séquence du cas d'utilisation
«SupprimerUtilisateur»

G- La conception du cas d'utilisation «GererAdherent»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

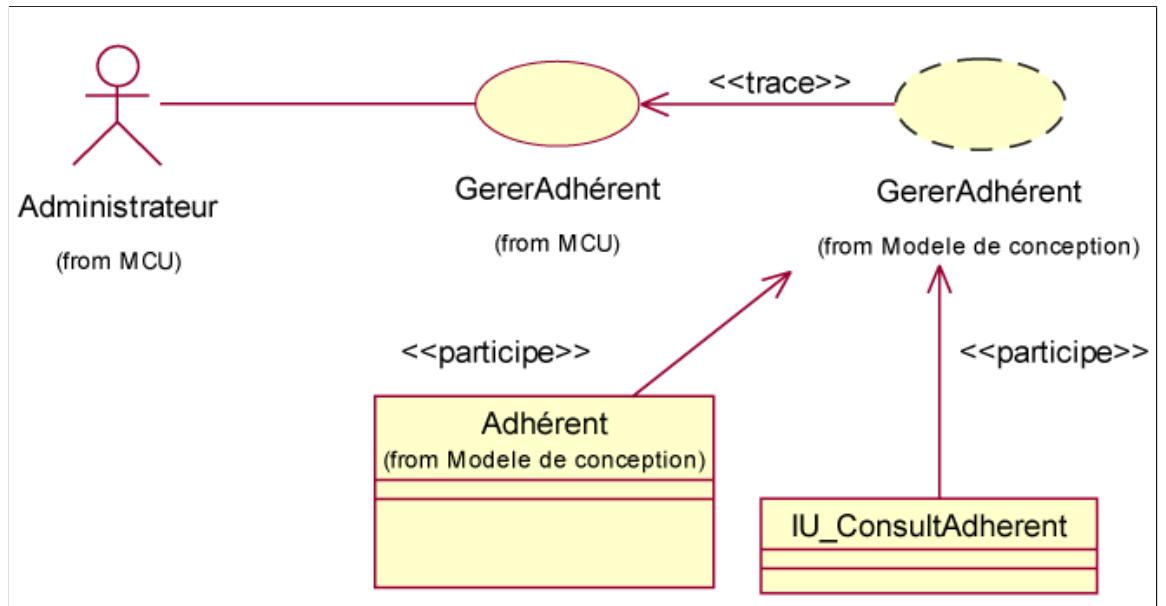


Figure 3.31: Traçabilité MCU/MC du cas d'utilisation
«GererAdhérent»

- Diagrammes de séquence :

- Diagramme de séquence du cas d'utilisation «SupprimerAdherent»:

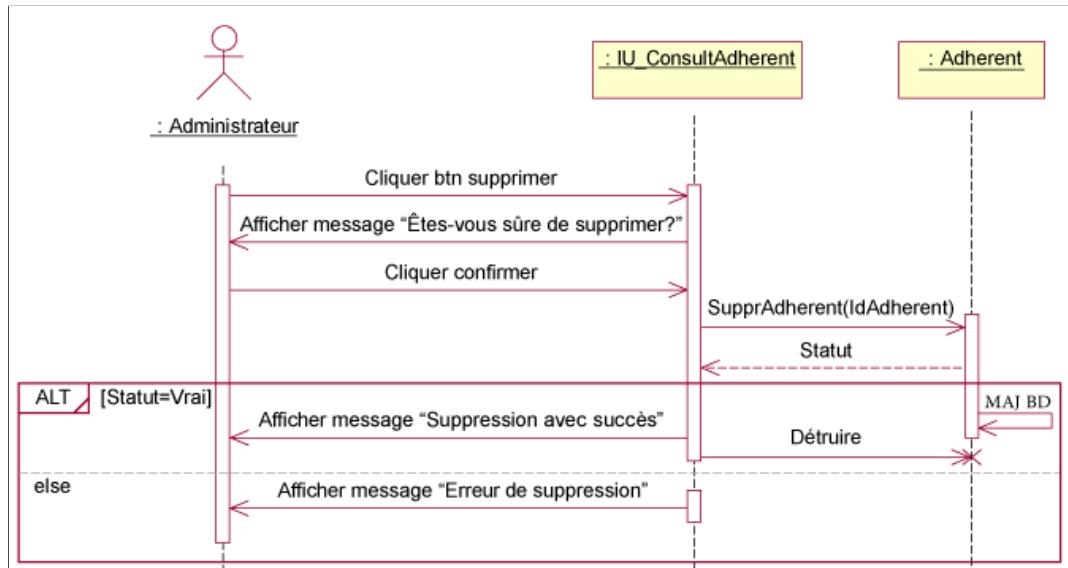


Figure 3.32: Diagramme de séquence du cas d'utilisation
«SupprimerAdherent»

3.3 Implémentation

3.3.1 Implémentation du cas d'utilisation «GererAgence»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

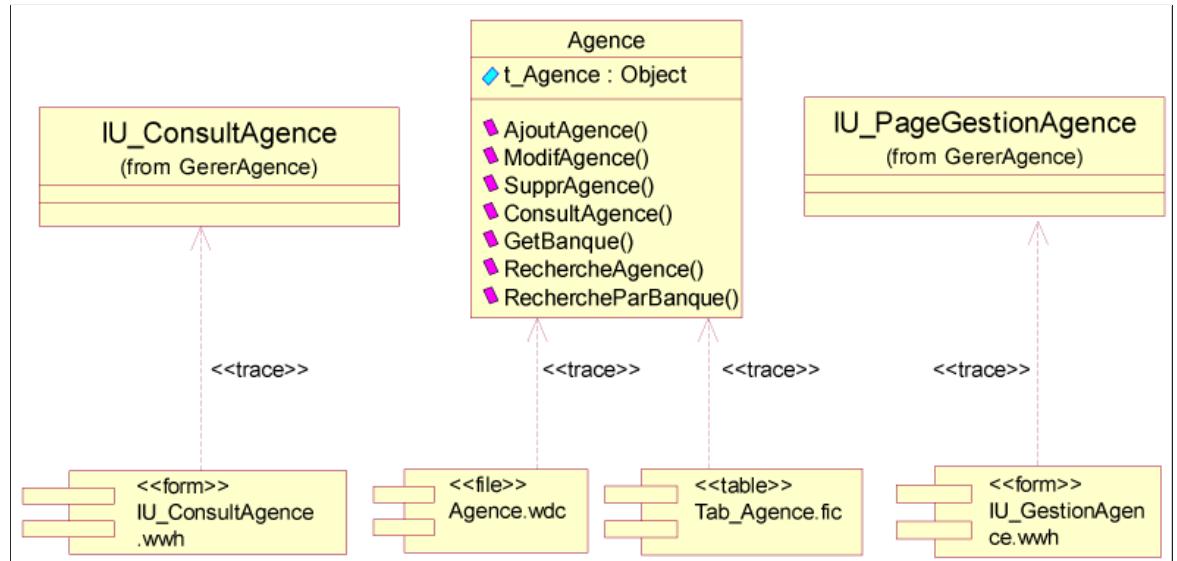


Figure 3.33: Traçabilité MC/MI du cas d'utilisation «GererAgence»

B- Diagramme de composant

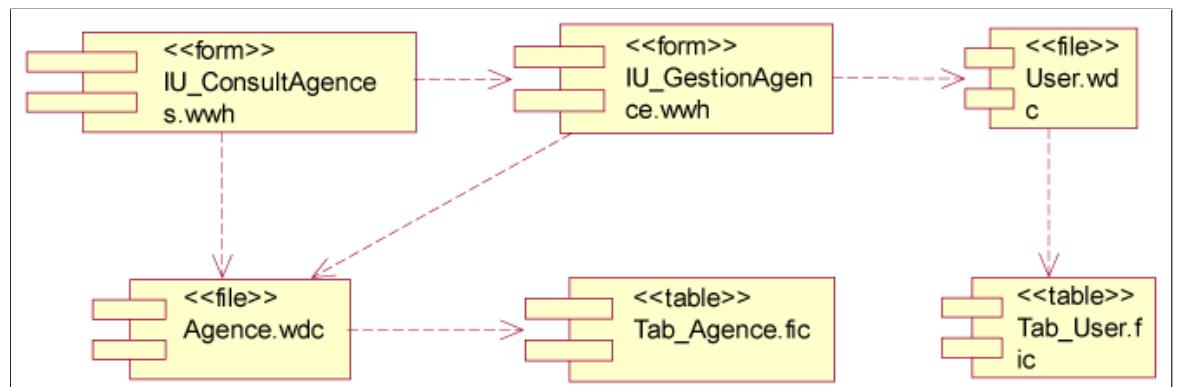


Figure 3.34: Diagramme de composant du cas d'utilisation «GererAgence»

3.3.2 Implémentation du cas d'utilisation «GererDAB»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

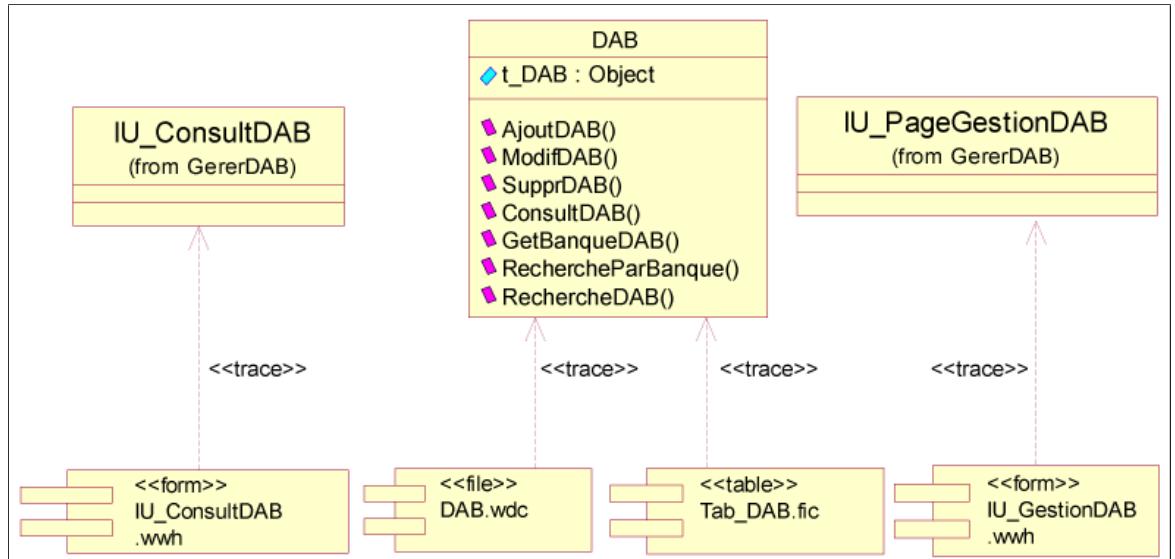


Figure 3.35: Traçabilité MC/MI du cas d'utilisation «GererDAB»

B- Diagramme de composant

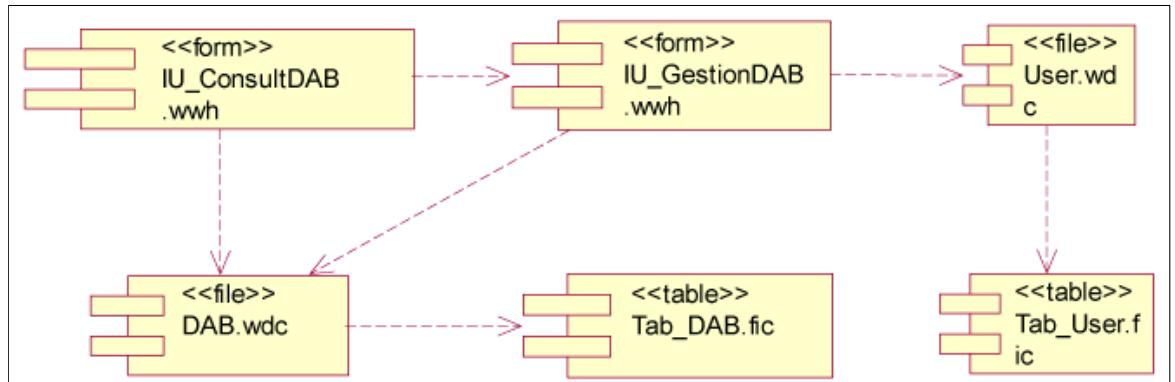


Figure 3.36: Diagramme de composant du cas d'utilisation «GererDAB»

3.3.3 Implémentation du cas d'utilisation «GererGouvernorat»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

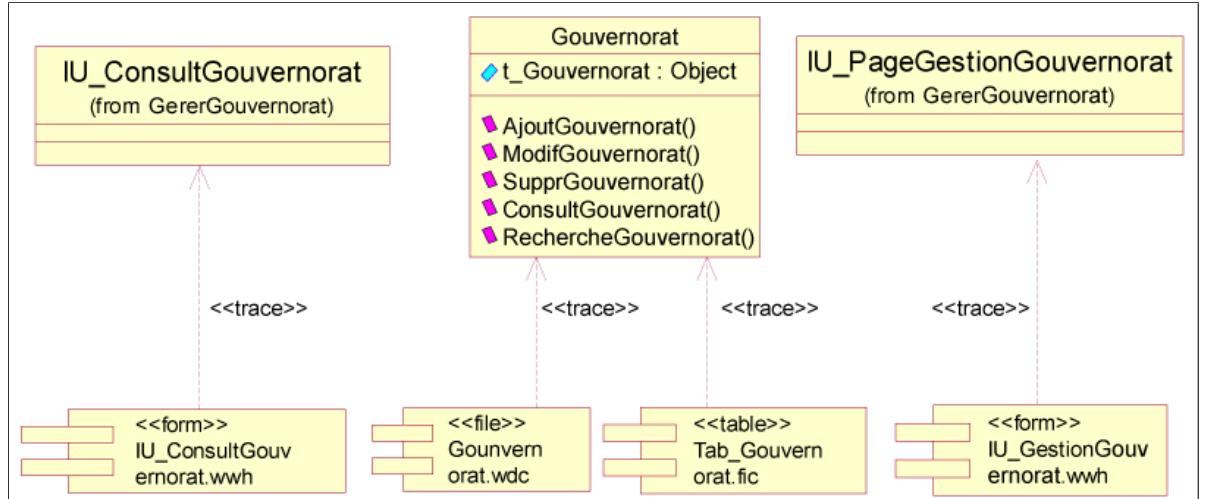


Figure 3.37: Traçabilité MC/MI du cas d'utilisation «GererGouvernorat»

B- Diagramme de composant

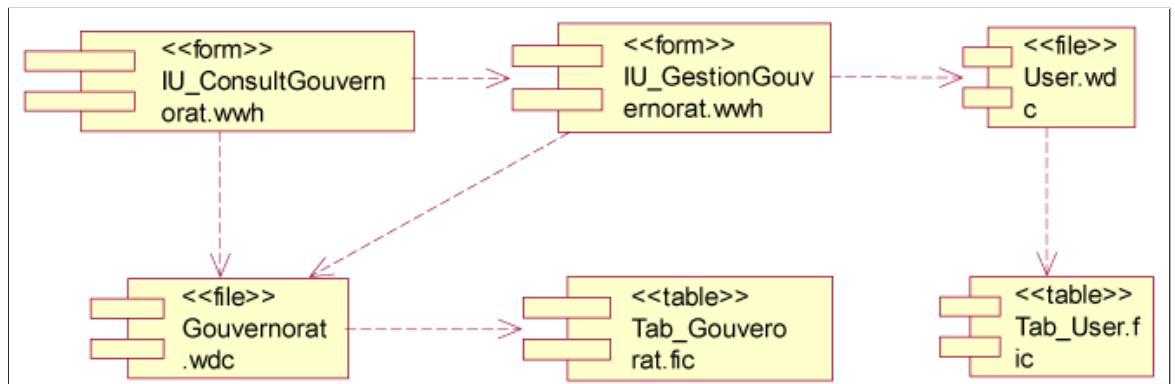


Figure 3.38: Diagramme de composant du cas d'utilisation «GererGouvernorat»

3.3.4 Implémentation du cas d'utilisation «GererDelegation»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

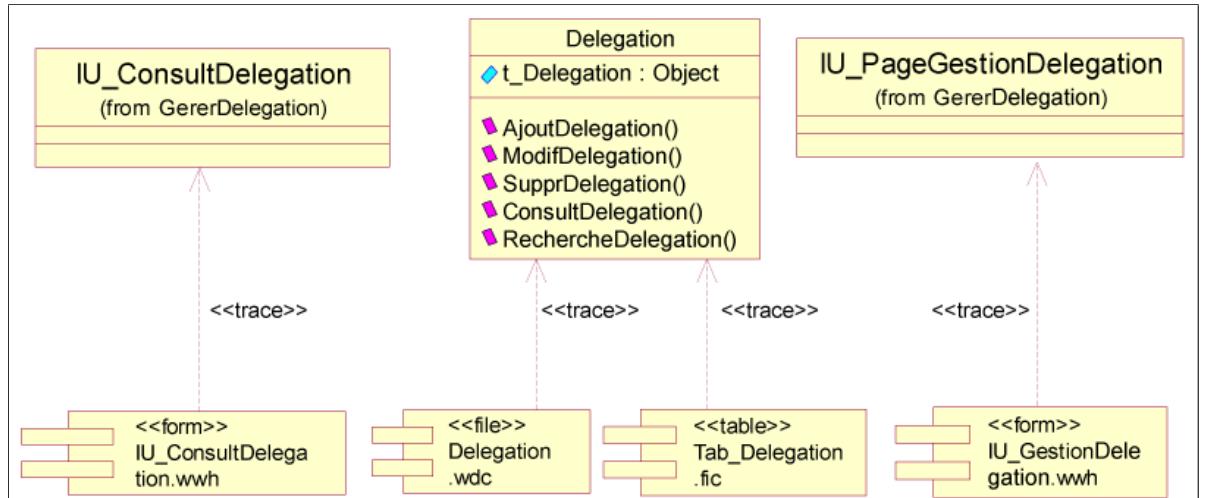


Figure 3.39: Traçabilité MC/MI du cas d'utilisation «GererDelegation»

B- Diagramme de composant

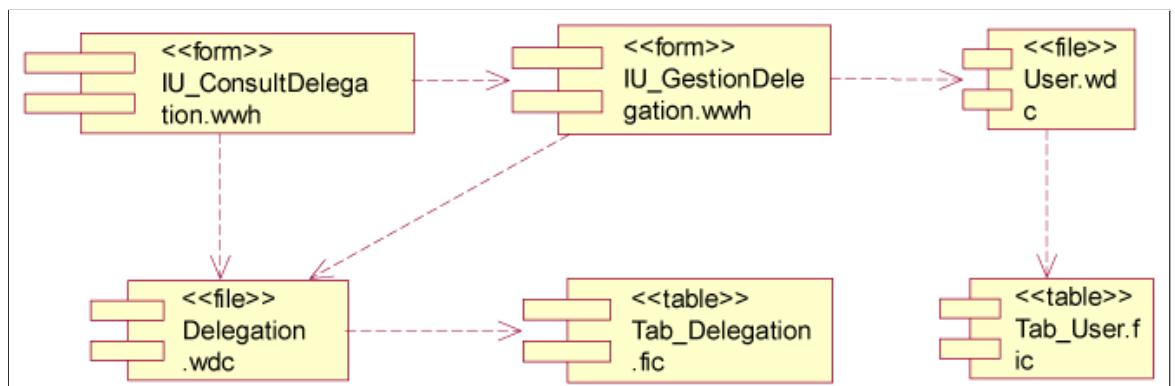


Figure 3.40: Diagramme de composant du cas d'utilisation
«GererDelegation»

3.3.5 Implémentation du cas d'utilisation «GererVille»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

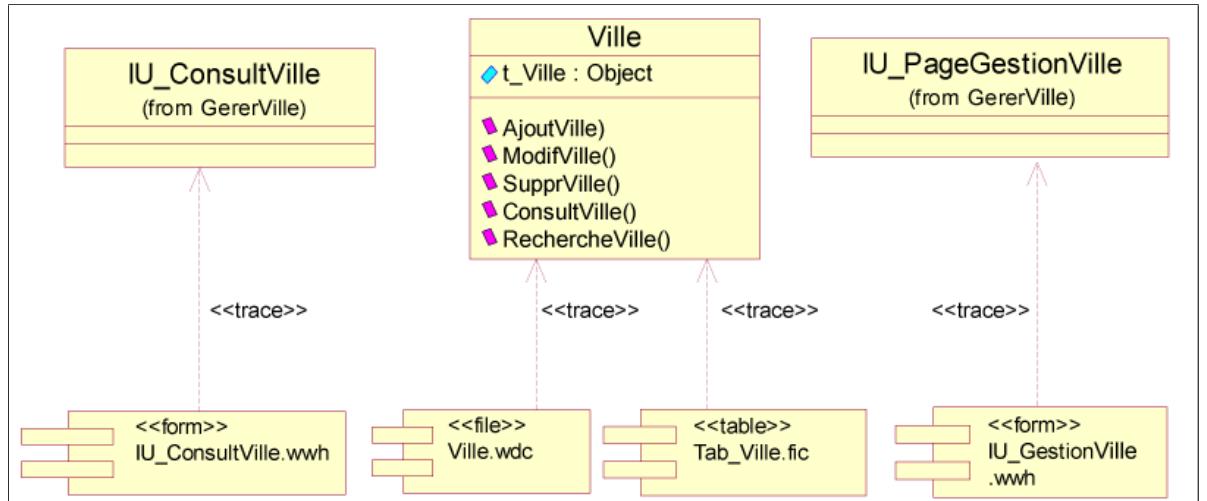


Figure 3.41: Traçabilité MC/MI du cas d'utilisation «GererVille»

B- Diagramme de composant

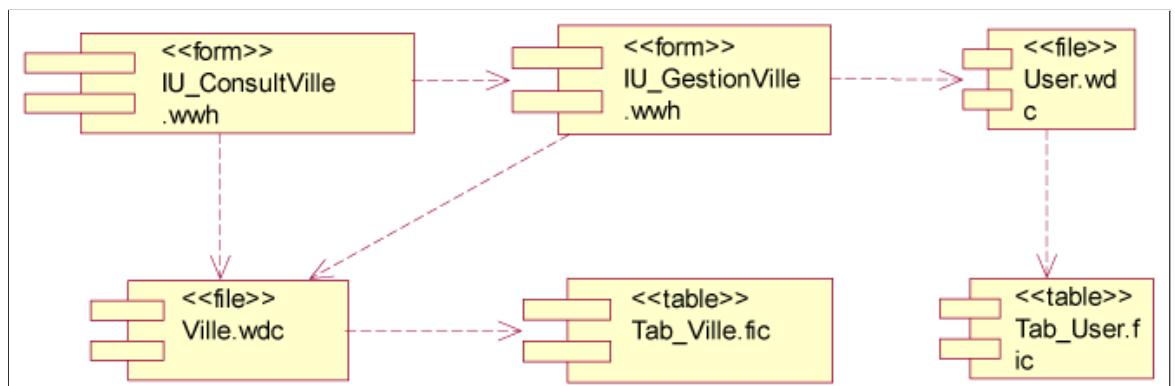


Figure 3.42: Diagramme de composant du cas d'utilisation «GererVille»

3.3.6 Implémentation du cas d'utilisation «GererUtilisateur»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

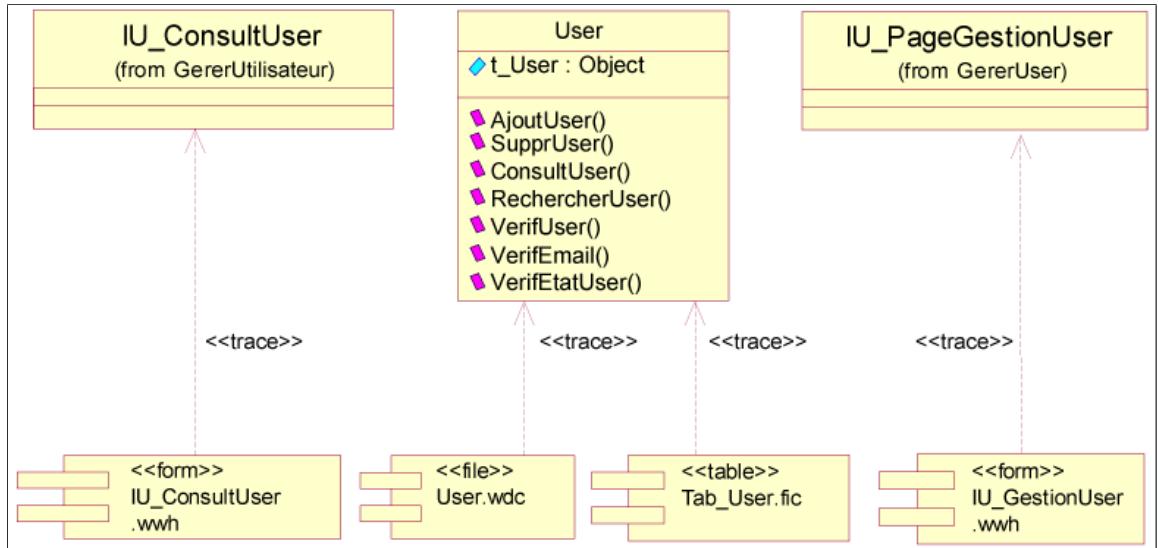


Figure 3.43: Traçabilité MC/MI du cas d'utilisation «GererUtilisateur»

B- Diagramme de composant

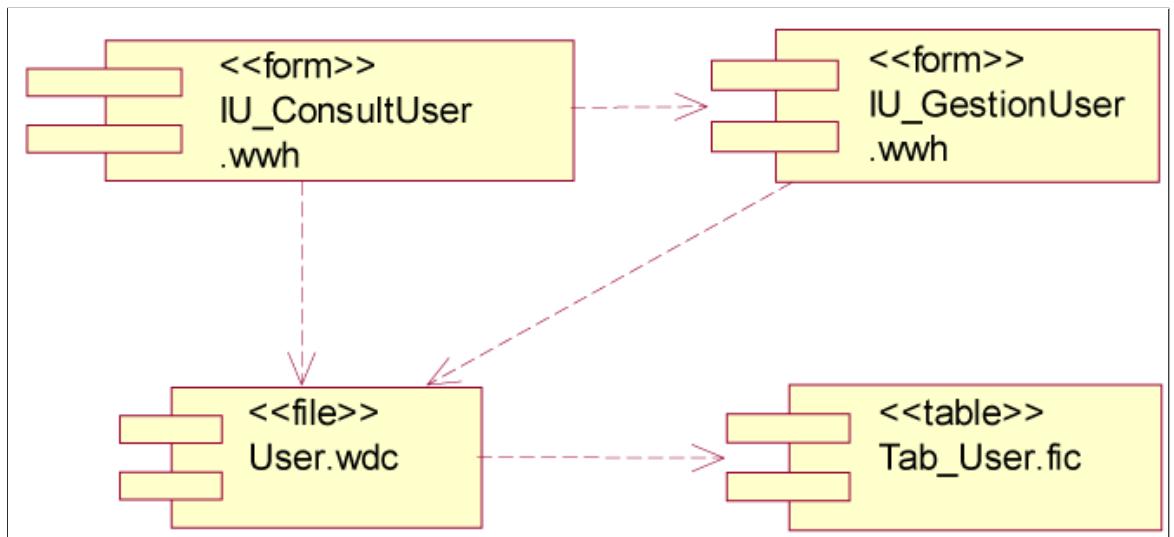


Figure 3.44: Diagramme de composant du cas d'utilisation
«GererUtilisateur»

3.3.7 Implémentation du cas d'utilisation «GererAdherent»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

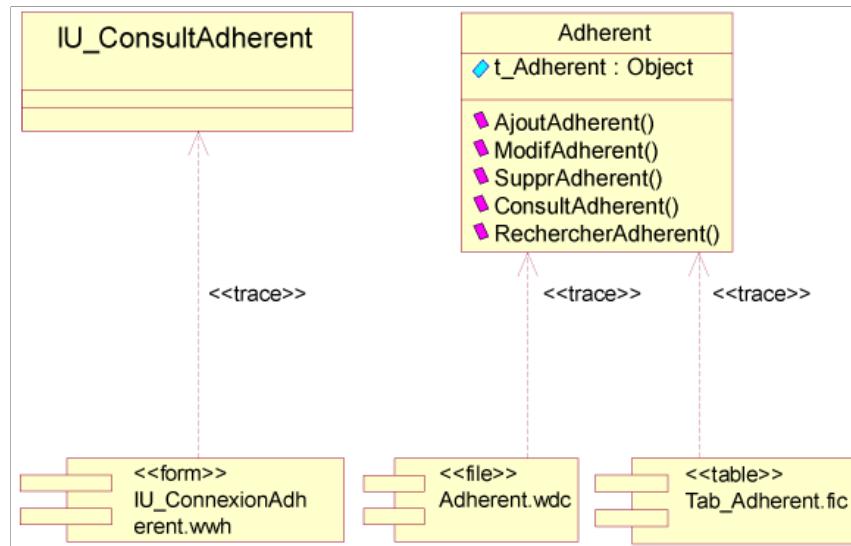


Figure 3.45: Traçabilité MC/MI du cas d'utilisation «GererAdherent»

B- Diagramme de composant

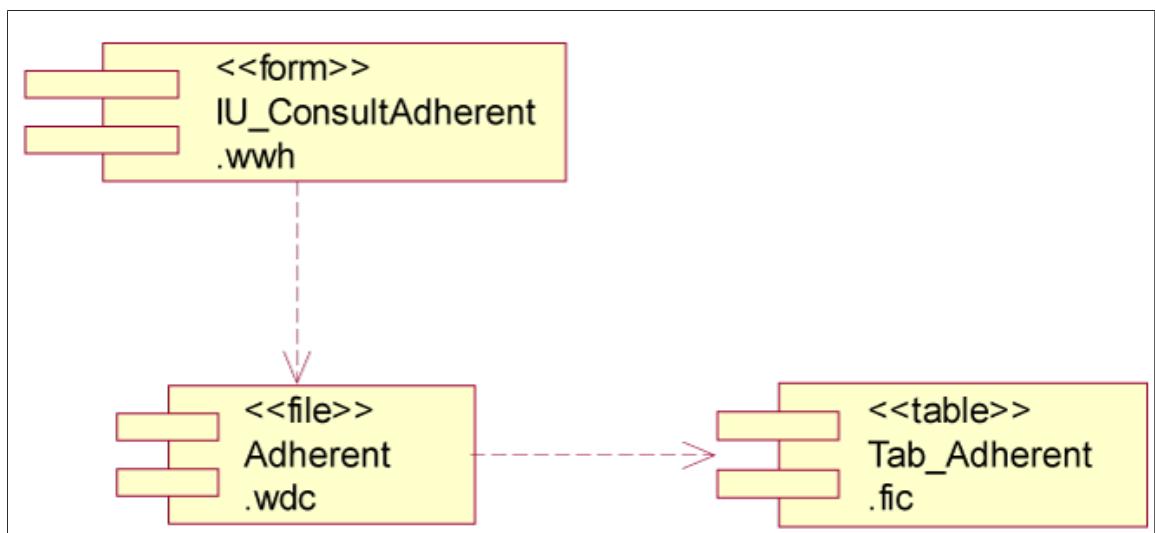


Figure 3.46: Diagramme de composant du cas d'utilisation «GererAdherent»

Conclusion

Au cours de cette phase, nous avons raffiné et conçu les cas d'utilisations de priorité 2, ainsi que leurs implémentations dans le but d'accomplir la conception de la base de données et de stabiliser l'architecture du système qui sera l'objectif du chapitre suivant.

4 La phase de construction

Introduction

Dans cette phase, nous allons capturer tous les besoins restants car il n'est pratiquement plus possible de le faire dans la prochaine phase. Ensuite, continuer la conception et surtout l'implémentation des cas d'utilisation de priorité 3 dans le but de réaliser un produit logiciel prêt à être exploité.

4.1 Capture des besoins

4.1.1 Raffinement des cas d'utilisation de priorité 3

A- Raffinement du cas d'utilisation «Consulter le trafic» :

niceblueCas d'utilisation	ConsulterTrafic
niceblueActeurs	Administrateur
nicebluePré-condition	Administrateur authentifié
nicebluePost-condition	Trafic consulté
niceblueScénario principal	- L'administrateur clique sur l'option trafic du menu principal - Le système récupère les données du trafic de la base et les affiche.

Table 4.1: Spécification du cas d'utilisation «ConsulterTrafic»

B- Raffinement du cas d'utilisation «Consulter les banques» :

niceblueCas d'utilisation	ConsulterBanque
niceblueActeurs	Adhérent
nicebluePost-condition	Liste des banques consultés
niceblueScénario principal	- L'adhérent clique sur le bouton banque de l'interface principale - Le système récupère les données des banques de la base et les affiche.

Table 4.2: Spécification du cas d'utilisation «ConsulterBanque»

C- Raffinement du cas d'utilisation «Consulter les agences» :

niceblueCas d'utilisation	ConsulterAgence
niceblueActeurs	Adhérent
nicebluePost-condition	Liste des agences consultés
niceblueScénario principal	- L'adhérent clique sur le bouton Agence de l'interface principale - L'administrateur choisit la banque désirée pour consulter ses agences.

- | | |
|--|--|
| | - Le système récupère les données des agences de la banque choisie à partir de la base et les affiche. |
|--|--|

Table 4.3: Spécification du cas d'utilisation «ConsulterAgence»

D- Raffinement du cas d'utilisation «ConsulterDAB» :

niceblueCas d'utilisation	ConsulterDAB
niceblueActeurs	Adhérent
nicebluePost-condition	Liste des agences consultés
niceblueScénario principal	<ul style="list-style-type: none"> - L'adhérent clique sur le bouton DAB de l'interface principale - L'administrateur choisit la banque désirée pour consulter ses DAB. - Le système récupère les données des DAB de la banque choisie à partir de la base et les affiche.

Table 4.4: Spécification du cas d'utilisation «ConsulterDAB»

4.2 Conception

4.2.1 Conception des cas d'utilisation de priorité 3

A- La conception du cas d'utilisation «ConsulterTrafic»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

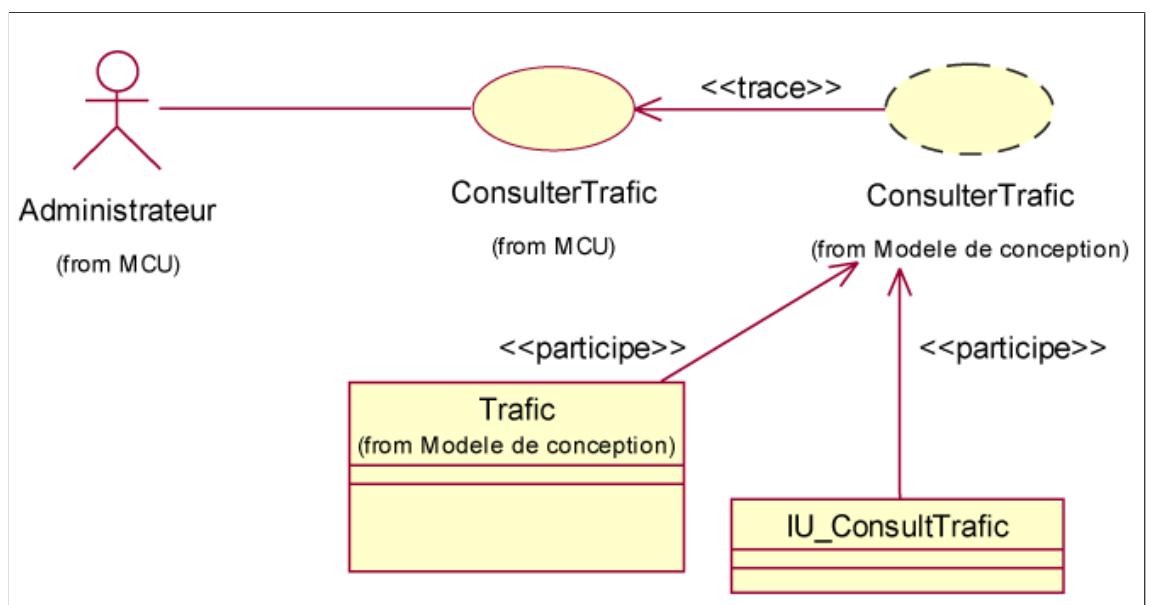


Figure 4.1: Traçabilité MCU/MC du cas d'utilisation «ConsulterTrafic»

- Diagramme de séquence :

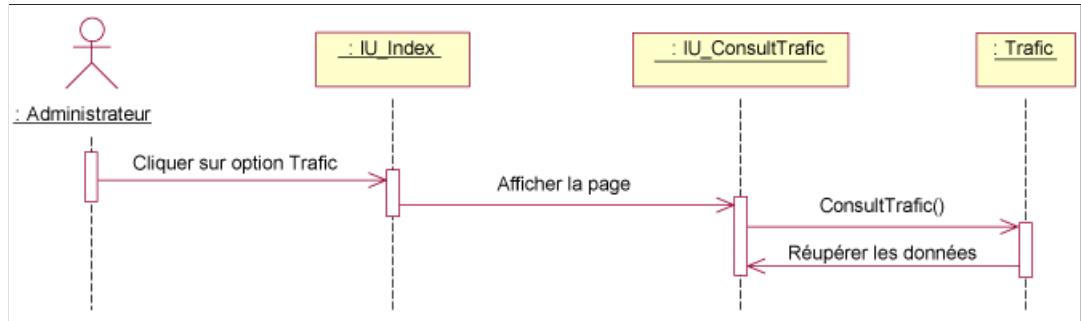


Figure 4.2: Diagramme de séquence du cas d'utilisation «ConsulterTrafic»

B- La conception du cas d'utilisation «ConsulterBanque»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

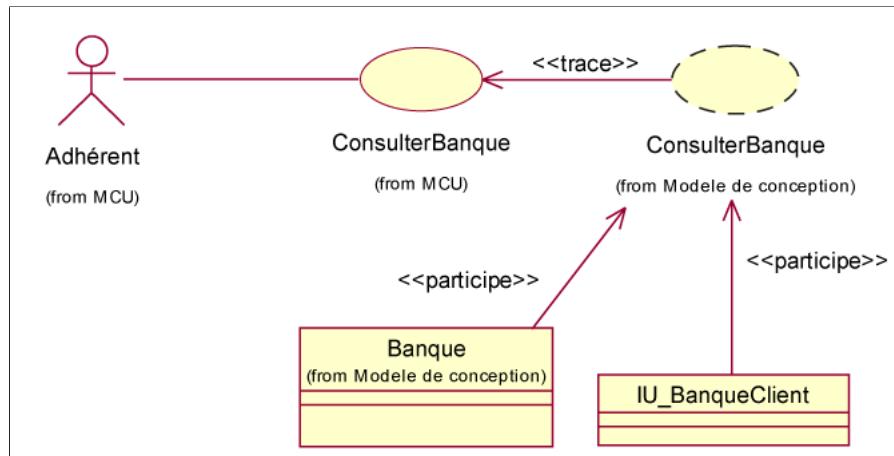


Figure 4.3: Traçabilité MCU/MC du cas d'utilisation «ConsulterBanque»

- Diagramme de séquence :

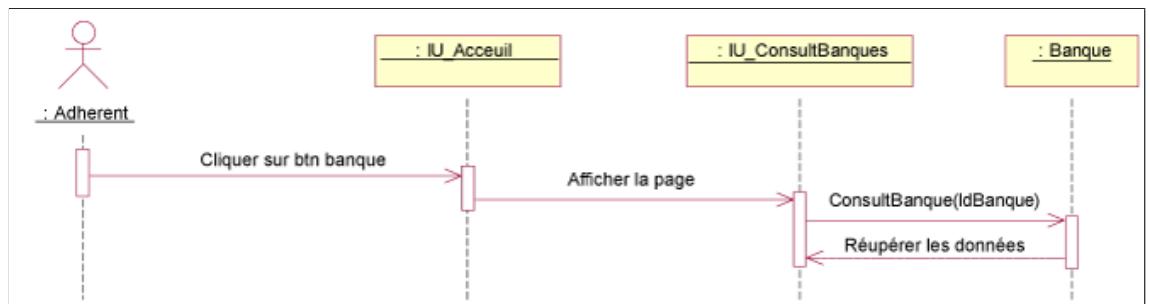


Figure 4.4: Diagramme de séquence du cas d'utilisation «ConsulterBanque»

C- La conception du cas d'utilisation «ConsulterAgence»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

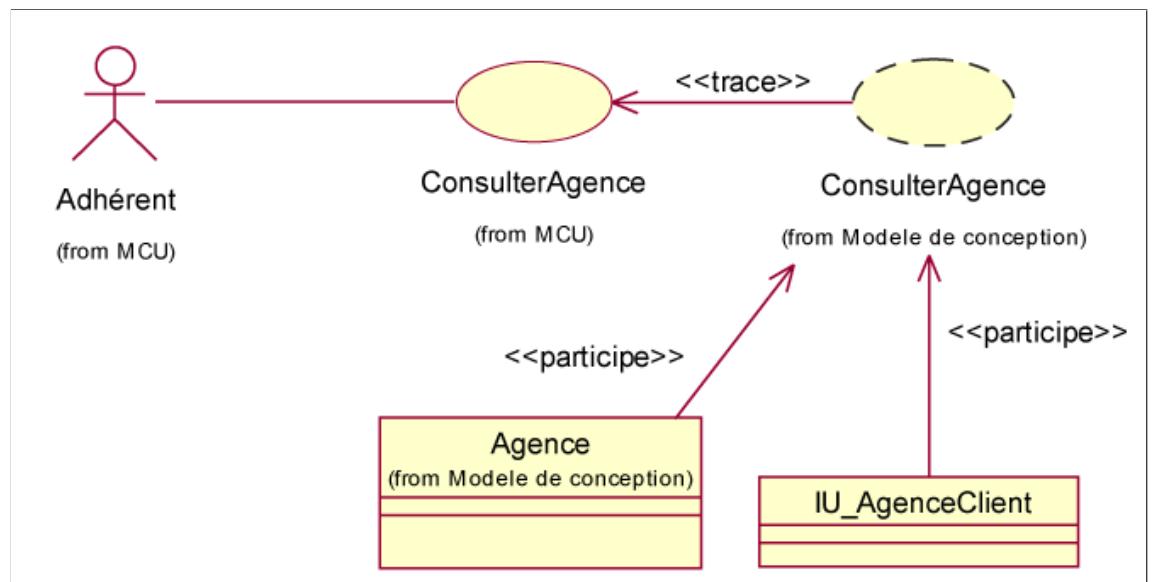


Figure 4.5: Traçabilité MCU/MC du cas d'utilisation «ConsulterAgence»

- Diagramme de séquence :

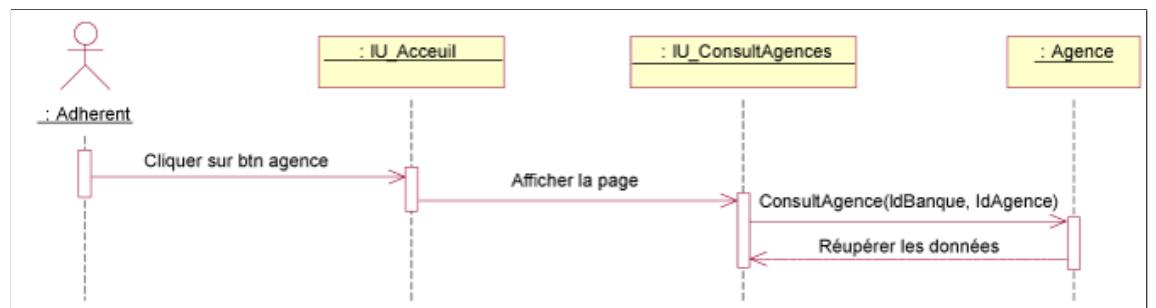


Figure 4.6: Diagramme de séquence du cas d'utilisation «ConsulterAgence»

D- La conception du cas d'utilisation «ConsulterDAB»

- Traçabilité entre le modèle de cas d'utilisation et le modèle de conception :

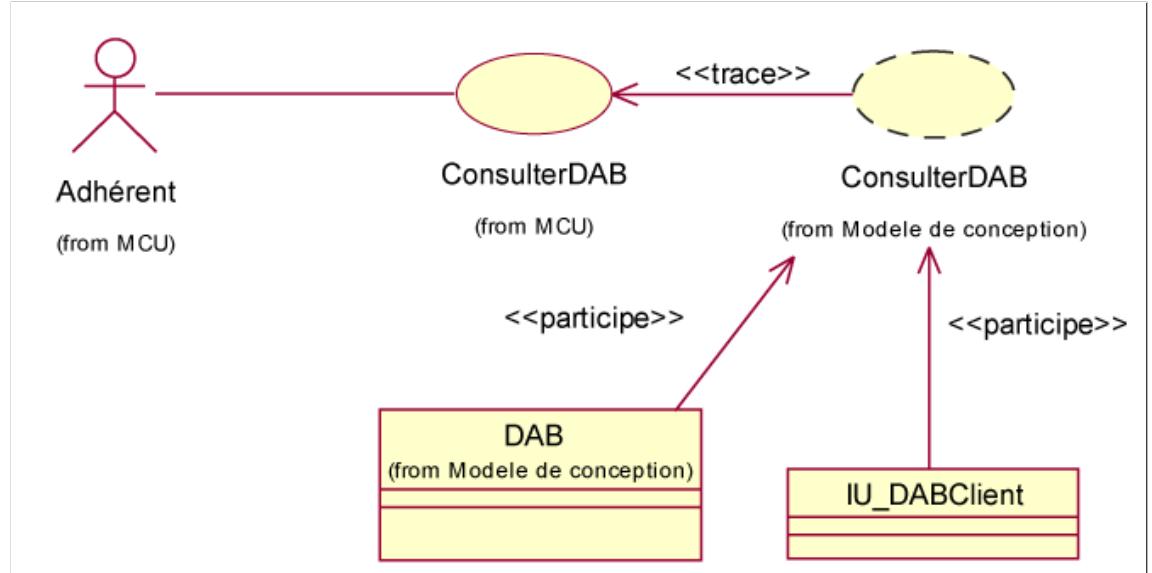


Figure 4.7: Traçabilité MCU/MC du cas d'utilisation «ConsulterDAB»

- Diagramme de séquence :

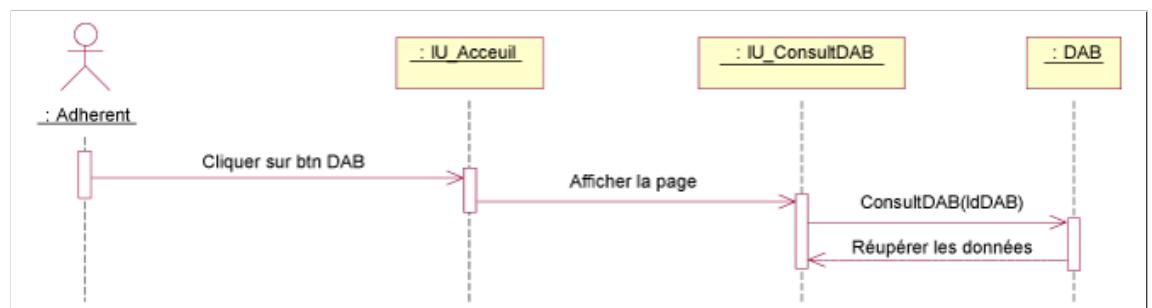


Figure 4.8: Diagramme de séquence du cas d'utilisation «ConsultDAB»

4.2.2 Conception des classes entités

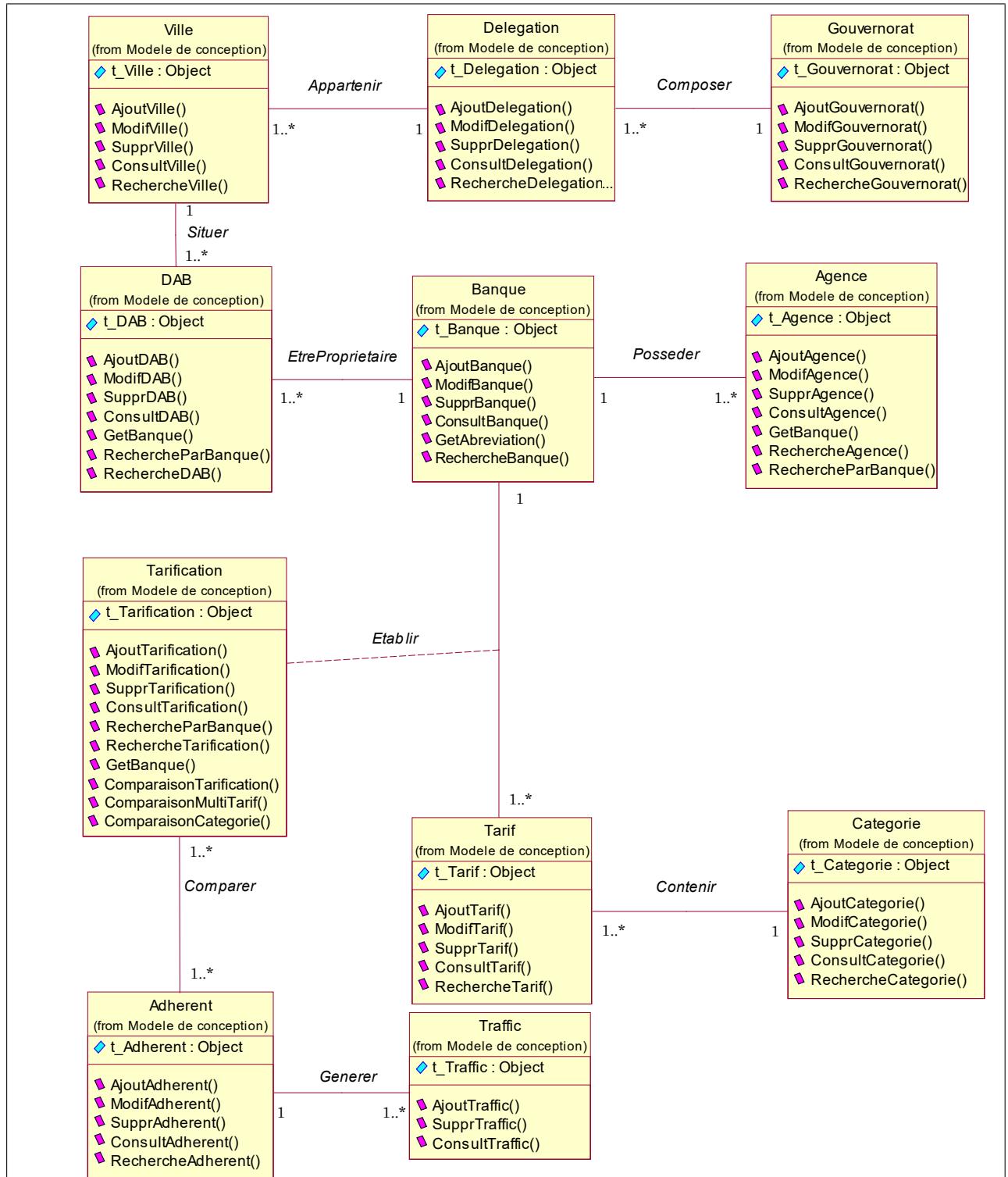


Figure 4.9: Diagramme de classes entités de l'application

4.2.3 Structure de la base de données

- Table Banque :

Description des rubriques et index d'un fichier de données
Tab Banque

Nombre de rubriques et index : 13 Taille en octets : 417 Afficher dans l'ordre physique

Cle	Nom	Libellé	Type	Taille
ID_Banque	ID Banque	Texte	2	
	Banque	Texte	30	
	Abreviation	Texte	10	
	Longitude	Numérique	6	
	Latitude	Numérique	6	
	Site_Web	Texte	50	
	Adresse	Texte	120	
	Tel	Texte	8	
	Lien_Tarifs	Texte	50	
	Email	Texte	50	
	Date_Creation	Date	8	
	Date_Modification	Date	8	
	ID_User	Texte	50	

Rubrique non reliée à un Metatype

Sous type : Chaîne

Clé : Sens de parcours

Clé unique Ascendant Descendant

Clé avec doublons

Paramètres de l'index et de la recherche pour clé texte

Sensible à la casse

Sensible à l'accentuation

Sensible aux espaces, ponctuation et car. spéciaux

Tableau Dimension : 2 Taille réelle

NULL Autorisé

Valeur par défaut : NULL

Valeur

Créer un Index full-text sur la rubrique

Figure 4.10: Table Banque

- Table Agence :

Description des rubriques et index d'un fichier de données
Tab Agence

Nombre de rubriques et index : 12 Taille en octets : 328 Afficher dans l'ordre physique

Cle	Nom	Libellé	Clé composée	Type	Taille
ID	ID	Texte	2		
ID_Banque	ID Banque	Texte	2		
ID_Agence	ID_Agence	Texte	3		
	Agence	Texte	50		
	Longitude	Numérique	6		
	Latitude	Numérique	6		
	Email	Texte	50		
	Adresse	Texte	120		
	Tel	Texte	8		
	Date_Creation	Date	8		
	Date_Modification	Date	8		
	ID_User	Texte	50		

Clé composée

Sous type : Clé composée

Clé : Non dé Ascendant Descendant

Clé unique

Clé avec doublons

Paramètres de l'index et de la recherche pour clé texte

Sensible à la casse

Sensible à l'accentuation

Sensible aux espaces, ponctuation et car. spéciaux

Tableau Dimension : 0 Taille réelle

Détail de la clé composée : ID_Banque+ID_Agence

Editer...

Figure 4.11: Table Agence

- Table DAB :

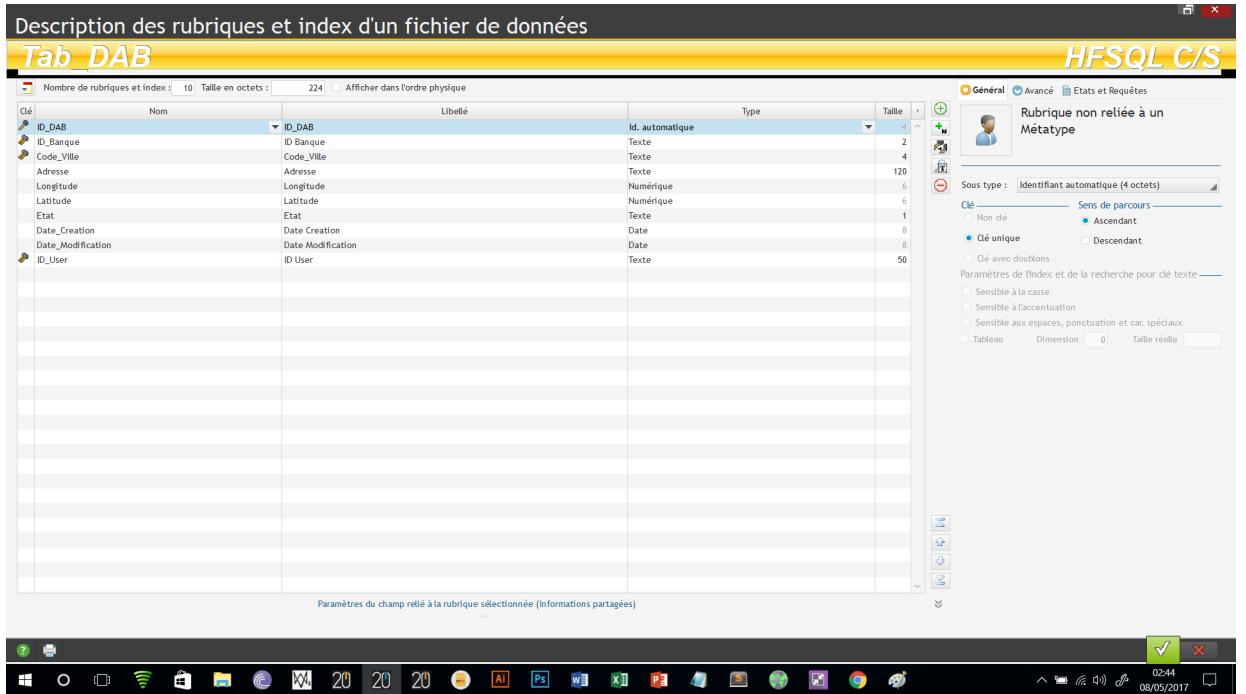


Figure 4.12: Table DAB

- Table Catégorie :

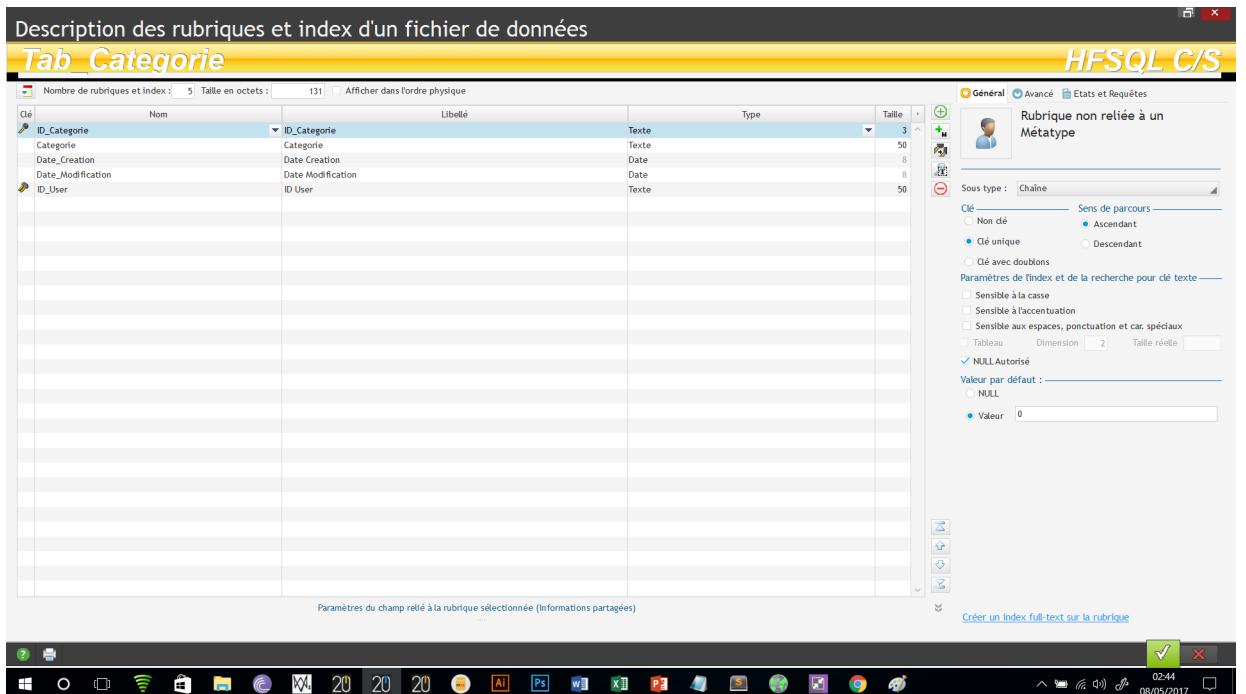


Figure 4.13: Table Catégorie

- Table Tarif :

Clé	Nom	Libellé	Type	Taille
ID_Tarif	Identifiant de Tarif	Id. automatique	Texte	50
ID_Categorie	ID_Categorie		Texte	3
Tarif	Tarif		Texte	50
Type_Tarif	Type_Tarif		Texte	2
Periodicite	Periodicite		Texte	2
Information	Information		Texte	8
assiette	assiette		Texte	20
Date_Creation	Date Creation		Date	8
Date_Modification	Date Modification		Date	8
ID_User	ID User		Texte	50

Figure 4.14: Table Tarif

- Table Tarification :

Clé	Nom	Libellé	Type	Taille
ID_Tarification	Identifiant de Tarif	Clé composée	Numérique	10
ID_Tarif	ID Banque		Texte	2
ID_Banque	Année		Texte	4
Annee	Valeur		Texte	20
Valeur	Min_Val		Monétaire	10
Min_Val	Max_Val		Monétaire	10
Max_Val	Date_Creation		Date	8
Date_Creation	Date_Modification		Date	8

Figure 4.15: Table Tarification

- Table Gouvernorat :

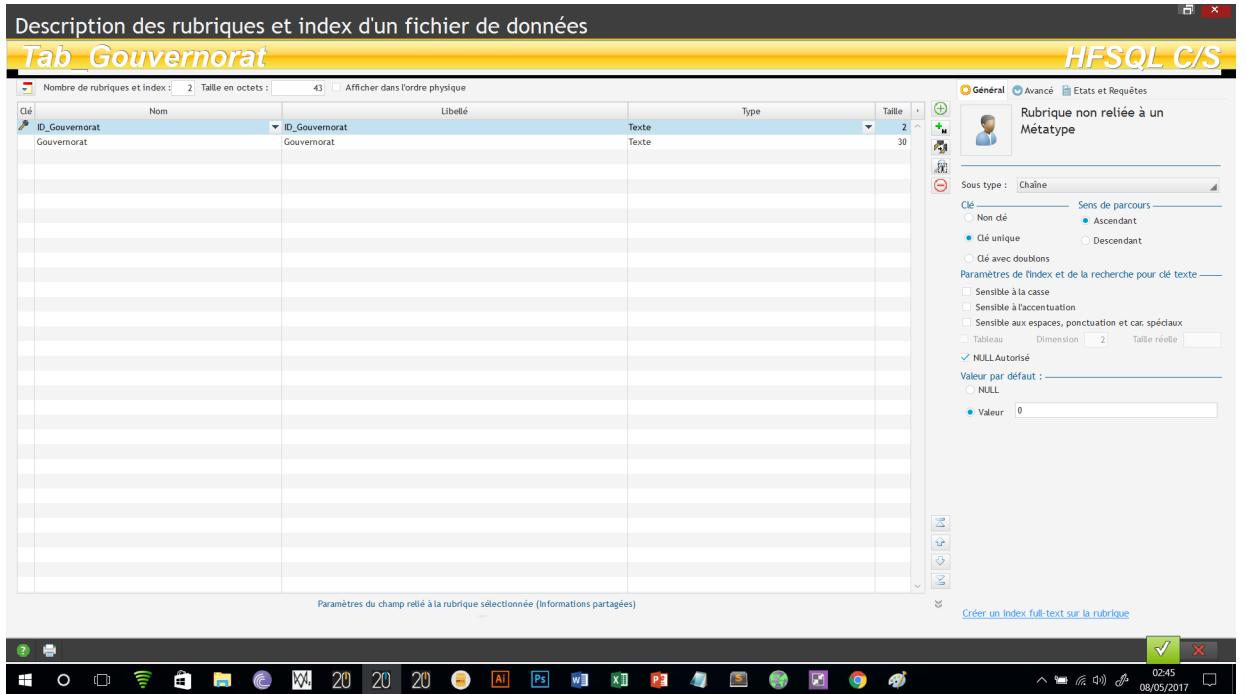


Figure 4.16: Table Gouvernorat

- Table Délégation :

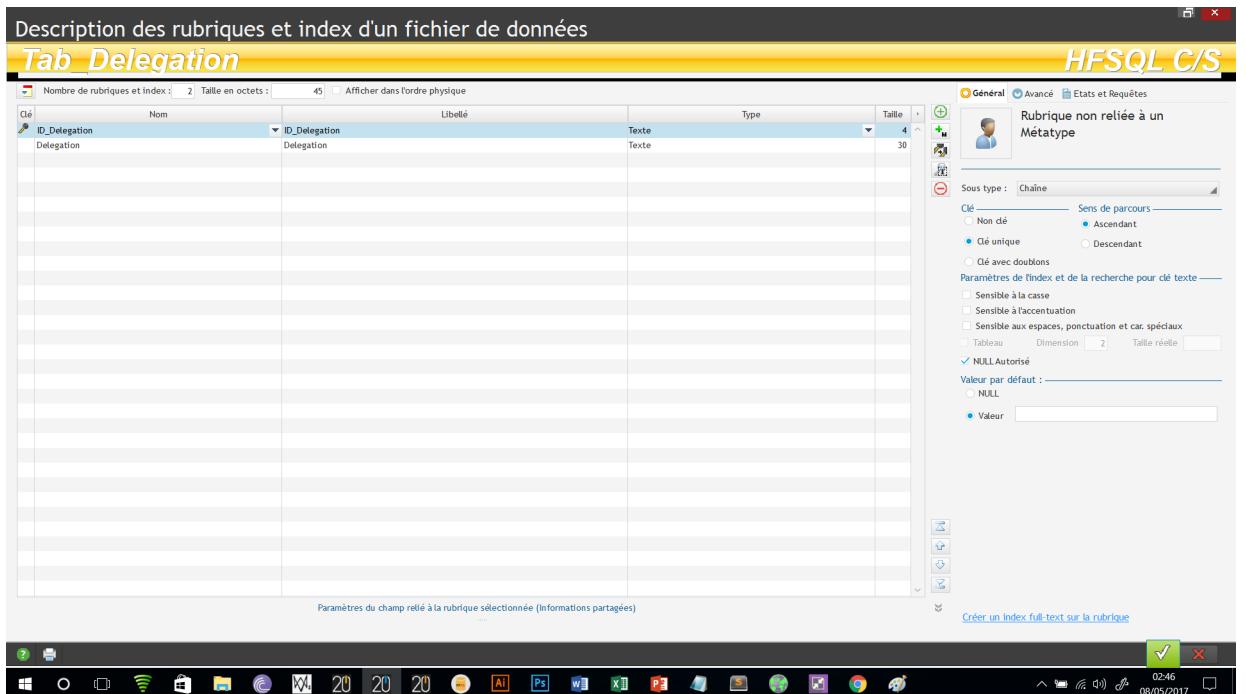


Figure 4.17: Table Délégation

- Table Ville :

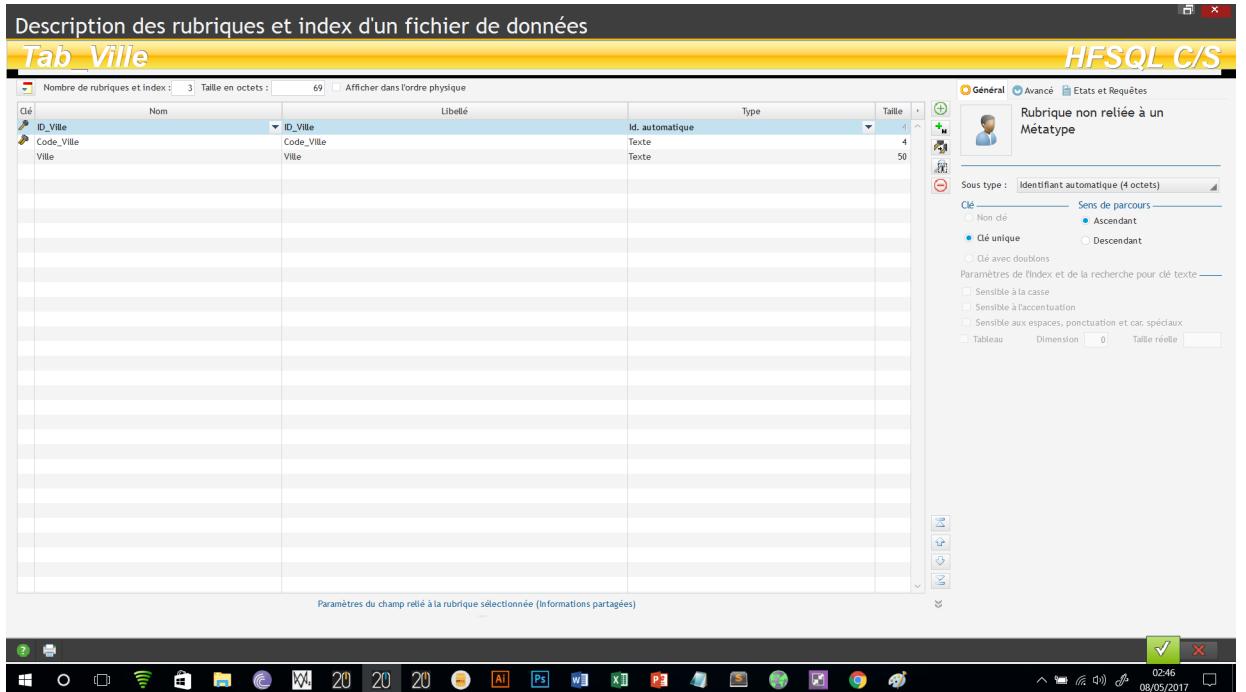


Figure 4.18: Table Ville

- Table User :

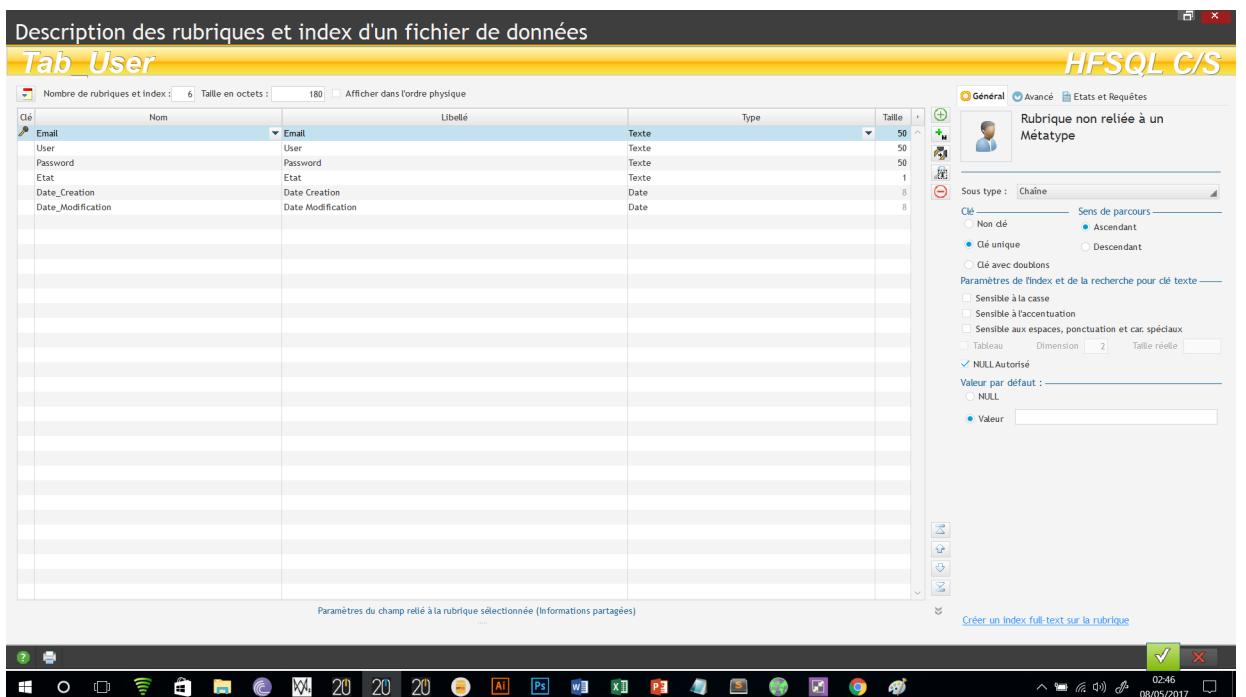


Figure 4.19: Table User

- Table Adherent :

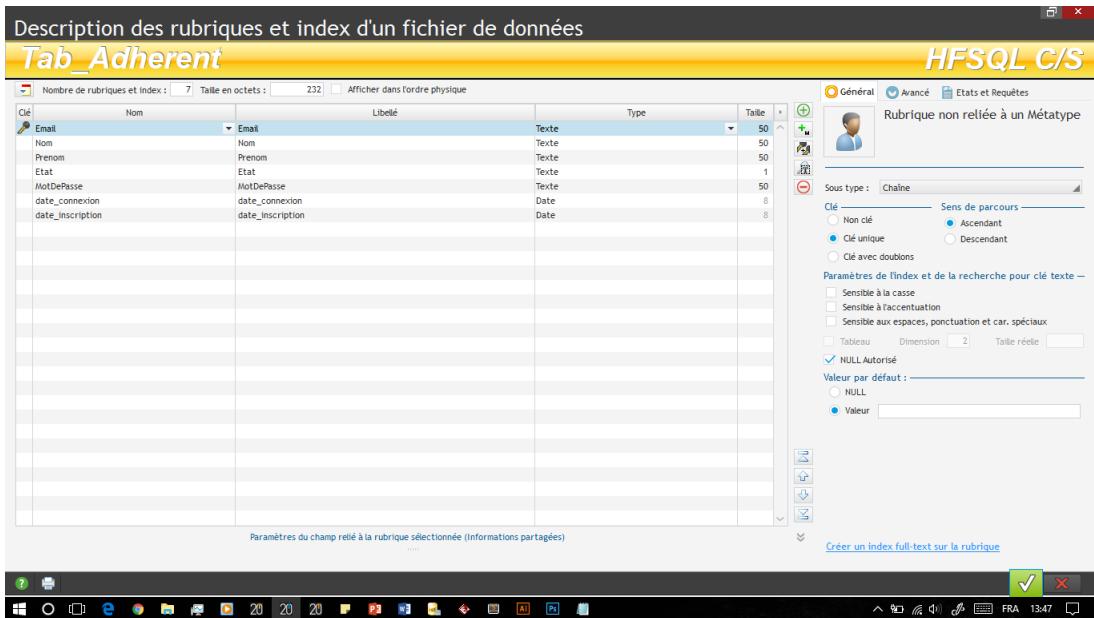


Figure 4.20: Table Adherent

- Table Traffic :

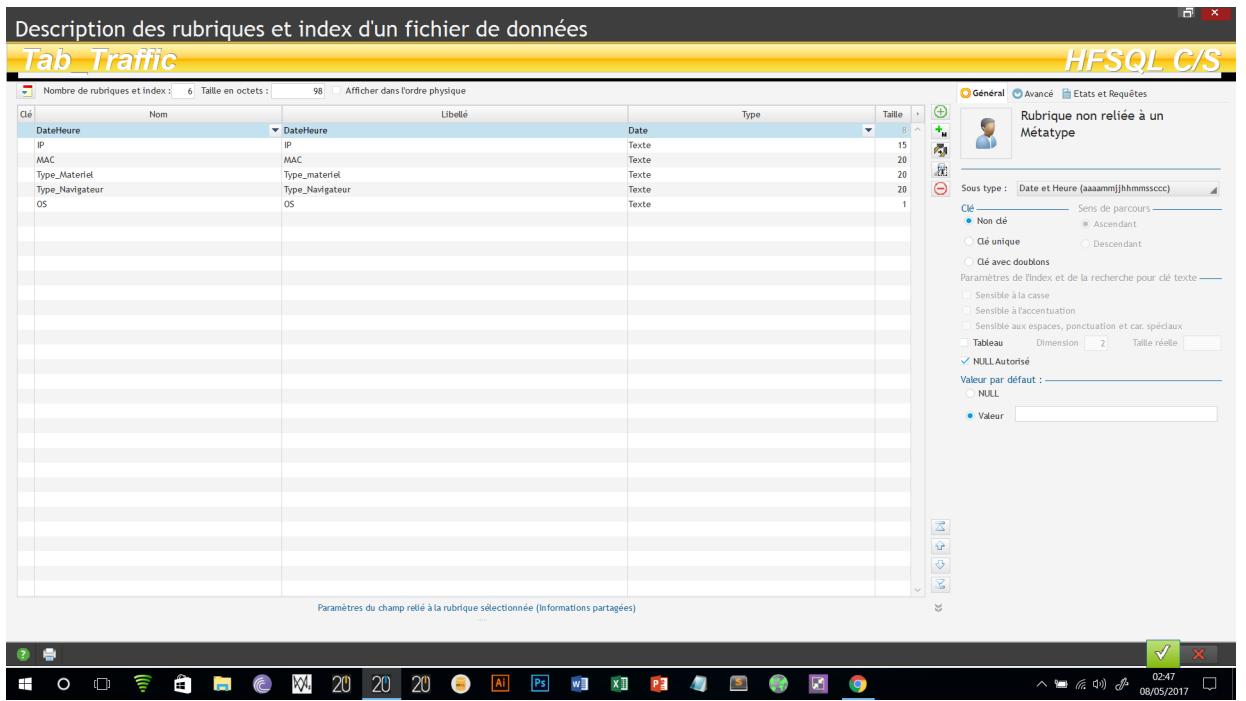


Figure 4.21: Table Traffic

4.3 Implémentation

4.3.1 Implémentation du cas d'utilisation «ConsulterTrafic»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

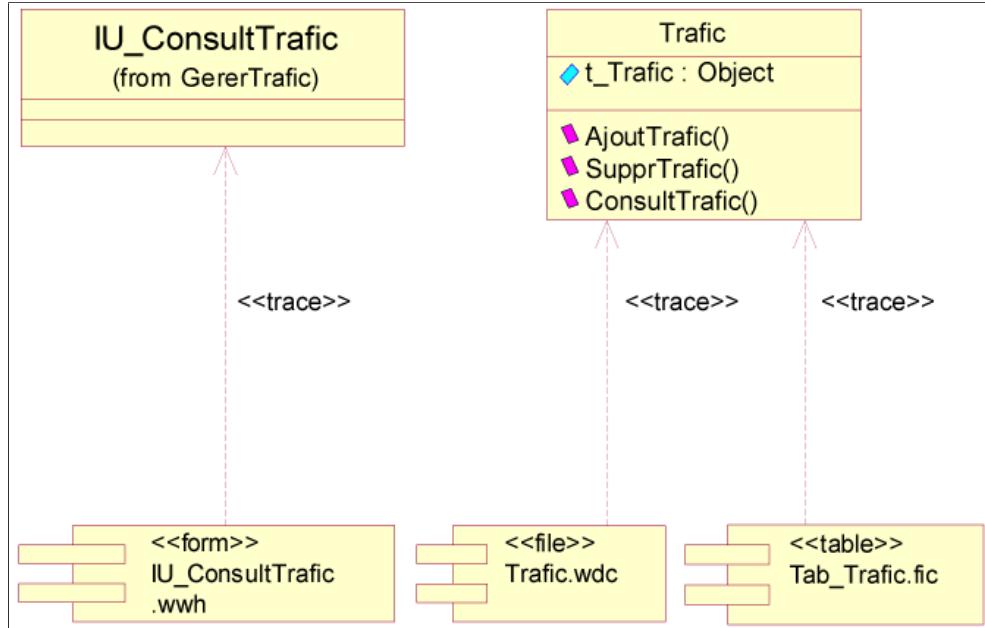


Figure 4.22: Traçabilité MC/MI du cas d'utilisation «ConsulterTrafic»

B- Diagramme de composant

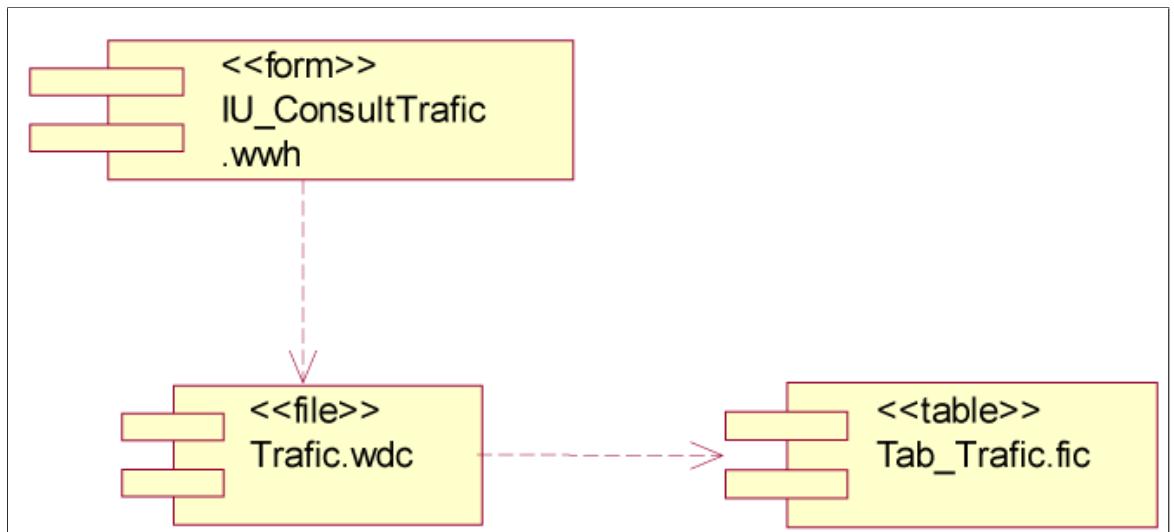


Figure 4.23: Diagramme de composant du cas d'utilisation «ConsulterTrafic»

4.3.2 Implémentation du cas d'utilisation «ConsulterBanque»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

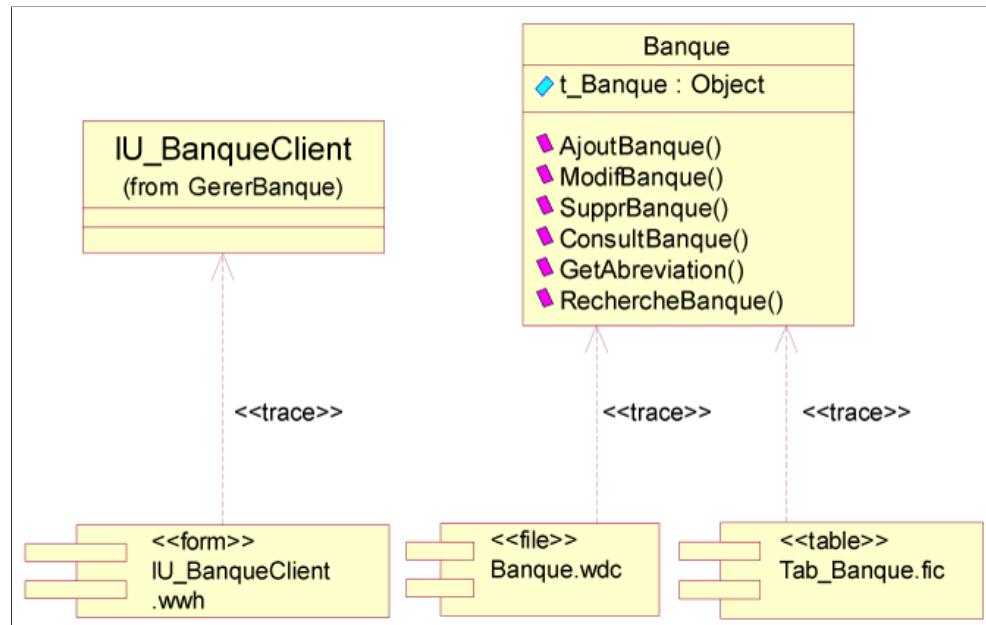


Figure 4.24: Traçabilité MC/MI du cas d'utilisation «ConsulterBanque»

B- Diagramme de composant

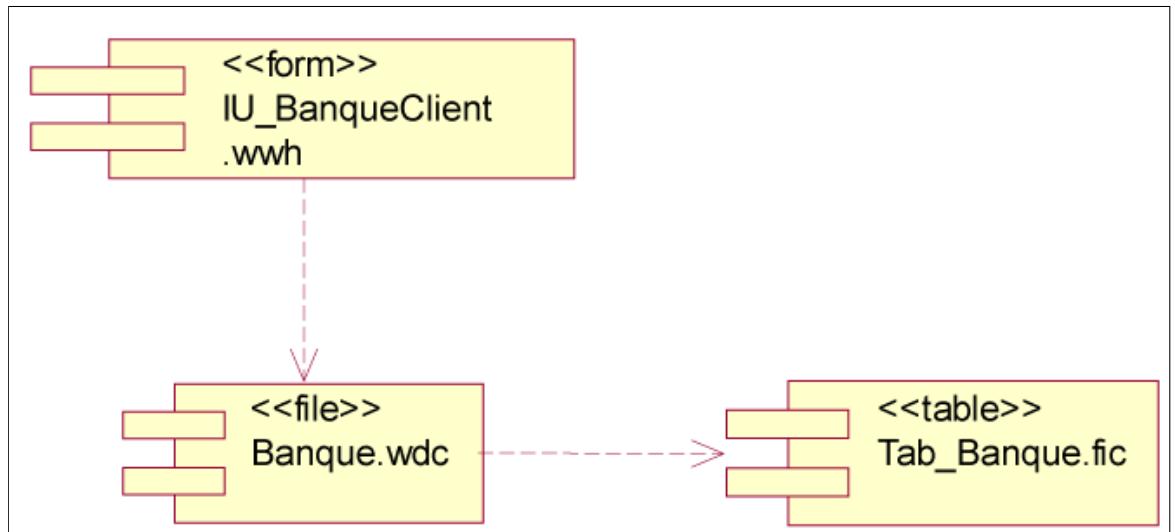


Figure 4.25: Diagramme de composant du cas d'utilisation «ConsulterBanque»

4.3.3 Implémentation du cas d'utilisation «ConsulterAgence»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

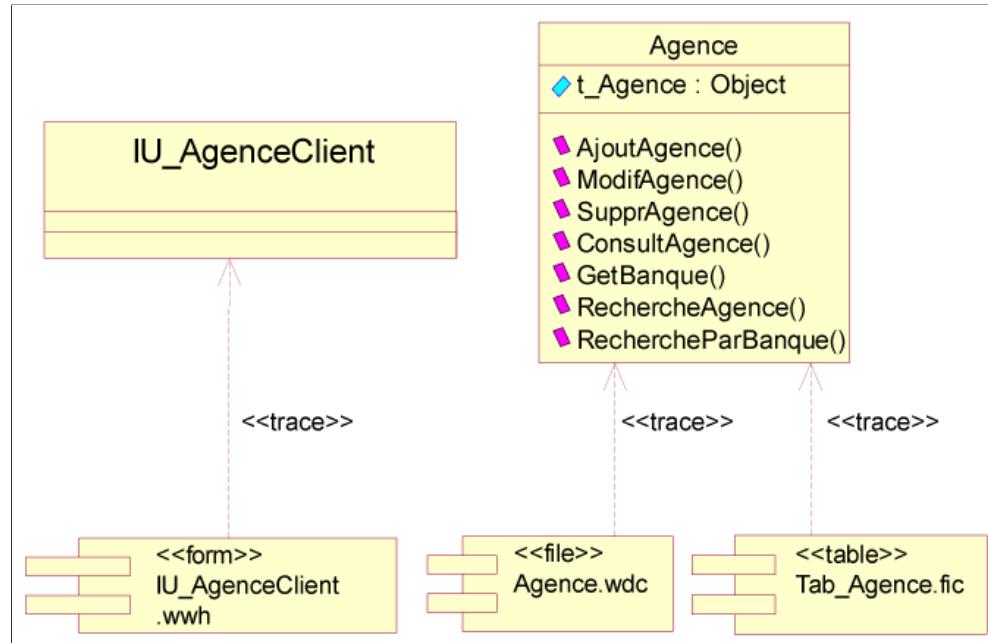


Figure 4.26: Traçabilité MC/MI du cas d'utilisation «ConsulterAgence»

B- Diagramme de composant

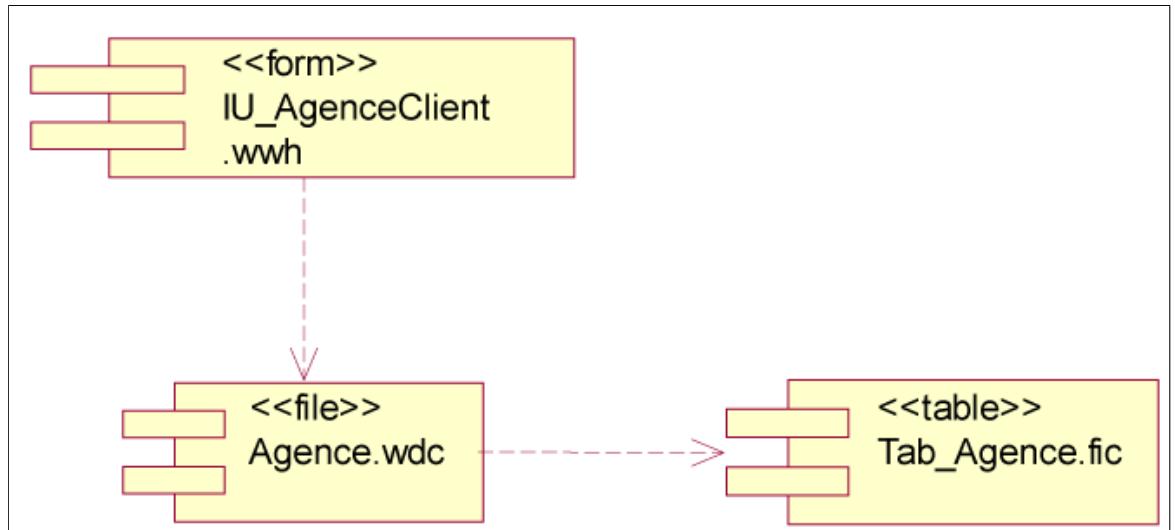


Figure 4.27: Diagramme de composant du cas d'utilisation
«ConsulterBanque»

4.3.4 Implémentation du cas d'utilisation «ConsulterDAB»

A- Traçabilité entre le modèle de conception (MC) et le modèle d'implémentation (MI)

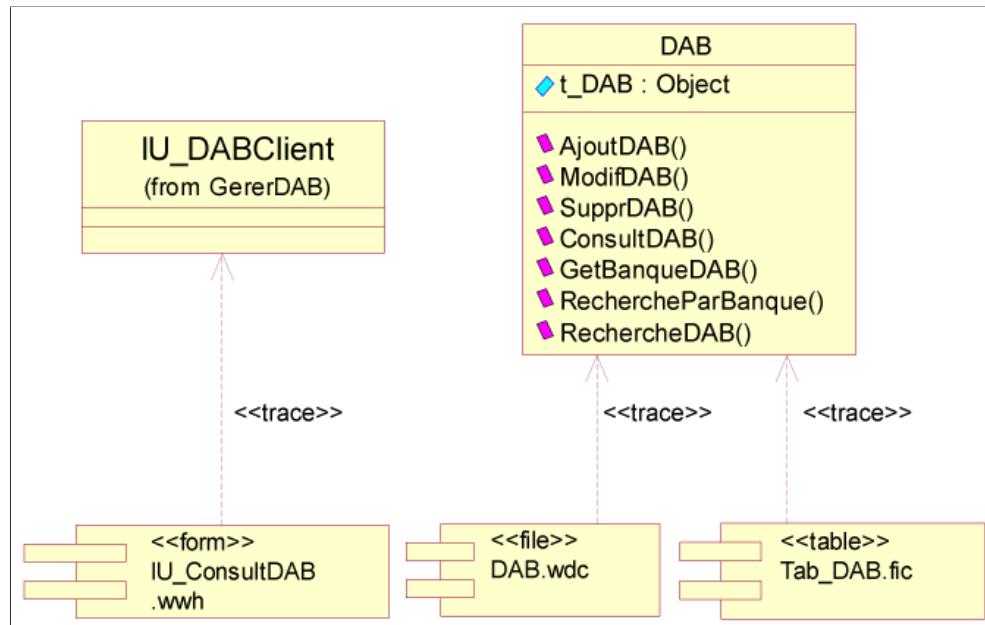


Figure 4.28: Traçabilité MC/MI du cas d'utilisation «ConsulterDAB»

B- Diagramme de composant

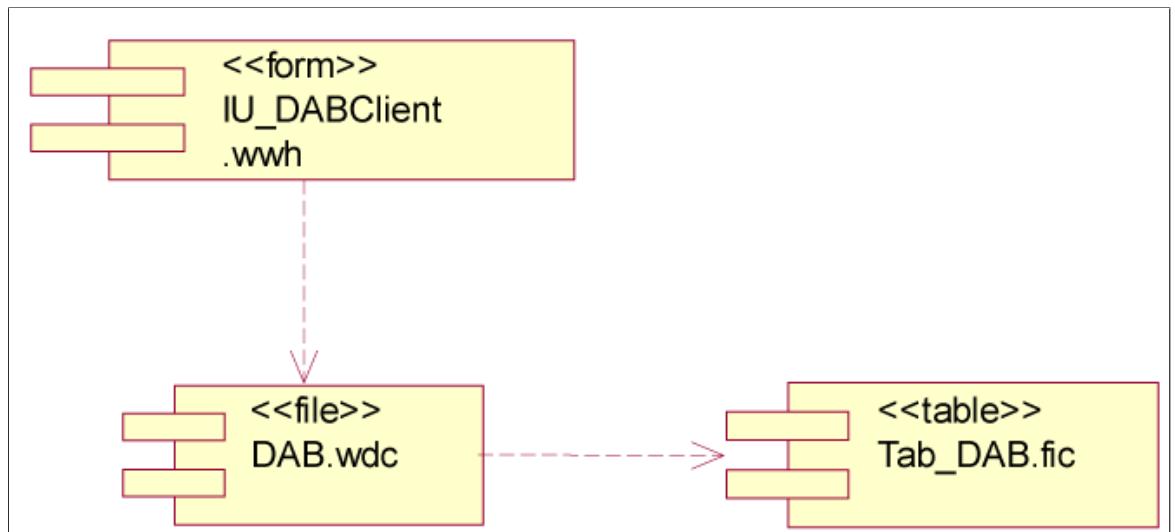


Figure 4.29: Diagramme de composant du cas d'utilisation «ConsulterDAB»

4.4 Diagramme de déploiement

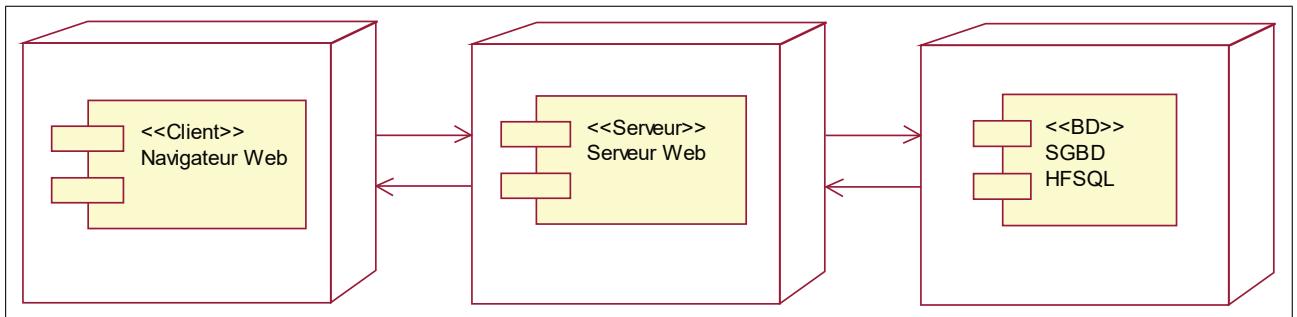


Figure 4.30: Diagramme de déploiement du système

Conclusion

A la fin de cette phase, nous avons achevé le raffinement, la conception et l'implémentation de tous les cas d'utilisation. Ceci nous permet d'obtenir comme artefact une base de données bien structurée et principalement, une architecture de référence stabilisée qui se transforme en produit exécutable.

5 La phase de transition

Introduction

En premier lieu, nous allons présenter l'architecture mise en place du système, les outils matériels et logiciels ainsi que les différents langages de programmation utilisés lors du développement. En deuxième lieu, nous dévoilons les interfaces graphiques de notre application accompagnées d'une description de son fonctionnement.

5.1 Architecture de l'application

Notre système suit l'architecture trois-tiers. Elle est composée de 3 couches:

- La couche présentation : correspond à la partie de l'application visible et interactive avec les utilisateurs.
- La couche logique métier : représente la partie qui implémente la logique de l'application.
- La couche de données: présente la partie gérante l'accès aux données du système.

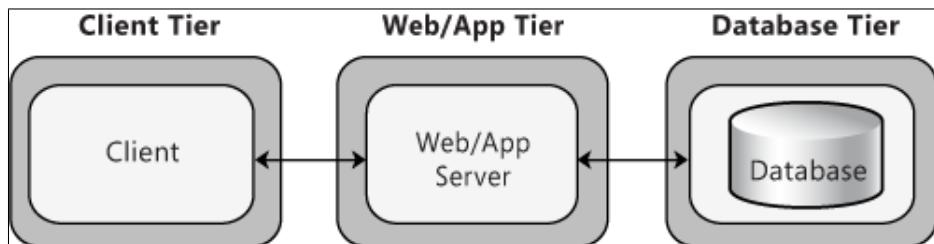


Figure 5.1: Architecture du système (3-tiers)

Parmi les avantages de cette architecture, on peut citer:

- Allègement du poste de travail client
- Prise en compte de l'hétérogénéité des plates-formes
- Rupture du lien de propriété exclusive entre application et données

5.2 Environnement de développement

Pour la construction de l'application, nous avons recours aux environnements suivants :

5.2.1 Les outils matériels

Le développement de l'application est réalisé par le biais de deux ordinateurs portables ayant les caractéristiques suivantes:

Caractéristiques	MSI GP62 6QE Leopard Pro	ASUS GL553VW-STRIX
Marque	MSI	Asus
Processeur	Intel Core i7-6700HQ 2.60GHz	Intel Core i5-6300HQ 2.30 GHz
RAM	8Go	12Go
Disque dur	1To	1To

Table 5.1: Matériels de base

5.2.2 Les outils logiciels



WEBDEV est un AGL Apparenté à WinDev et WinDev Mobile, édité par la société française PC SOFT. Il permet de créer des sites Web et des sites Intranet statiques, mais surtout dynamiques, c'est-à-dire reliés à des bases de données. WebDev utilise le langage WLanguage pour le code côté serveur (préparation du contenu des pages) comme pour le code côté navigateur (interactions avec l'utilisateur). Il est également possible d'utiliser directement SQL, PHP, HTML/XHTML, CSS et Javascript. Il prend également en charge de manière intégrée AJAX à partir de la version 10. [4]



HFSQ est un puissant SGBDR intégré aux environnements de développement WINDEV, WINDEV Mobile et WEBDEV. [5]



Rational Rose est un outil de modélisation d'application édité par la société IBM. Il permet entre autre la réalisation de diagrammes UML. [6]



Adobe Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur édité par Adobe. [7]



Adobe Illustrator Adobe Illustrator est un logiciel de création graphique vectorielle. Il fait partie de la gamme Adobe et il offre des outils de dessin vectoriel puissants. [8]

5.3 Les interfaces graphiques de l'application

5.3.1 S'authentifier admin

L'administrateur ou l'adhérent saisit son email et son mot de passe, puis il clique sur le bouton «Connexion», ensuite le système vérifie la combinaison email et mot de passe pour vérifier s'ils sont corrects, enfin le système affiche l'interface du menu principal approprié.

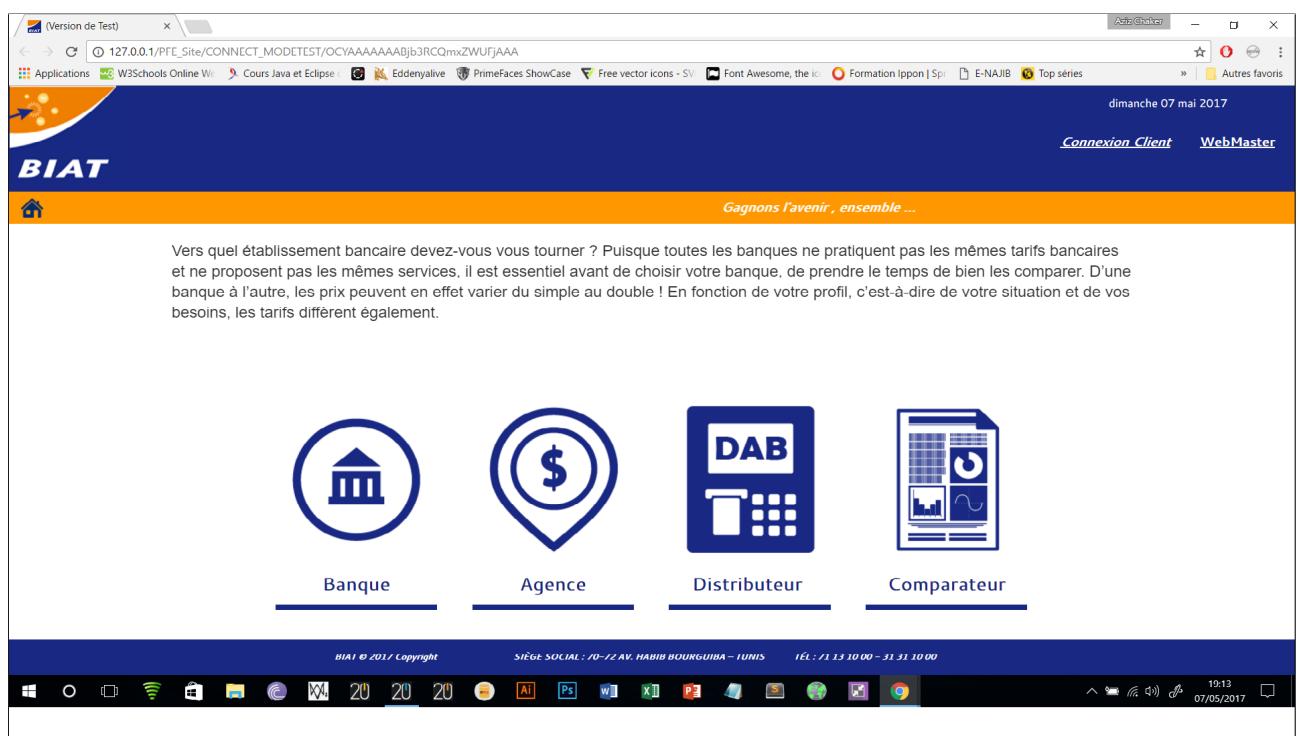


Figure 5.2: Interface «Accueil»

CHAPTER 5. LA PHASE DE TRANSITION

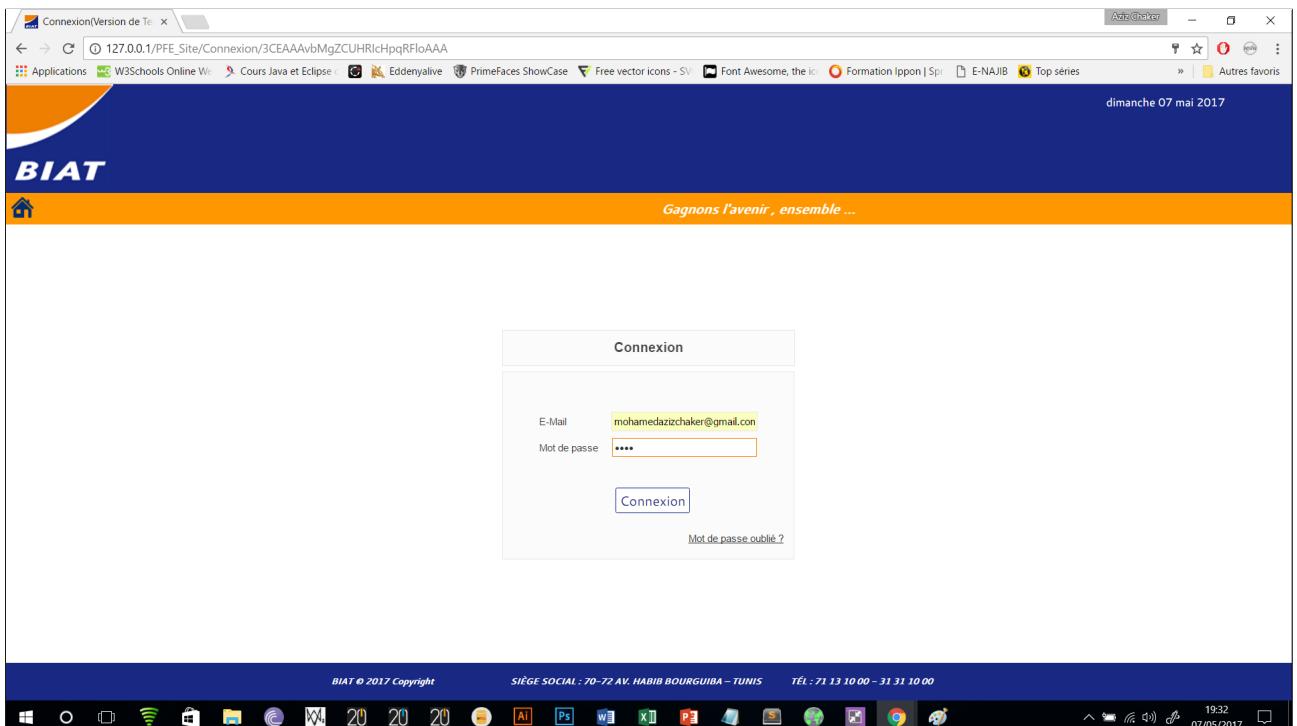


Figure 5.3: Interface «Connexion Administrateur»

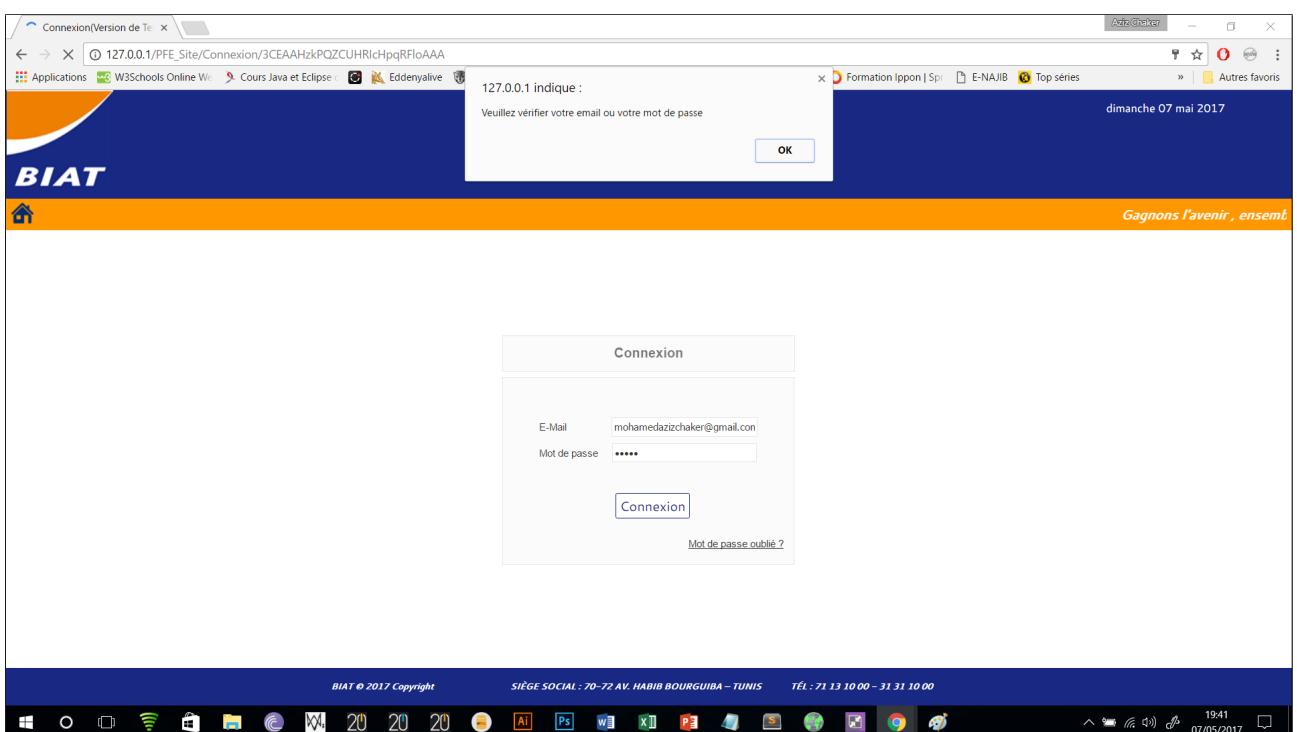


Figure 5.4: Interface «Erreur connexion administrateur»

CHAPTER 5. LA PHASE DE TRANSITION

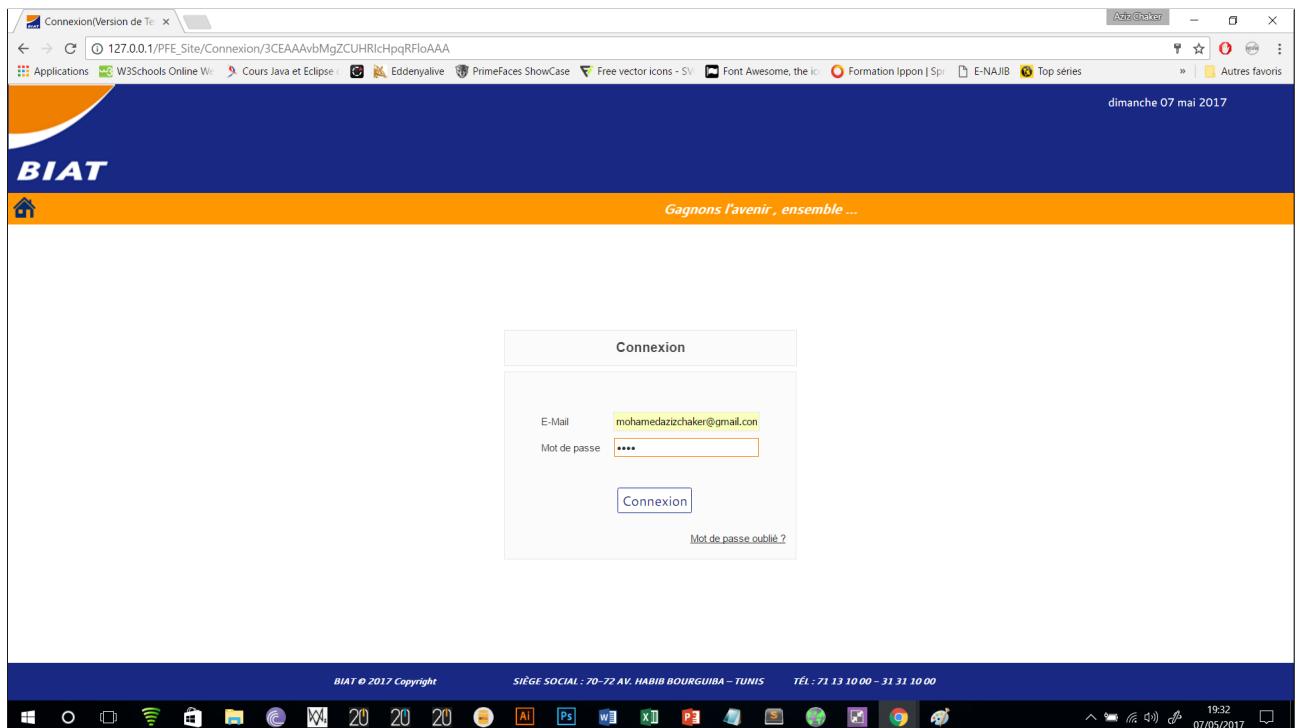


Figure 5.5: Interface «Connexion avec mot de passe correct»

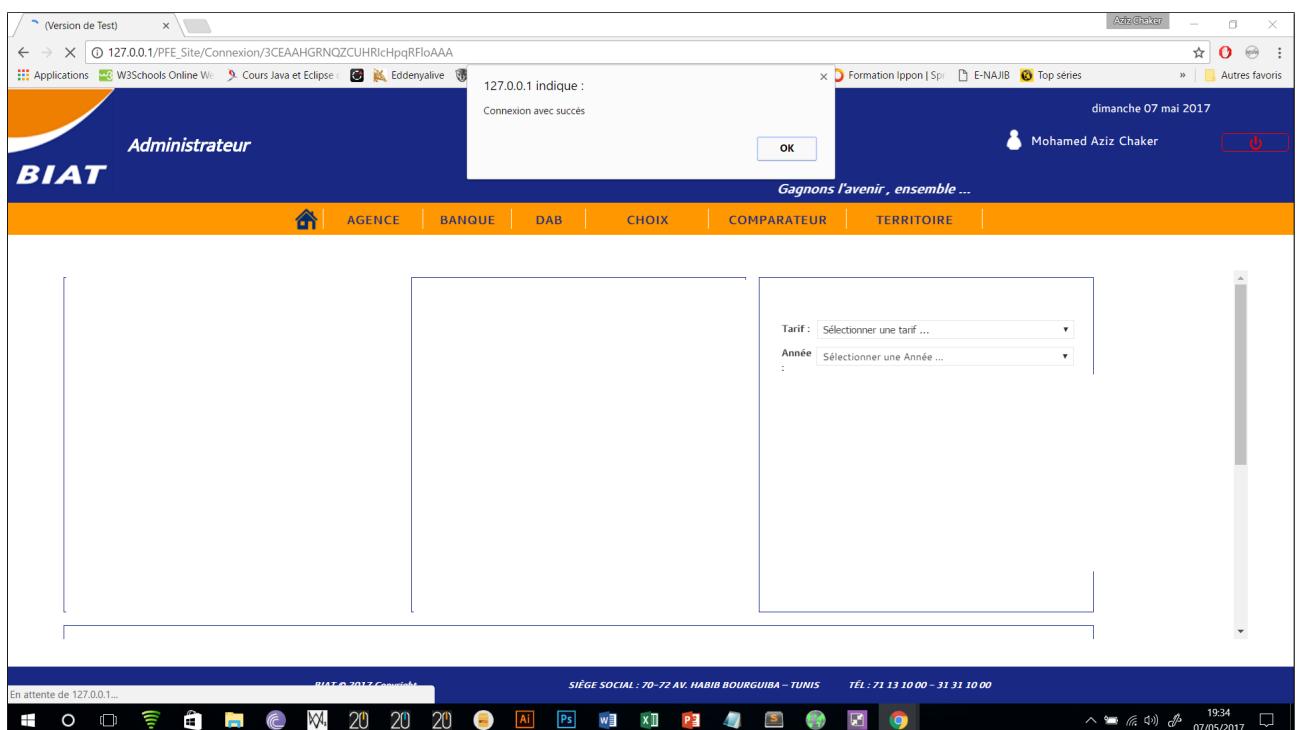


Figure 5.6: Interface «Connexion avec succès»

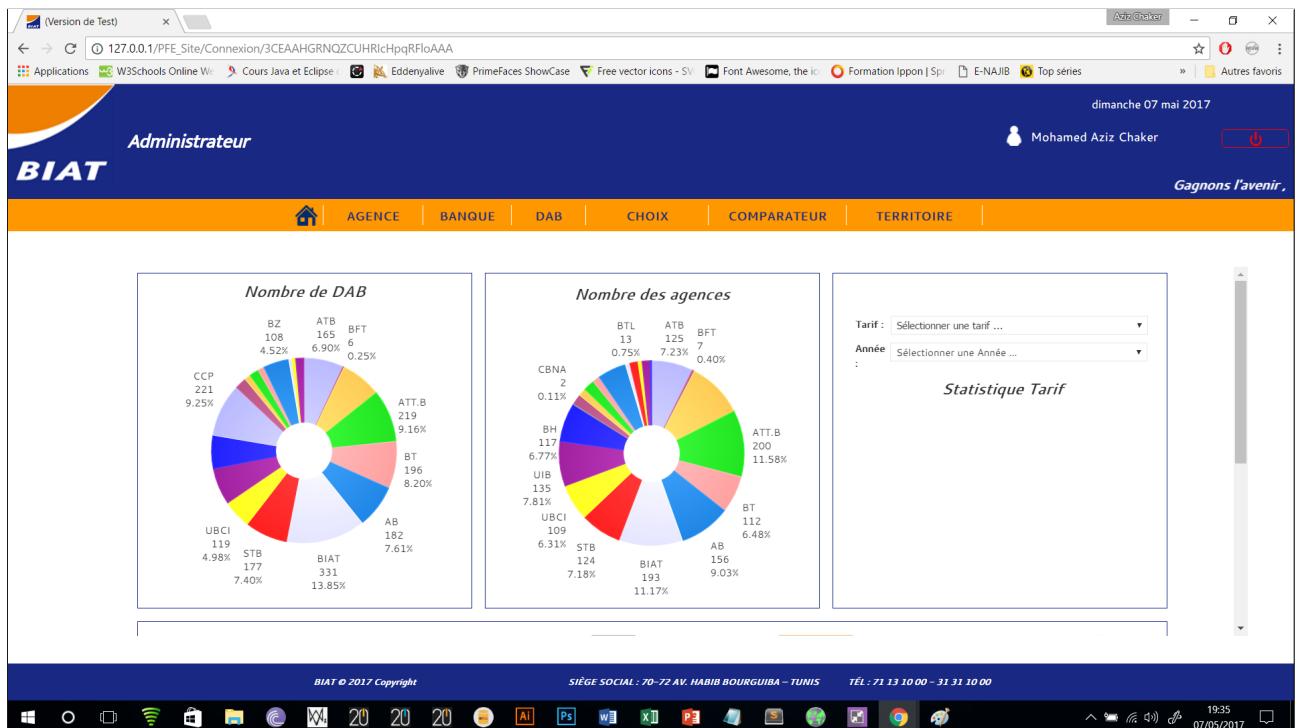


Figure 5.7: Interface «Tableau de bord»

5.3.2 Réinitialiser mot de passe

Lorsque l'administrateur oublie son mot de passe, Il clique sur le lien «Mot de Passe oublié ?», puis l'administrateur se trouve dans l'interface «Réinitialiser mot de passe» , il saisit son email et le captcha valides et clique sur le bouton envoyer pour recevoir un mot de passe temporel dans sa boite mail. Par la suite ce dernier se reconnecte avec le mot de passe temporel, de ce fait il se trouve obligatoirement dans l'interface «Changer mot de passe» d'où il saisit le mot de passe temporel, un nouveau mot de passe différent du temporel et le captcha puis il clique sur le bouton envoyer. Enfin, l'administrateur doit se reconnecter avec son nouveau mot de passe.

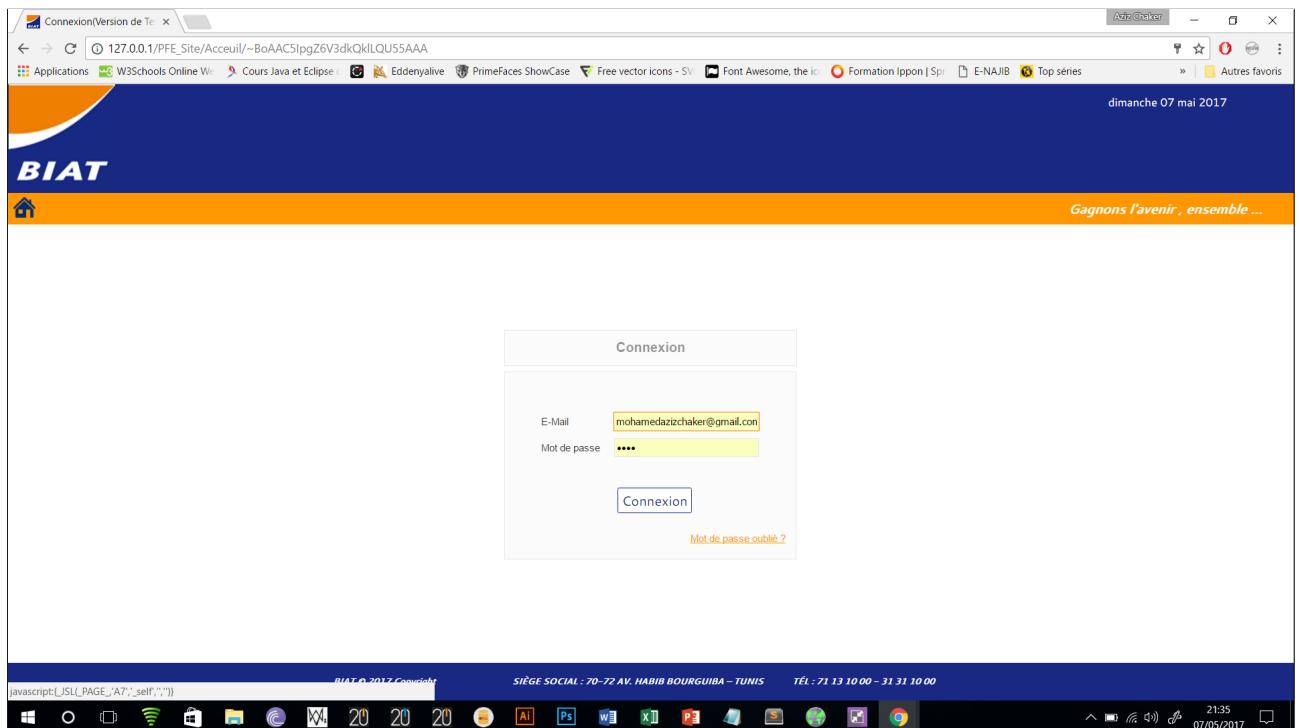


Figure 5.8: Interface «Mot de passe oublié»

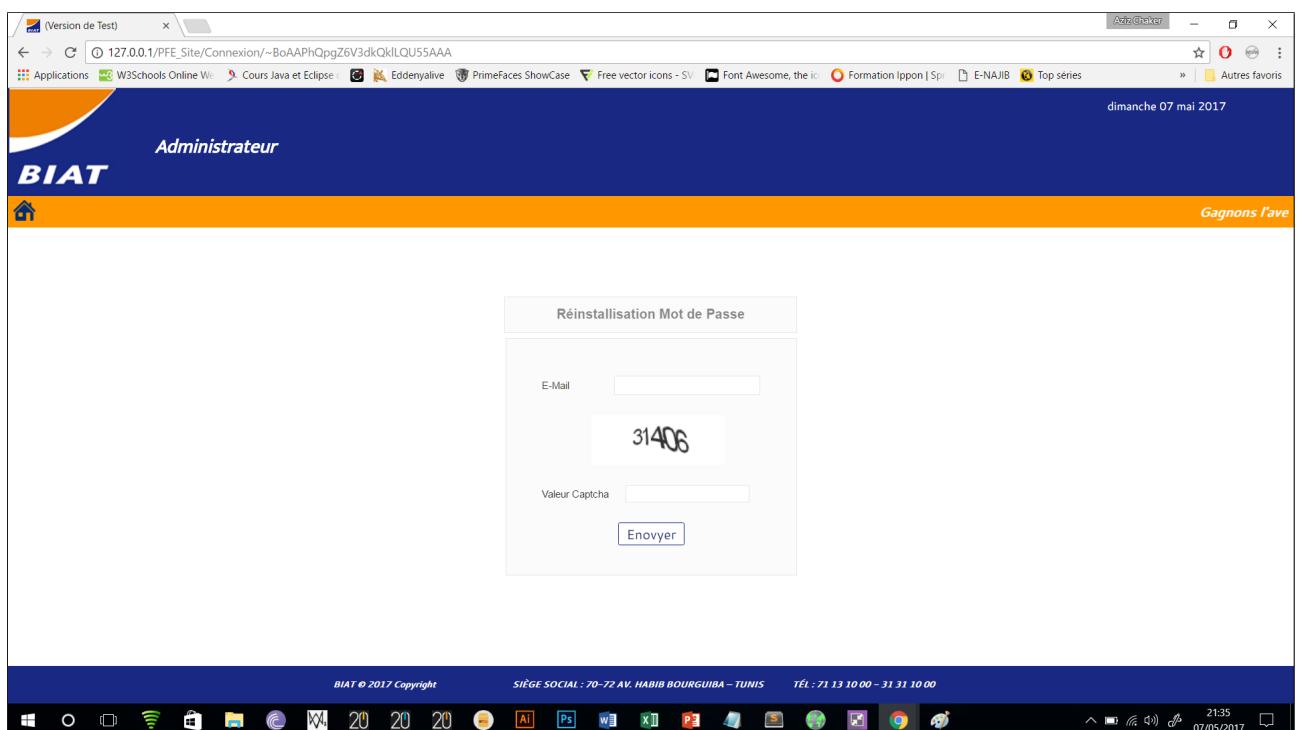


Figure 5.9: Interface «Réinitialiser mot de passe»

CHAPTER 5. LA PHASE DE TRANSITION

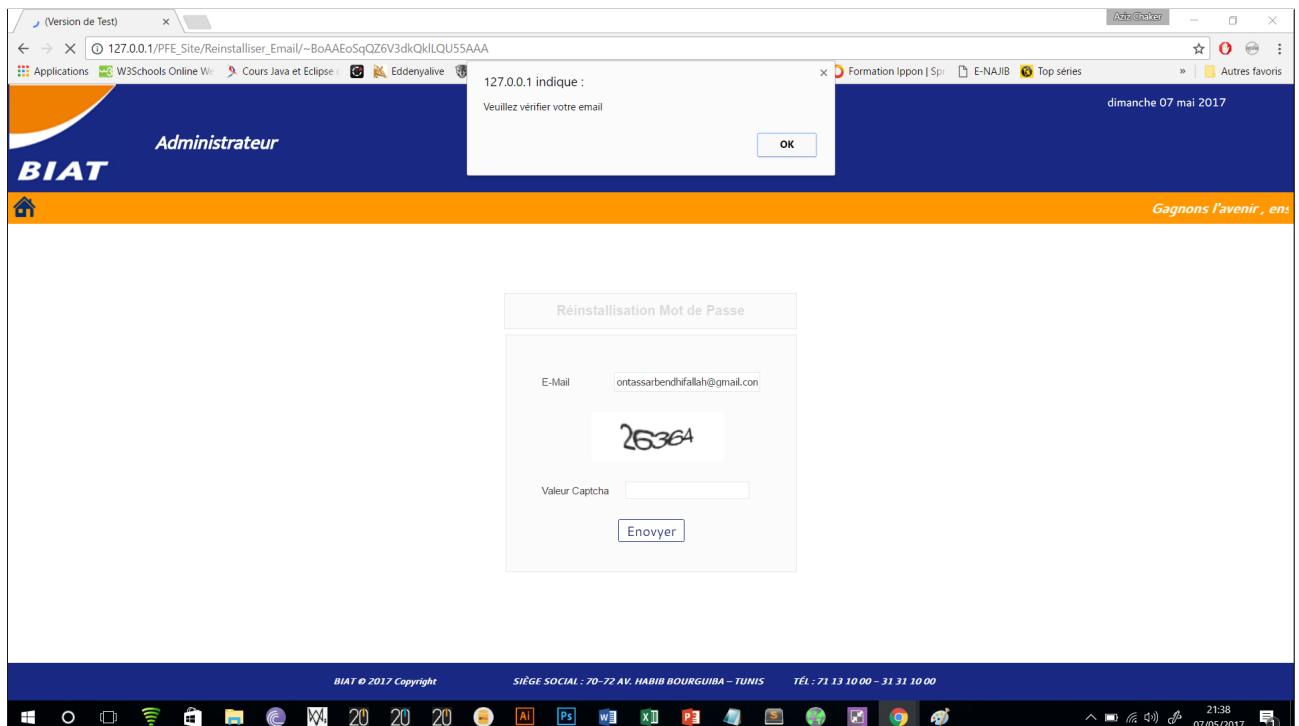


Figure 5.10: Interface «Vérifier email»

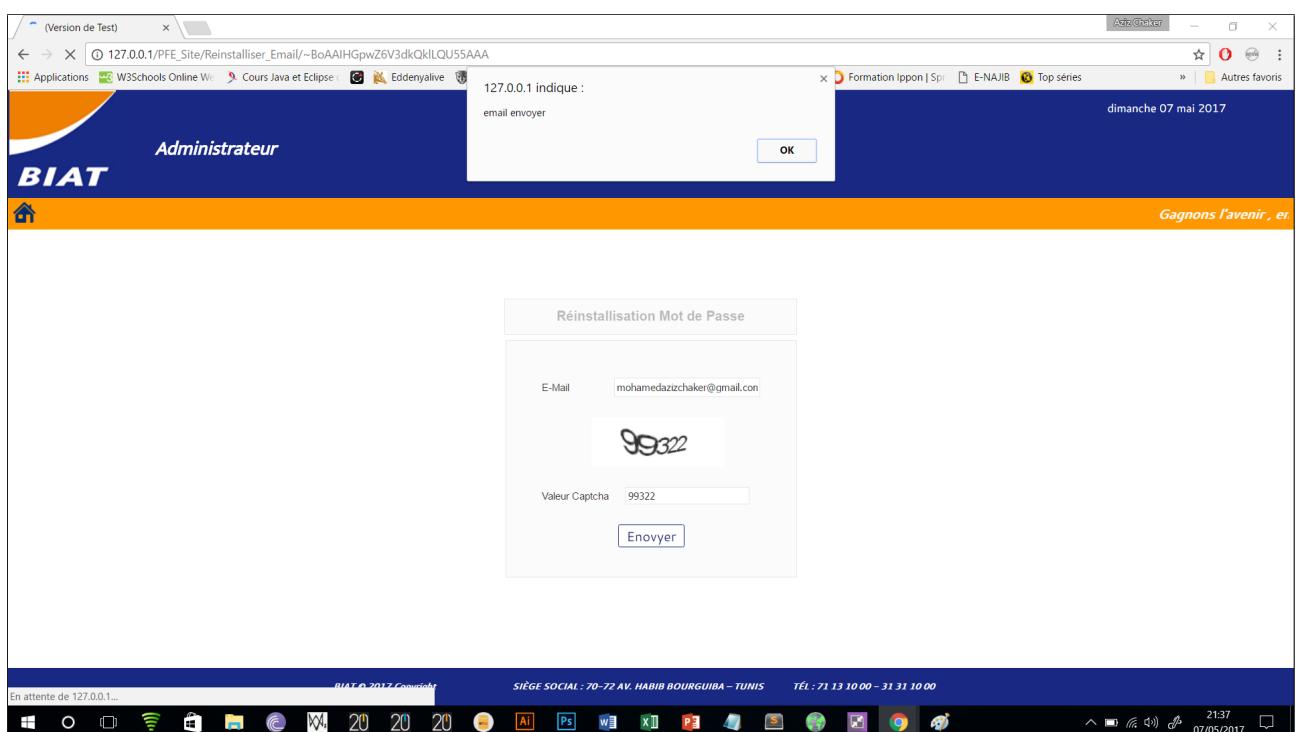


Figure 5.11: Interface «Succès d'envoi»

CHAPTER 5. LA PHASE DE TRANSITION

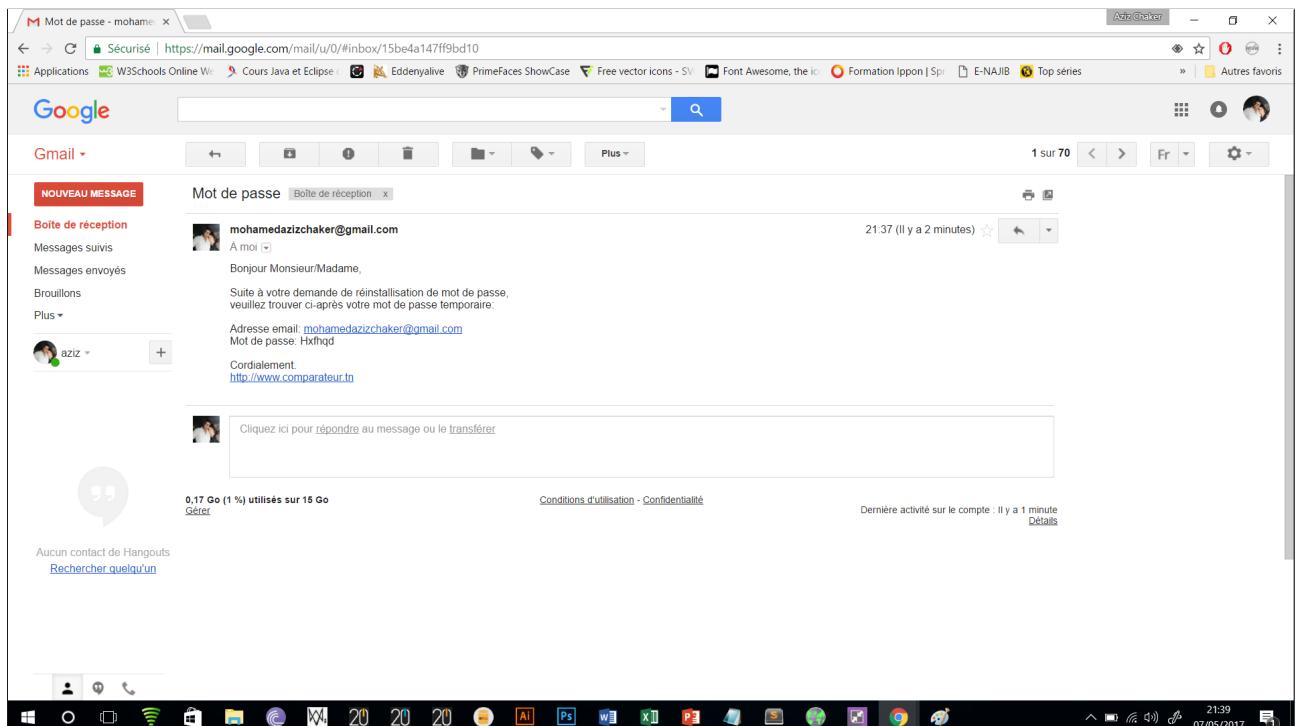


Figure 5.12: Interface «Réception de l'email»

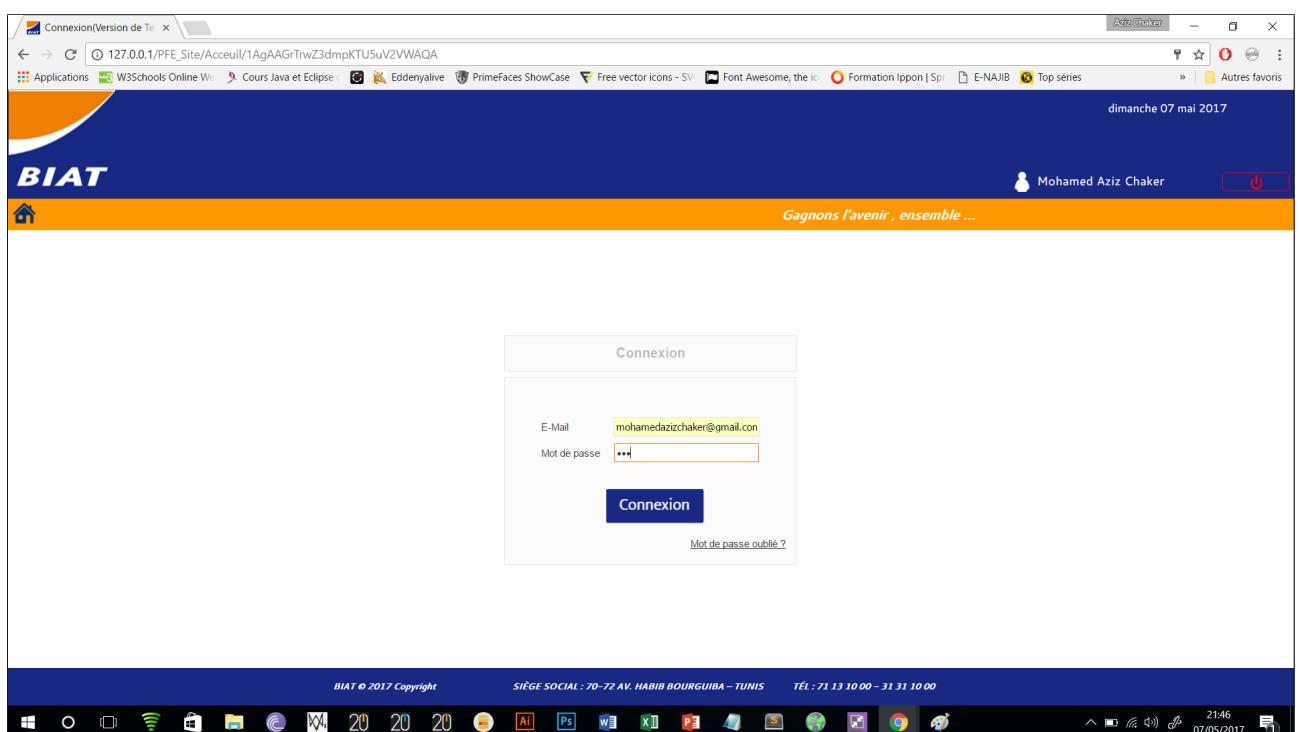


Figure 5.13: Interface «Reconnexion»

CHAPTER 5. LA PHASE DE TRANSITION

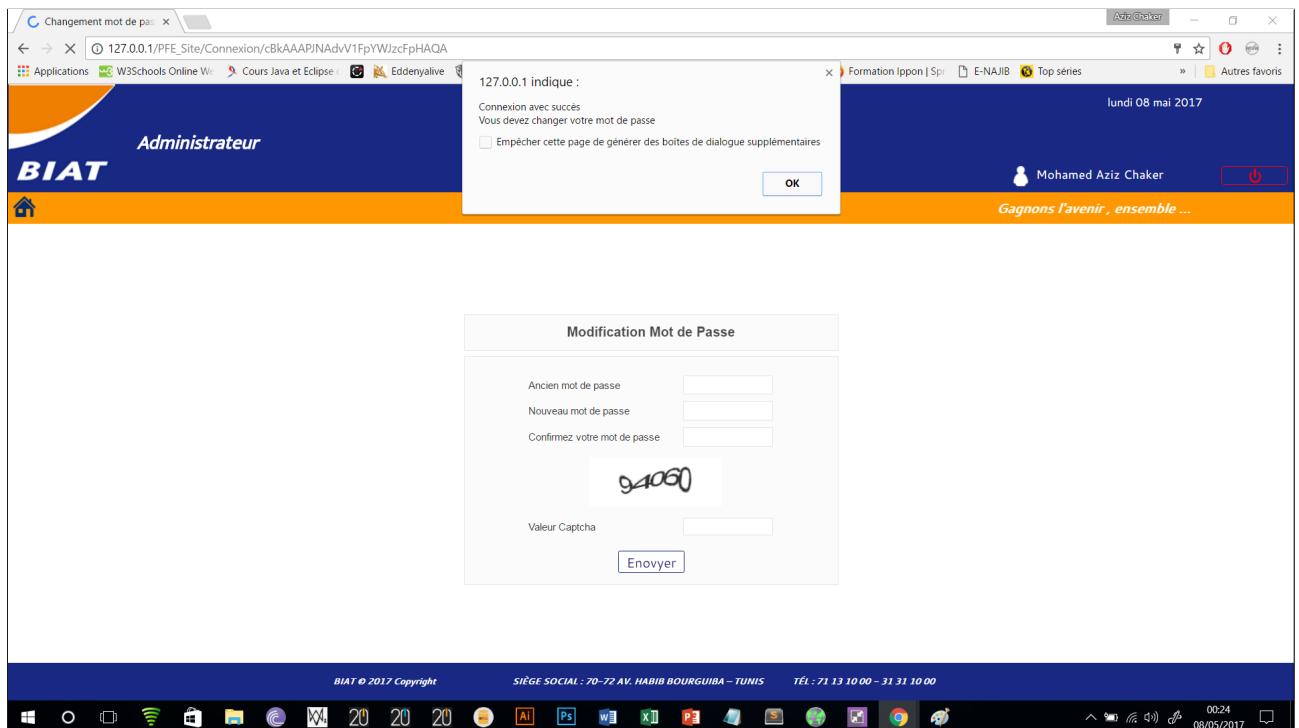


Figure 5.14: Interface «Changer mot de passe»

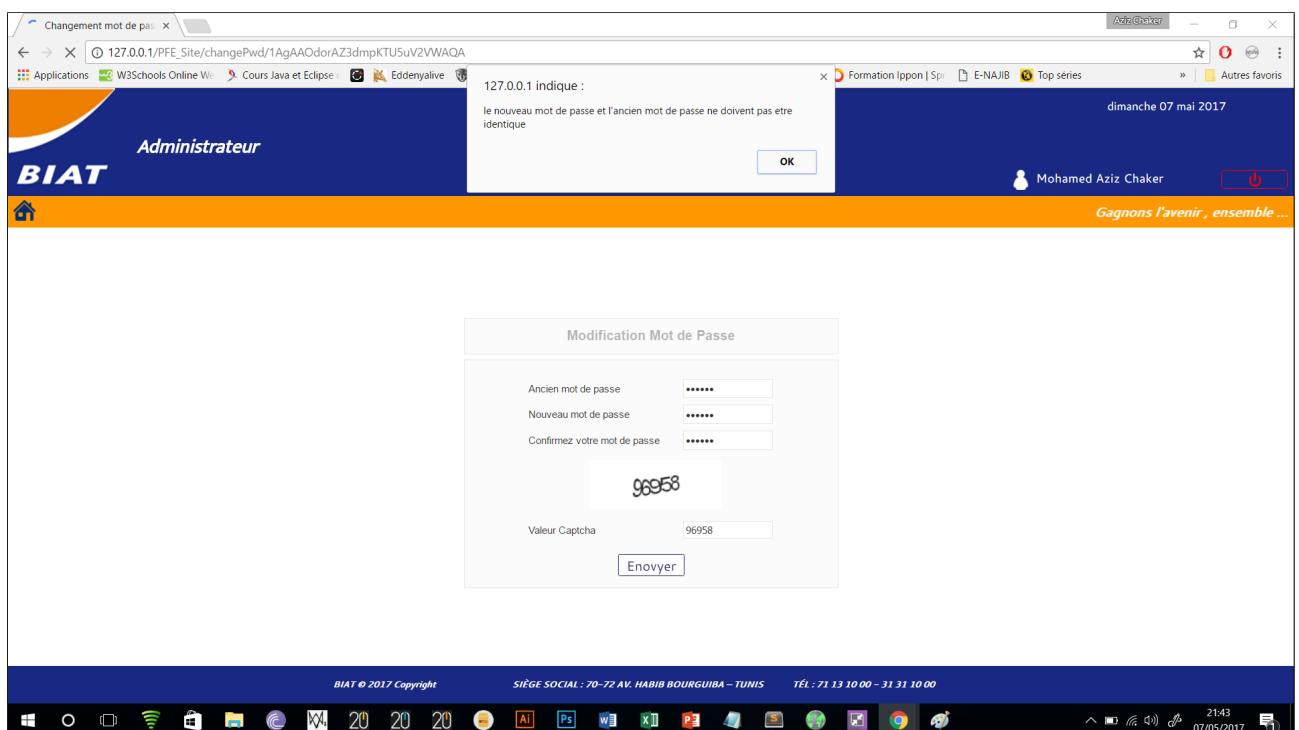


Figure 5.15: Interface «Avertissement ancien mot de passe»

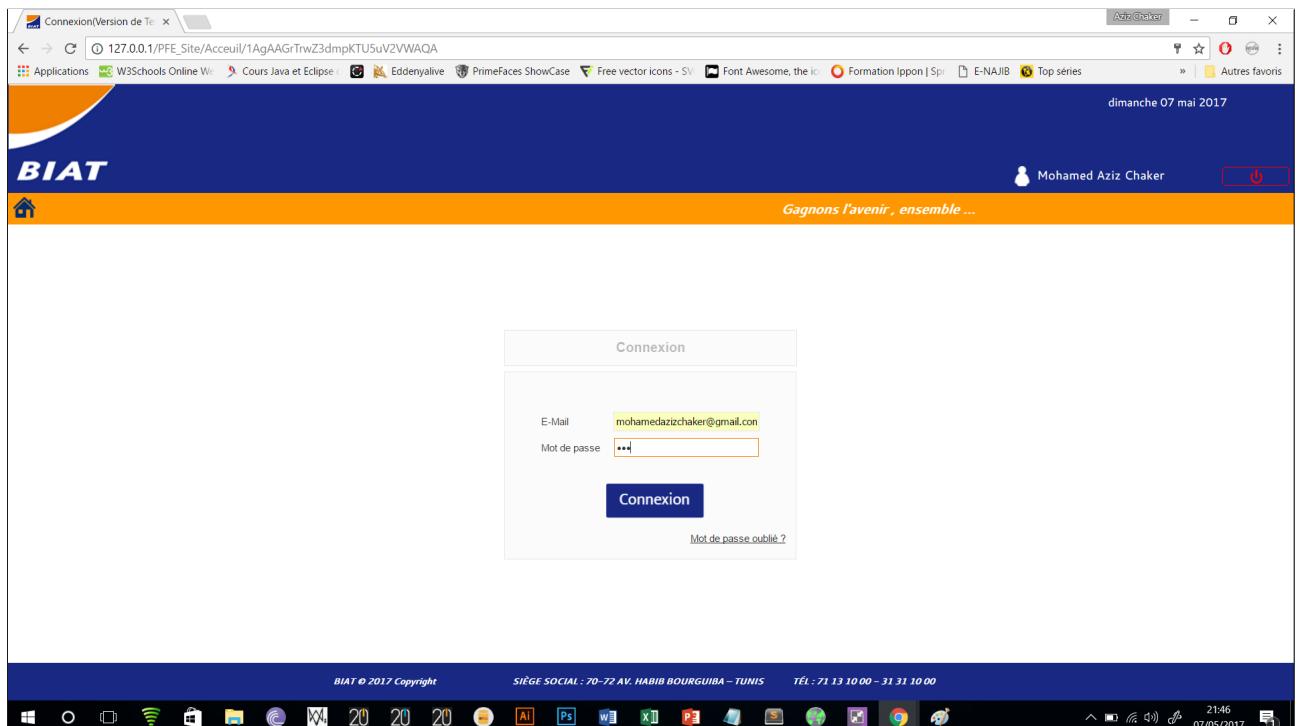


Figure 5.16: Interface «Reconnexion»

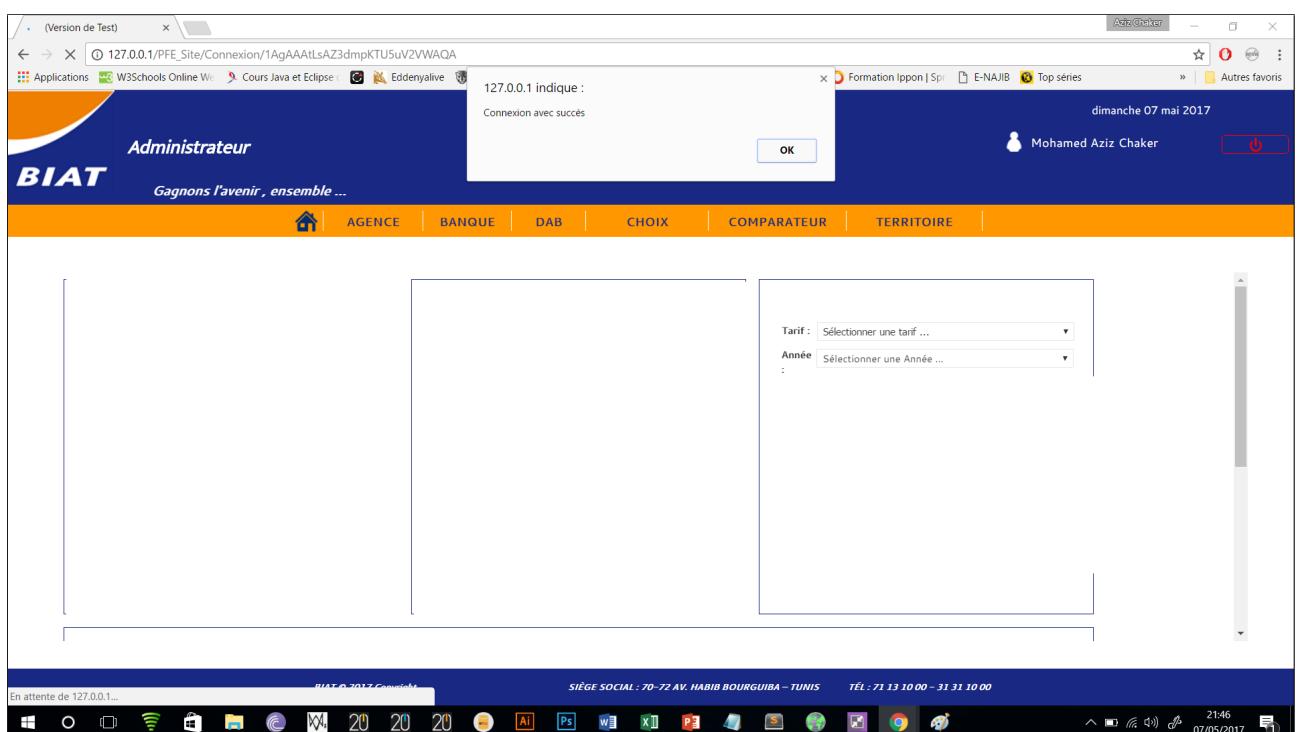


Figure 5.17: Interface «Reconnexion avec succès»

5.3.3 Ajout Banque

L'administrateur clique sur le bouton «Nouvelle banque» de l'interface de consultation, puis Il saisit l'identifiant unique de la nouvelle banque, ensuite Le système vérifie l'inexistence de cet identifiant dans la base. Après, l'administrateur continue à saisir les informations restantes de la banque. Enfin, le système enregistre les données saisies et affiche un message indiquant l'ajout avec succès de la nouvelle banque.

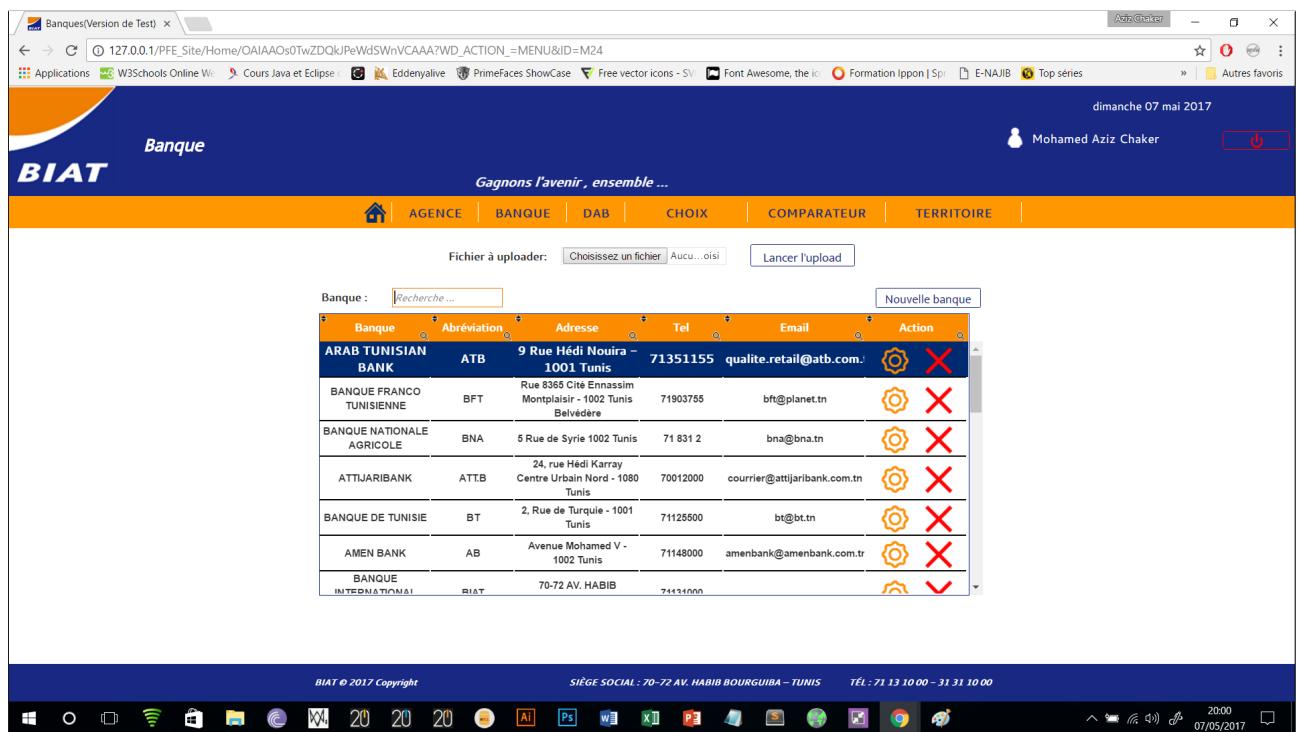


Figure 5.18: Interface «Consulter banque»

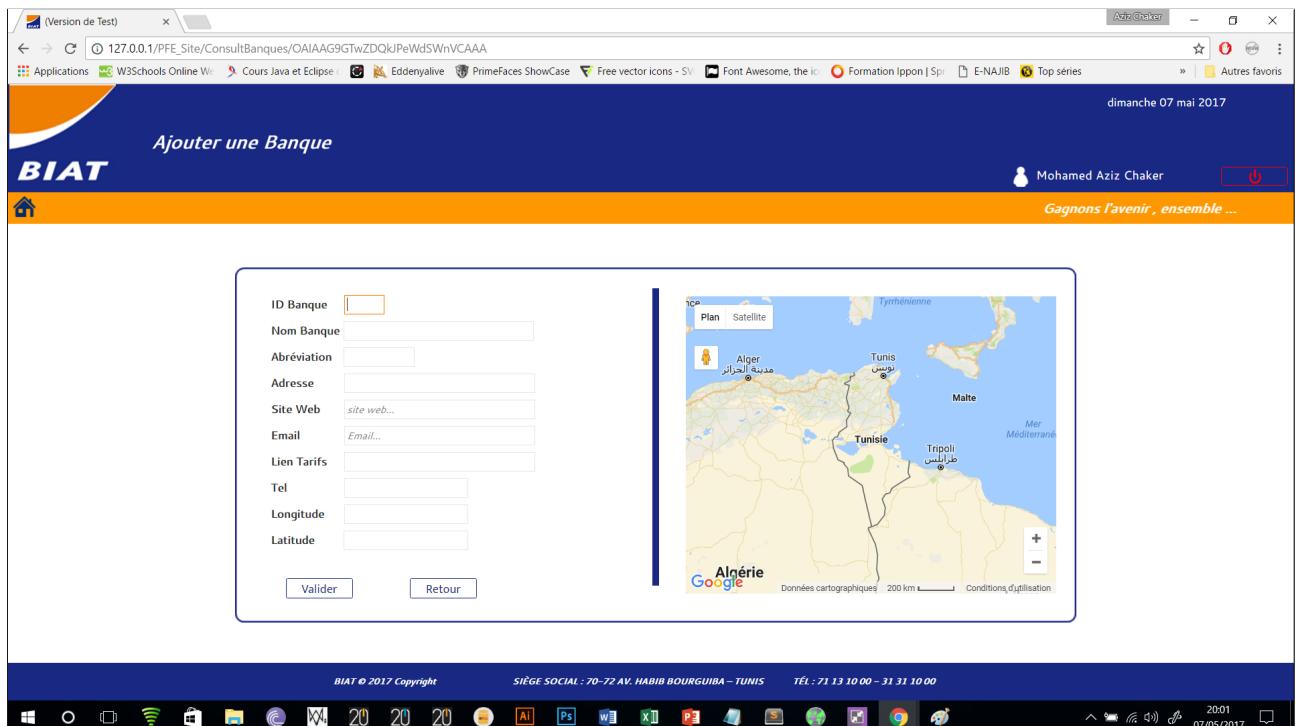


Figure 5.19: Interface «Ajouter banque»

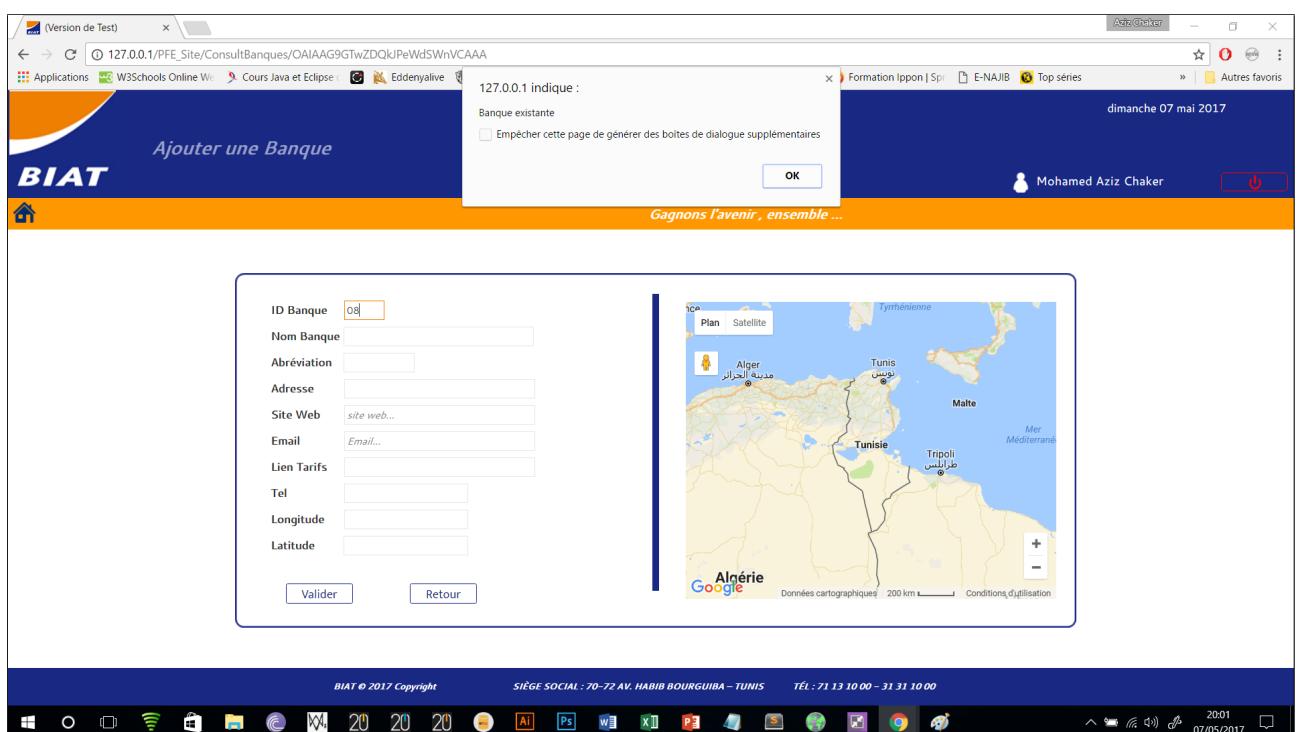


Figure 5.20: Interface «Erreur identifiant banque existant»

CHAPTER 5. LA PHASE DE TRANSITION

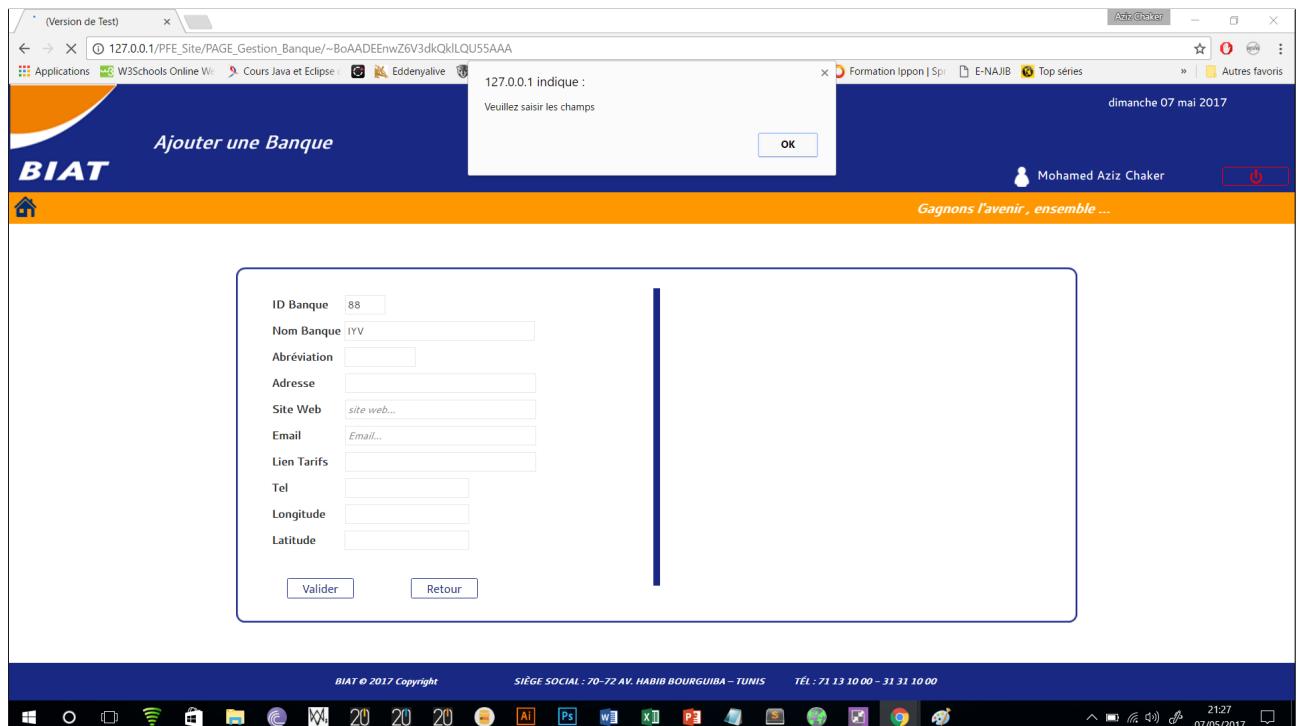


Figure 5.21: Interface «Erreurs vérifier les champs»

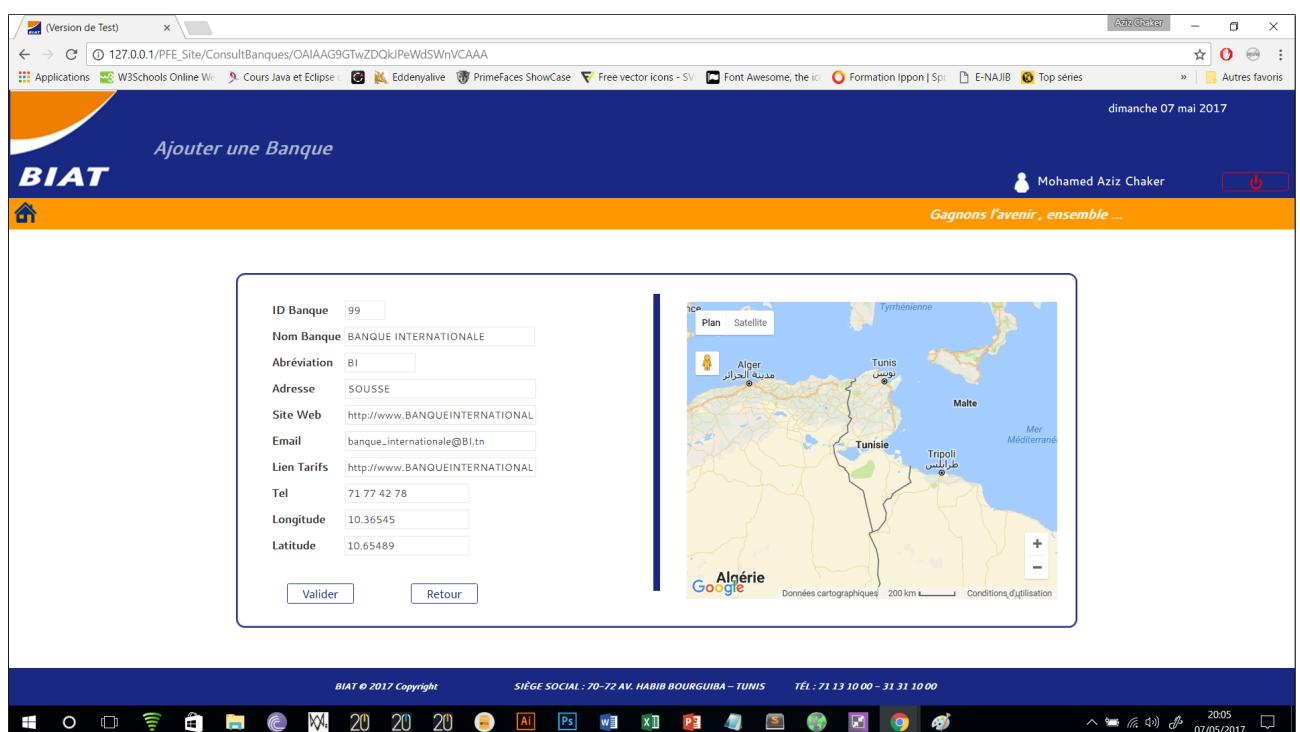


Figure 5.22: Interface «Informations banque saisies»

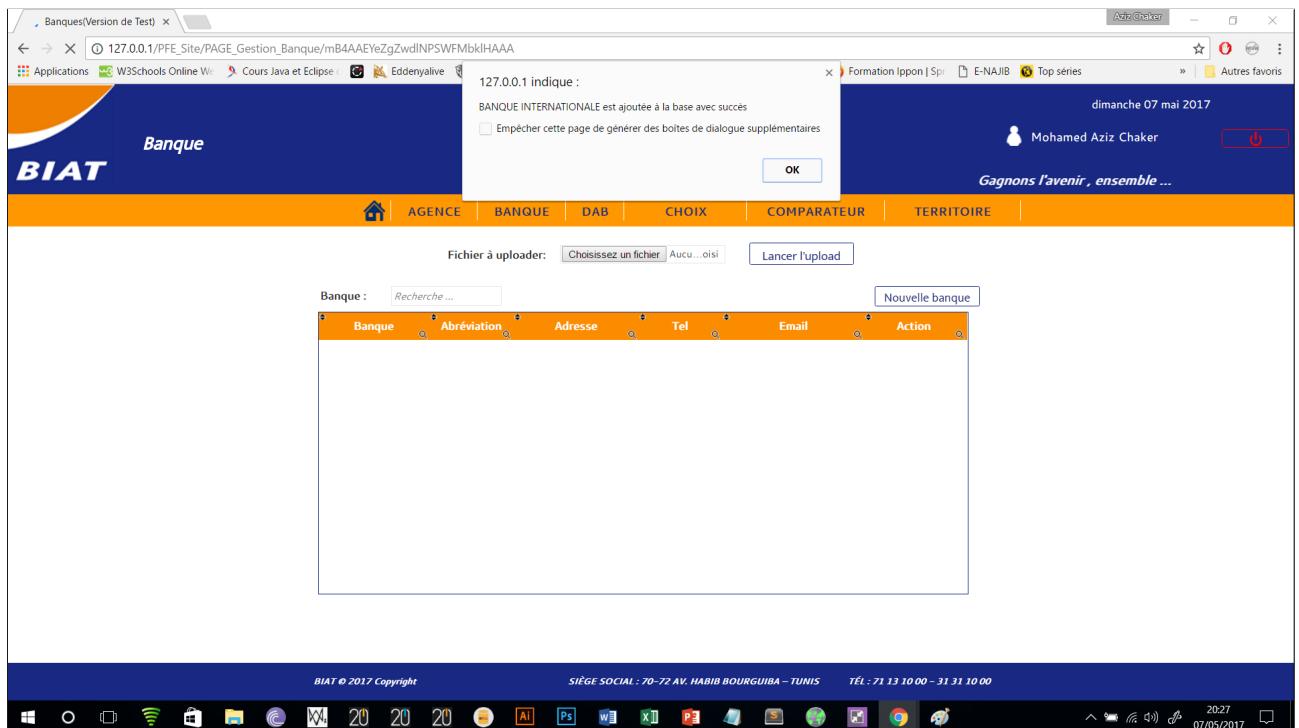


Figure 5.23: Interface «Ajout banque avec succès»

5.3.4 Ajout Tarif

L'administrateur clique sur le bouton «Nouvelle tarif» de l'interface de consultation, puis il saisit l'identifiant unique de la catégorie correspondante à la nouvelle tarif. Le système vérifie l'existence de cet identifiant dans la base, ensuite l'administrateur continue à saisir les informations restantes de la tarif. Enfin le système enregistre les données saisies et affiche un message indiquant l'ajout avec succès de la nouvelle tarif.

CHAPTER 5. LA PHASE DE TRANSITION

The screenshot shows a web browser window titled '(Version de Test)' with the URL '127.0.0.1/PFE_Site/Home/cCYAAN0dbgdKQ1BzTEF4Uk5MAAA?WD_ACTION_=MENU&ID=M27'. The page is titled 'Tarif' and features the BIAT logo. The top navigation bar includes links for 'AGENCE', 'BANQUE', 'DAB', 'CHOIX', 'COMPARATEUR', and 'TERRITOIRE'. A search bar labeled 'Fichier:' with the placeholder 'Choisissez un fichier' and a button 'Lancer l'upload' is present. Below this is a table titled 'Tarif' with columns: 'ID Catégorie', 'Tarif', 'Assiette', 'Type Tarif', 'Information', 'Périodicité', and 'Action'. The table contains six rows of data, each with a delete icon ('X') in the 'Action' column.

Figure 5.24: Interface «Consulter tarif»

The screenshot shows a web browser window titled '(Version de Test)' with the URL '127.0.0.1/PFE_Site/TarifConsult/cCYAAD6ubgdKQ1BzTEF4Uk5MAAA'. The page is titled 'Ajouter une Tarif' and features the BIAT logo. The top navigation bar includes links for 'AGENCE', 'BANQUE', 'DAB', 'CHOIX', 'COMPARATEUR', and 'TERRITOIRE'. A search bar labeled 'Fichier:' with the placeholder 'Choisissez un fichier' and a button 'Lancer l'upload' is present. Below this is a form for adding a new rate, containing fields for 'ID Catégorie', 'Tarif', 'Assiette', 'Type Tarif', 'Information', and 'Périodicité'. At the bottom of the form are 'Valider' and 'Retour' buttons. The footer of the page includes copyright information 'BIAT © 2017 Copyright' and contact details 'SIÈGE SOCIAL : 70-72 AV. HABIB BOURGUIBA – TUNIS' and 'TÉL : 71 13 10 00 – 31 31 10 00'.

Figure 5.25: Interface «Ajout tarif»

CHAPTER 5. LA PHASE DE TRANSITION

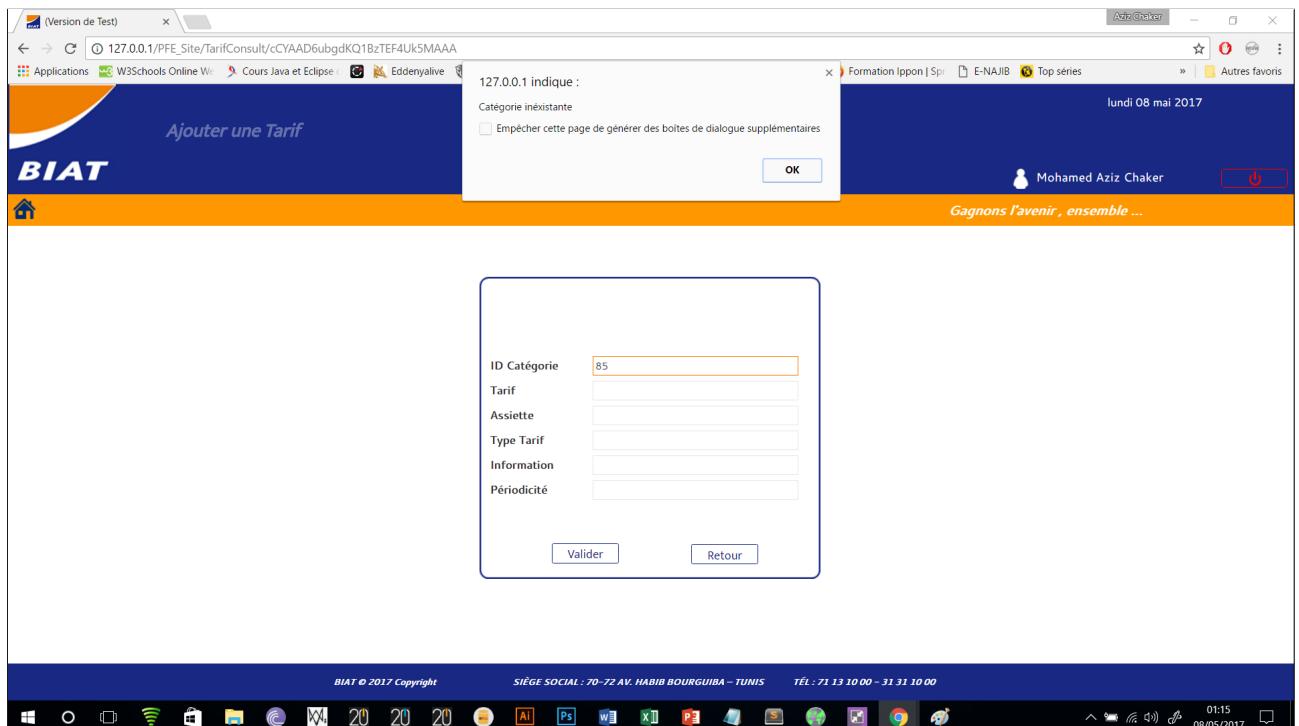


Figure 5.26: Interface «Erreur ID categorie existant»

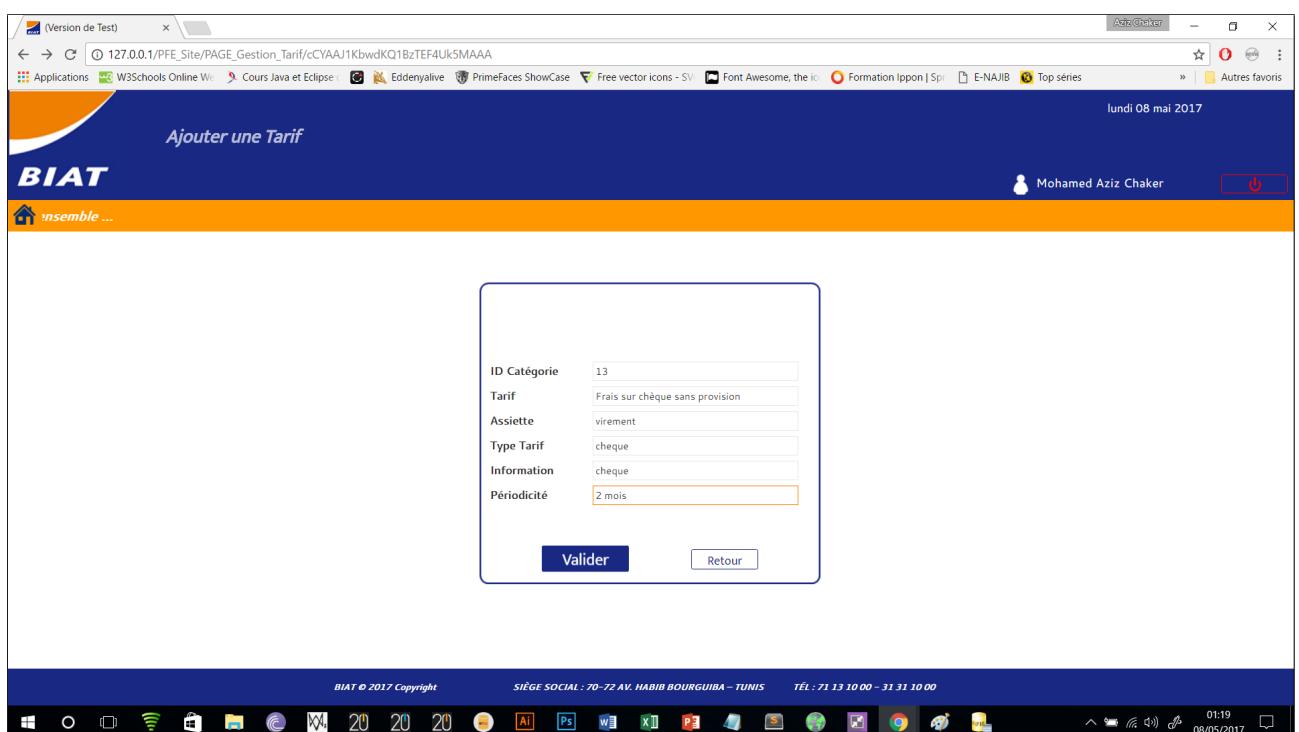


Figure 5.27: Interface «Informations tarif saisies»

CHAPTER 5. LA PHASE DE TRANSITION

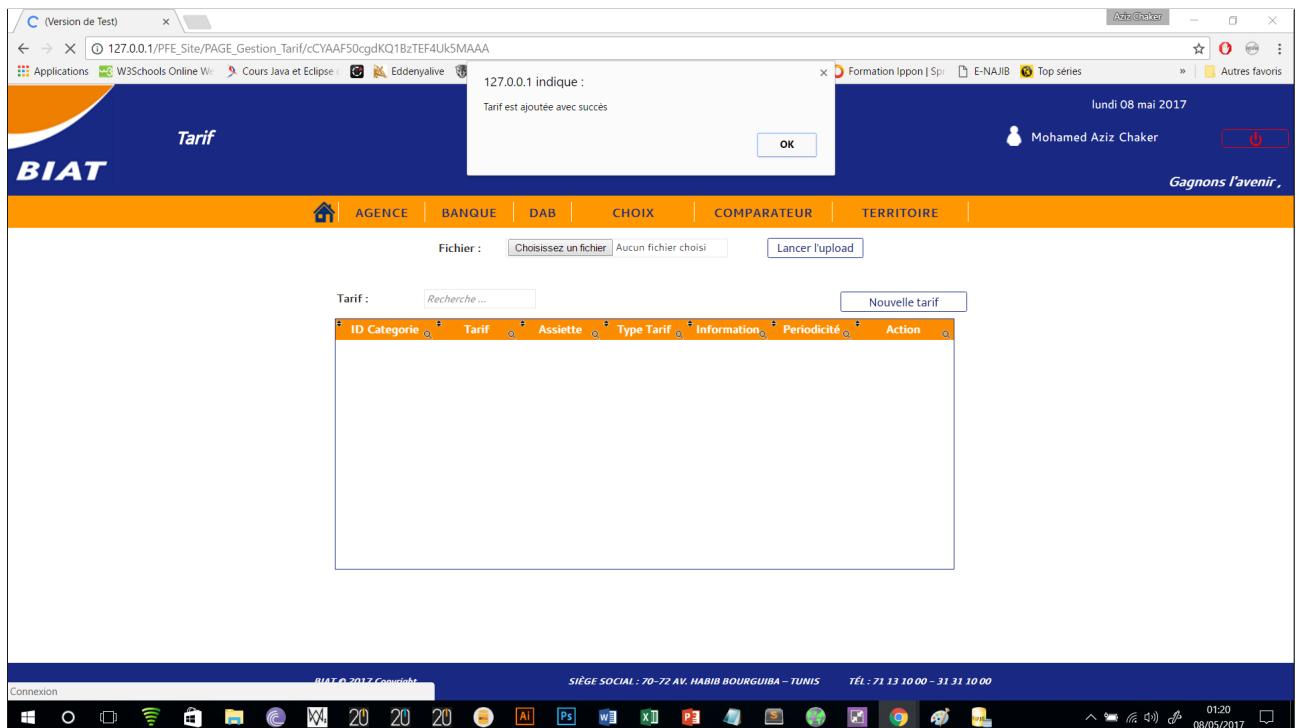


Figure 5.28: Interface «Ajout tarif avec succès»

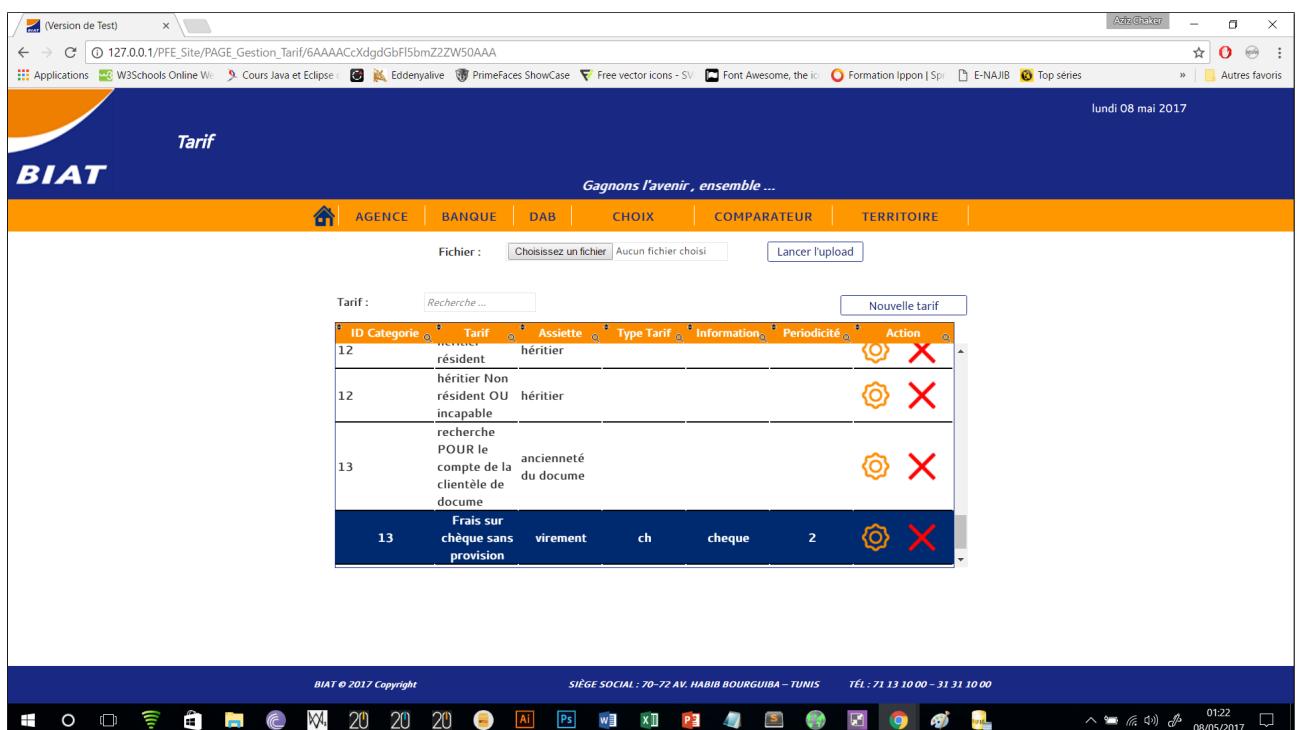


Figure 5.29: Interface «Nouvelle tarif ajoutée»

5.3.5 Modifier tarification

L'administrateur choisit la tarification à modifier depuis l'interface de consultation, puis il clique sur l'icône de la modification. Le système affiche la page de la modification avec ses informations. Ensuite l'administrateur modifie les informations désirées. Après Il clique sur le bouton «valider». Enfin Le système met à jour la tarification avec les nouvelles informations et affiche un message indiquant la modification avec succès.

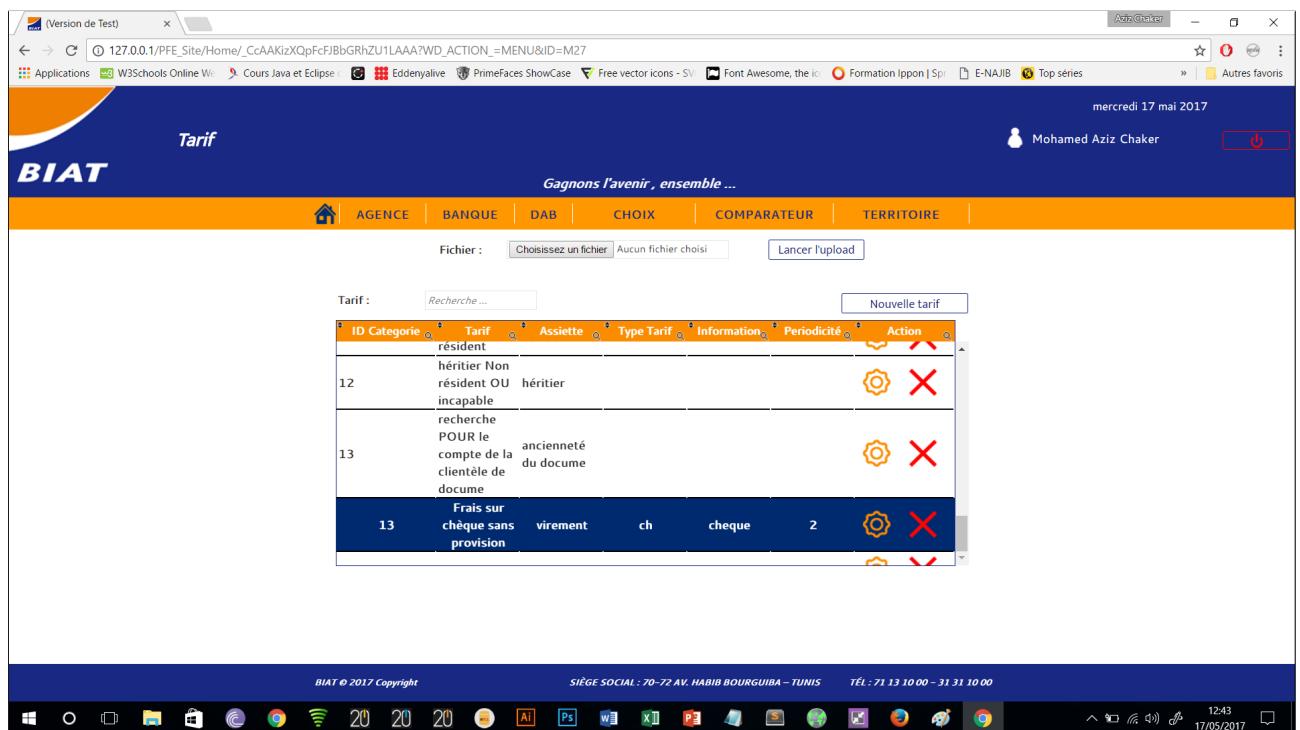


Figure 5.30: Interface «Consulter tarification»

CHAPTER 5. LA PHASE DE TRANSITION

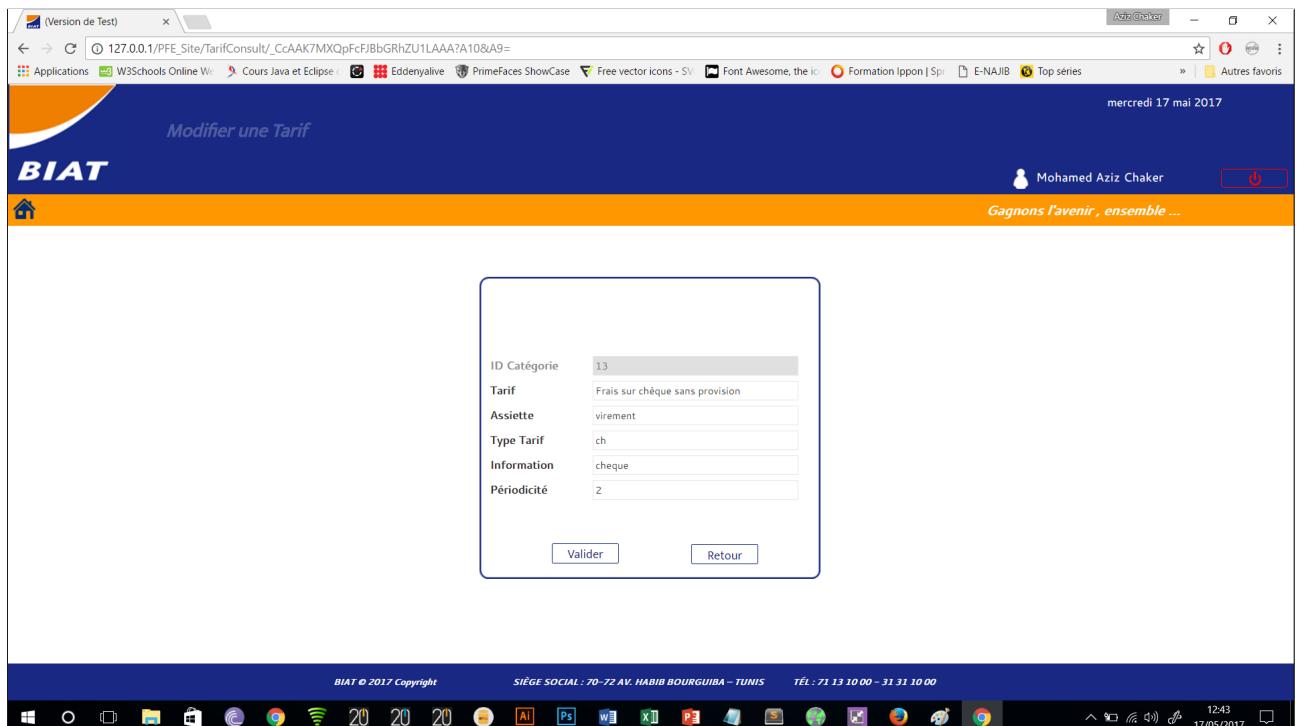


Figure 5.31: Interface «Modifier tarification»

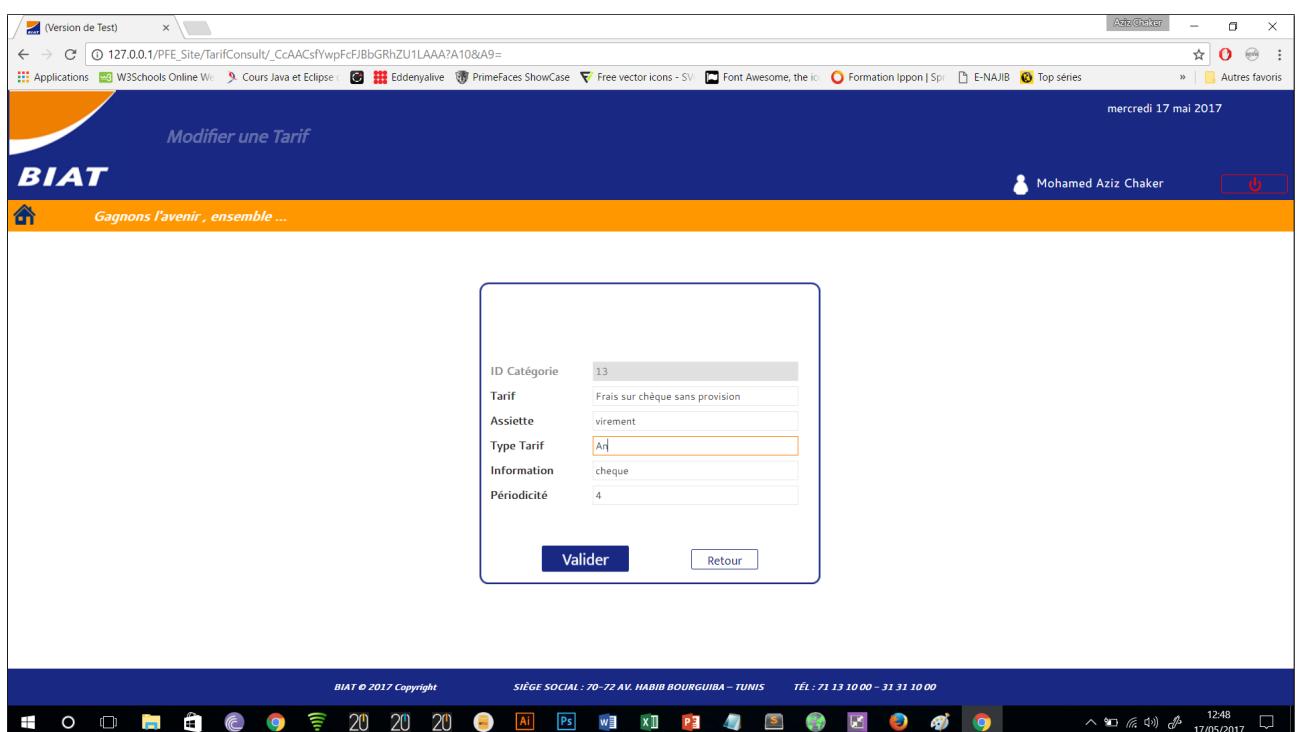


Figure 5.32: Interface «Modification des informations»

CHAPTER 5. LA PHASE DE TRANSITION

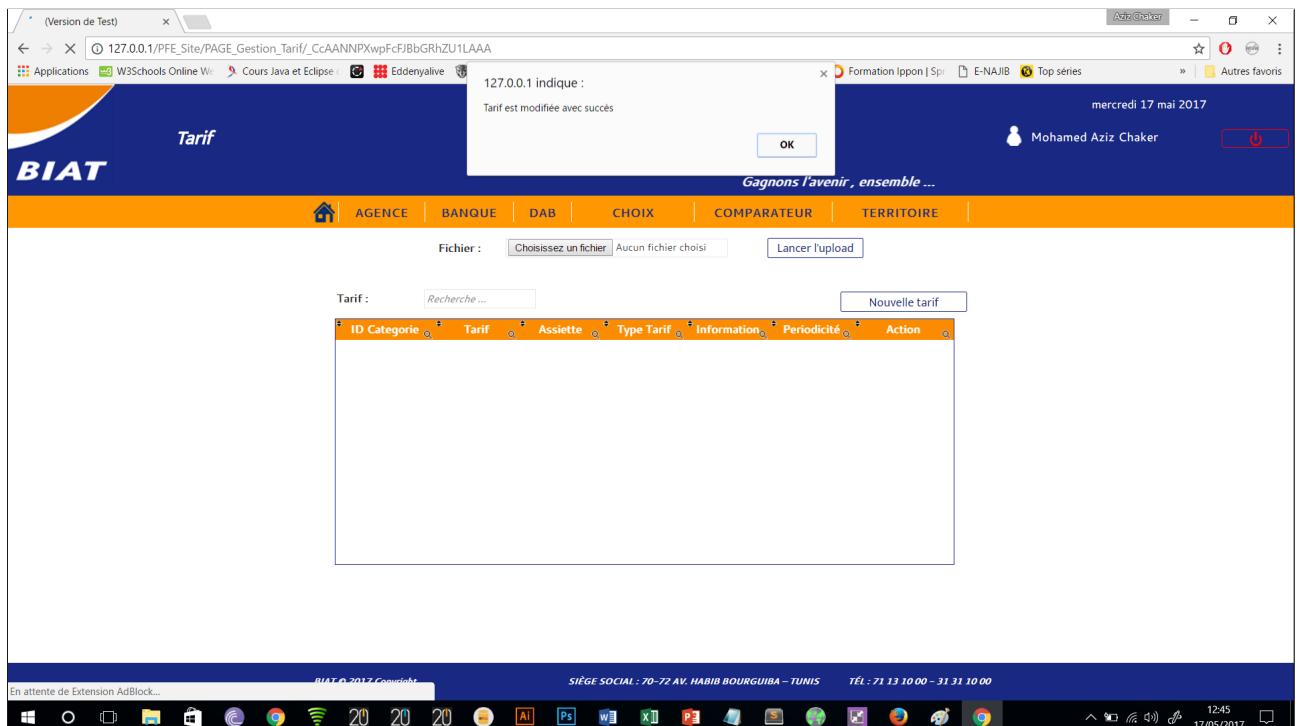


Figure 5.33: Interface «Succès de modification»

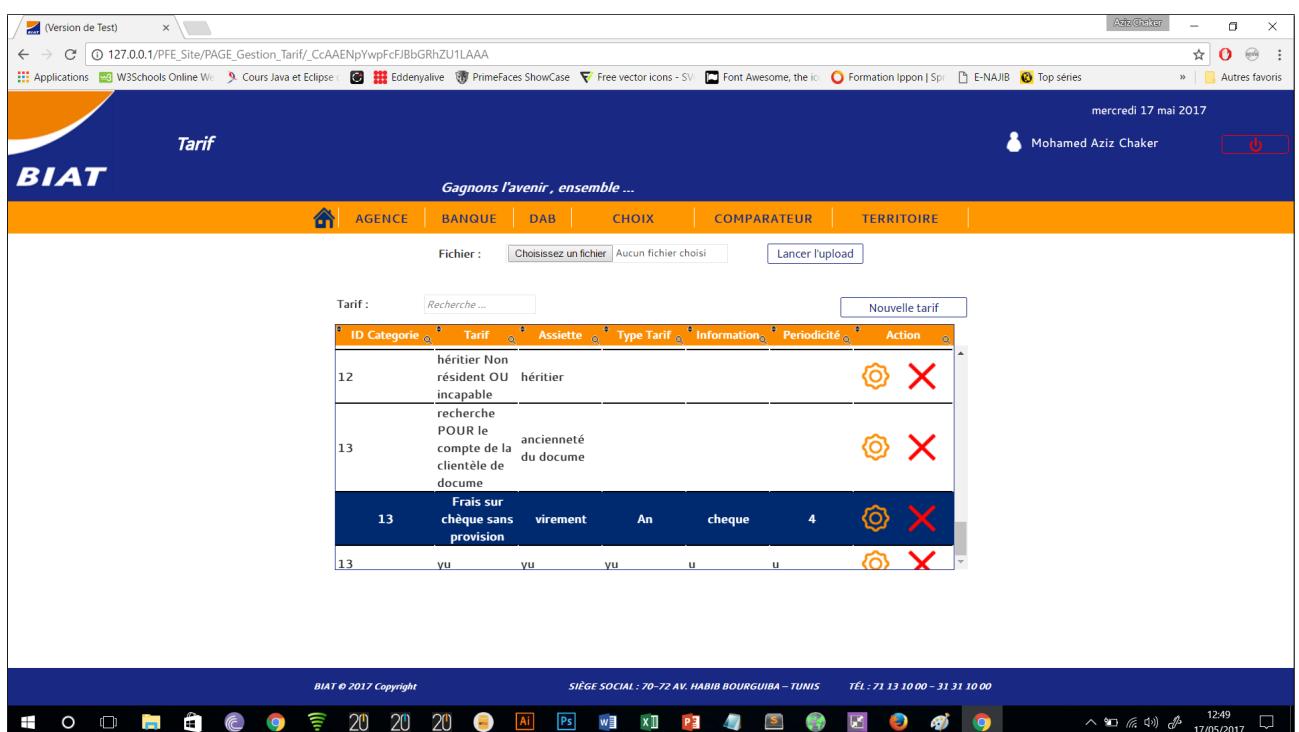


Figure 5.34: Interface «Consultation mis à jour»

5.3.6 Comparaison des tarifications

L'administrateur ou l'adhérent sélectionne les tarifs ou la catégorie des tarifs à comparer, puis Il sélectionne ensuite les banques à comparer. Enfin Le système compare les valeurs des tarifications et affiche les banques sélectionnées par ordre croissant des valeurs comparées.

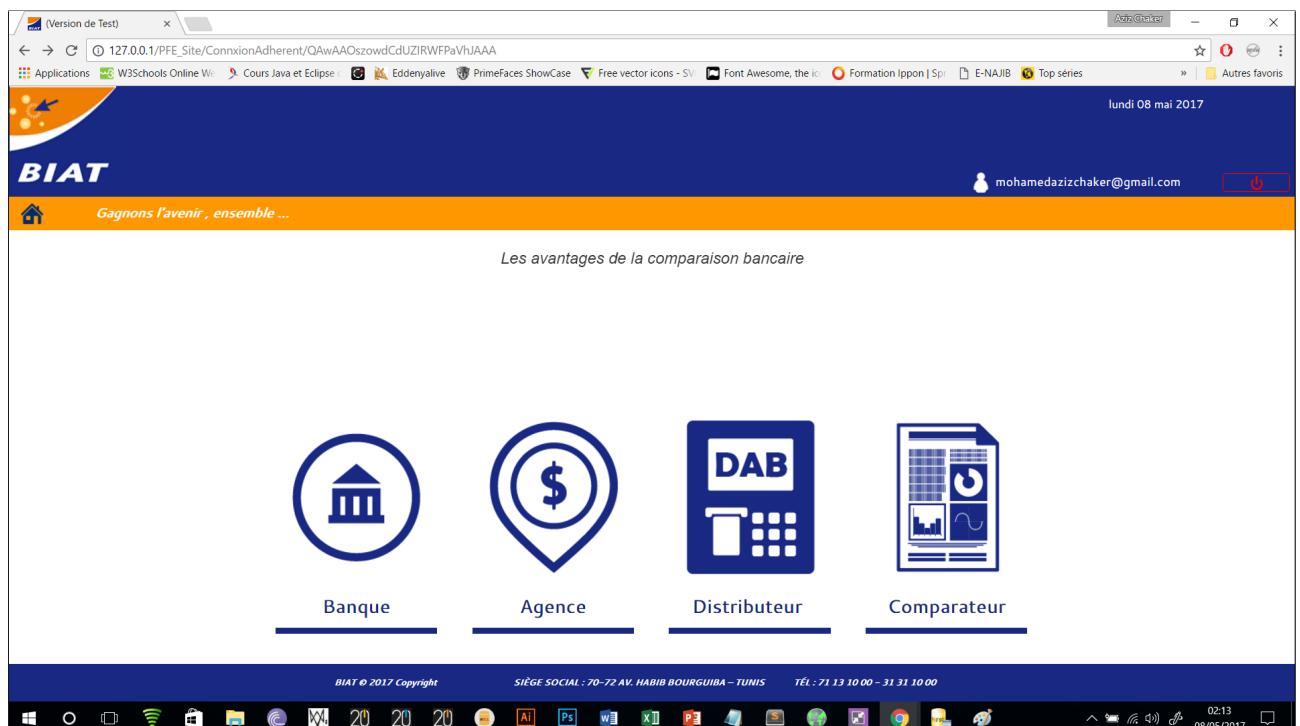


Figure 5.35: Interface «Accueil»

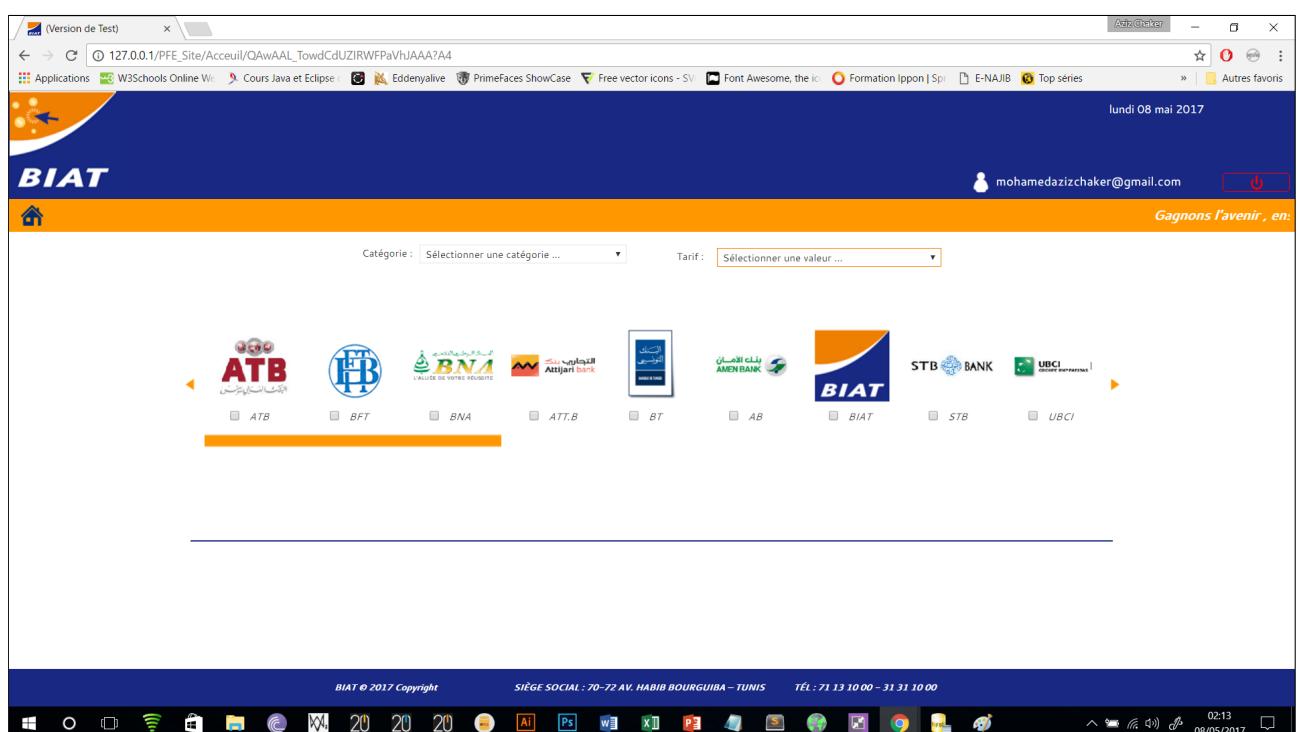


Figure 5.36: Interface «Comparateur»

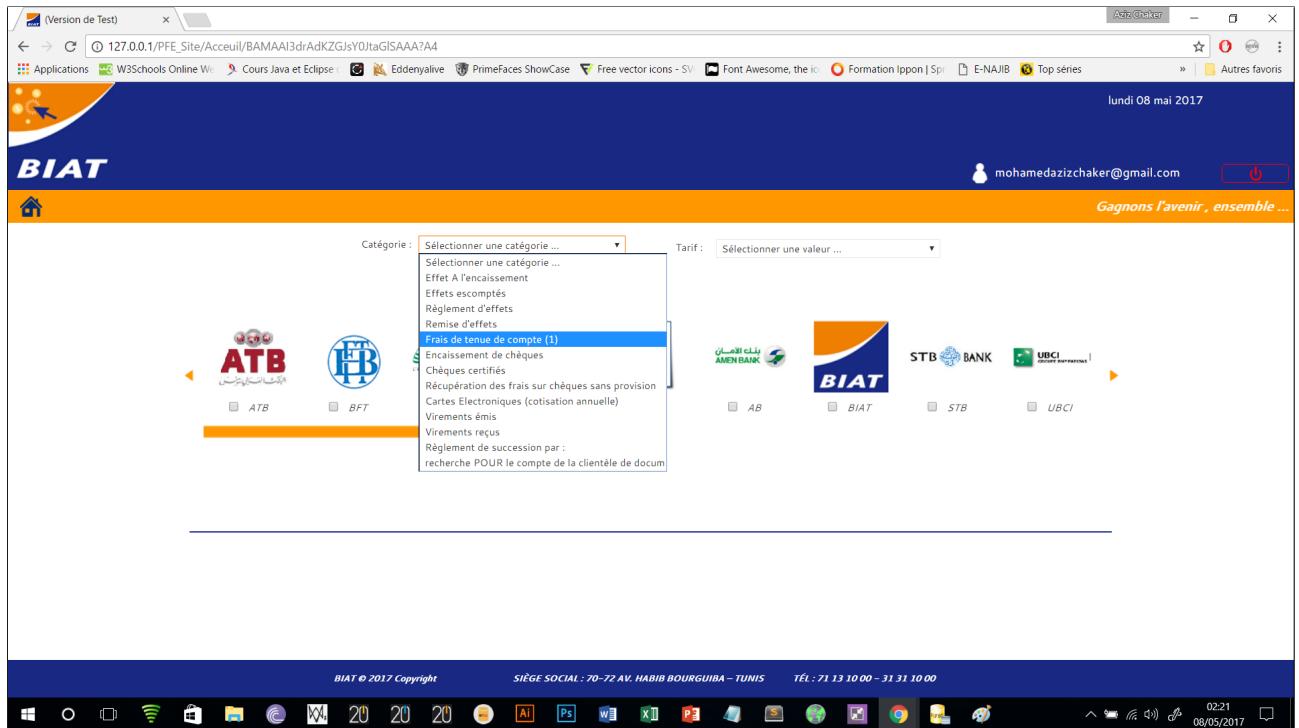


Figure 5.37: Interface «Sélectionner catégorie»

Banque	Tarif	Valeur	Min	Max
Frais de tenue de compte (1)				
BIAT	Compte chèque	12	0	0
CBNA	Compte chèque	100	0	0
ALBARAKA	Compte chèque	120	0	0
ABC	Compte chèque	177	0	0
Frais de tenue de compte (1)				
CBNA	Compte courant	0	0	0
ALBARAKA	Compte courant	15	15	55
BIAT	Compte courant	25	0	0
ABC	Compte courant	45	45	85
Frais de tenue de compte (1)				
BIAT	Compte d'épargne	1.5	0	0

Figure 5.38: Interface «Tarifs comparées d'une catégorie»

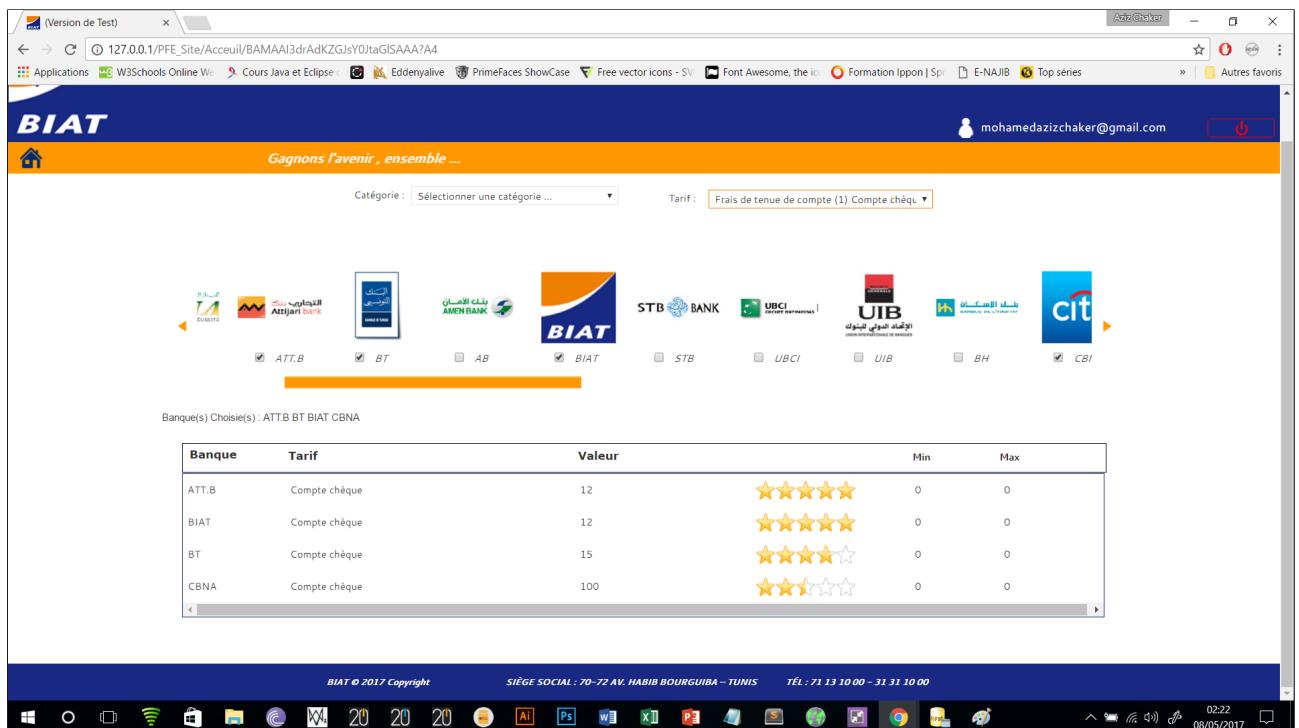


Figure 5.39: Interface «Tarifications comparées d'une tarif»

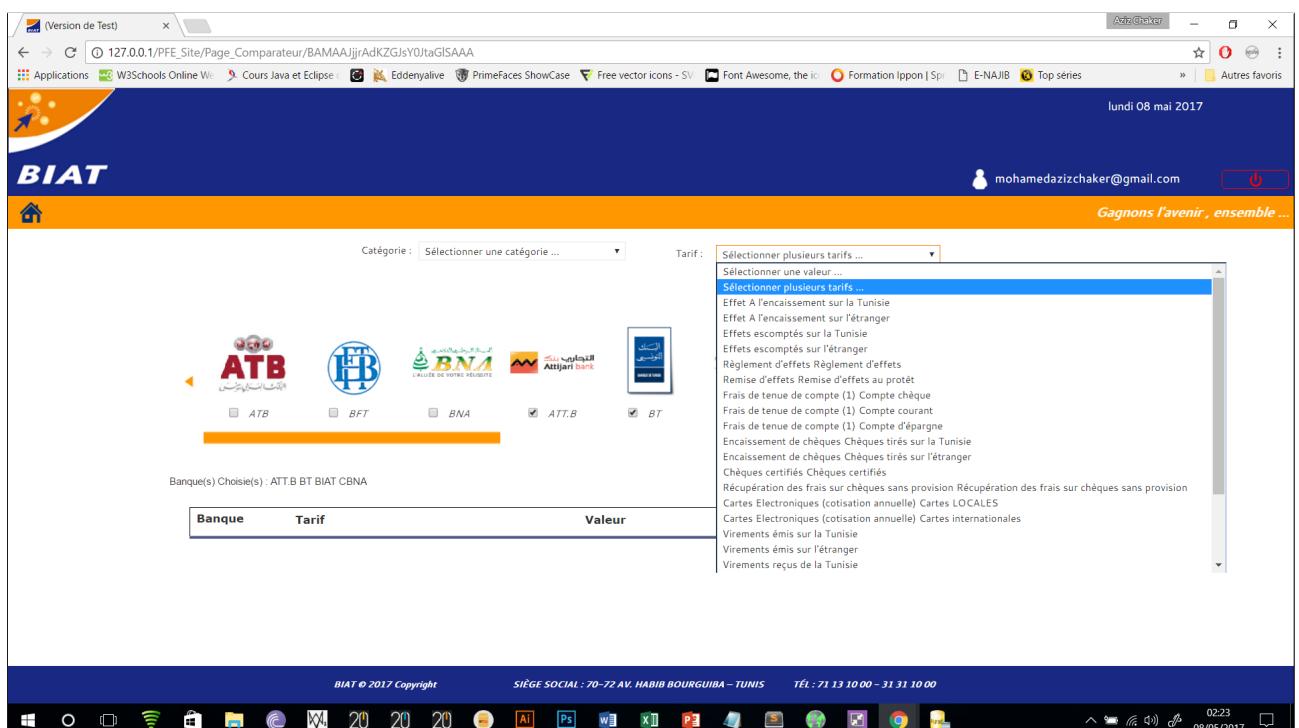


Figure 5.40: Interface «Sélectionner plusieurs tarifs»

CHAPTER 5. LA PHASE DE TRANSITION

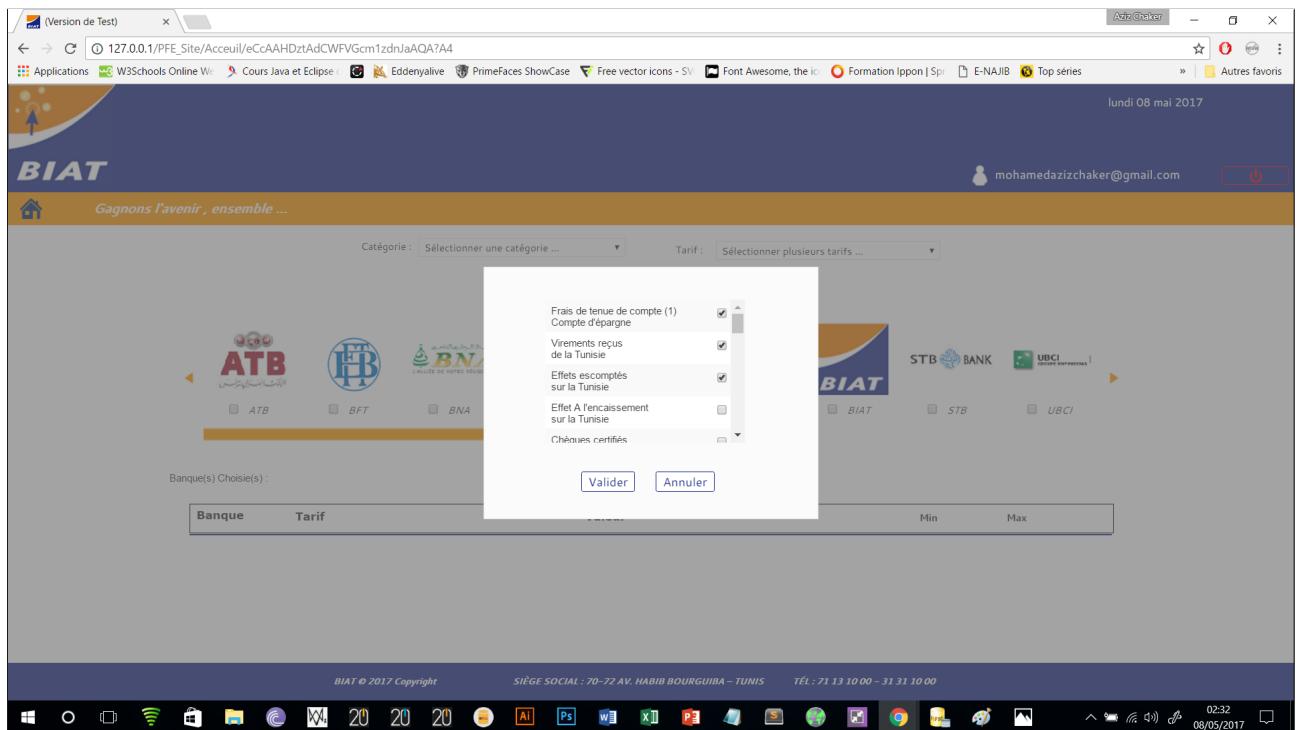


Figure 5.41: Interface «Choisir les différentes tarifs»

The screenshot shows a comparison table for different banks. The table has columns for 'Banque', 'Tarif', 'Valeur', 'Min', and 'Max'. The 'Valeur' column includes a star rating. The footer contains copyright information ('BIAT © 2017 Copyright'), a address ('SIÈGE SOCIAL : 70-72 AV. HABIB BOURGUIBA – TUNIS'), and a phone number ('TÉL : 71 13 10 00 – 31 31 10 00'). The taskbar at the bottom shows various application icons.

Banque	Tarif	Valeur	Min	Max
BIAT	Frais de tenue de compte (1) Compte d'épargne Virements reçus de la Tunisie Effets escomptés sur la Tunisie	2.50		
BNA	Frais de tenue de compte (1) Compte d'épargne Virements reçus de la Tunisie Effets escomptés sur la Tunisie	2.77		
BH	Frais de tenue de compte (1) Compte d'épargne Virements reçus de la Tunisie Effets escomptés sur la Tunisie	36.50		
CBNA	Frais de tenue de compte (1) Compte d'épargne Virements reçus de la Tunisie Effets escomptés sur la Tunisie	55.00		
QNB	Frais de tenue de compte (1) Compte d'épargne Virements reçus de la Tunisie Effets escomptés sur la Tunisie	77.50		

Figure 5.42: Interface «Résultat du comparaison»

Conclusion

Durant la dernière phase, nous avons décrit le scénario de fonctionnement de notre application tout en présentant quelques interfaces utilisateurs afin de bien clarifier ses activités, en tenant compte de leur déroulement. Grâce à l’interaction de l’environnement logiciel et matériel, nous avons eu une application exécutable que nous allons livrer à une exploitation réelle dans l’environnement du client (mise en place et installation).

Conclusion générale

Ce rapport de synthèse présente brièvement le stage que nous avons réalisé au sein de la Banque Internationale Arabe de Tunisie (BIAT). Durant ces trois mois, nous avons conçu et développé une application de gestion et comparaison des tarifications bancaires destinée aux clients.

Pour modéliser notre application, nous avons fait recours à la méthodologie appelée "Processus Unifié" et suivi toutes ses phases (incubation, élaboration, construction, transition) aussi bien que ses activités. Cela nous a permis de relever les avantages d'un travail itératif et incrémental où les besoins ont été enrichis au fur et à mesure de l'avancement dans les différentes itérations.

Quant au développement, nous avons exploité l'architecture trois tiers et appliqué les notions fondamentales de l'orienté objet à une échelle réelle en manipulant l'outils de développement WEBDEV. En plus, nous avons appris à utiliser le langage L^AT_EX.

Nous avons confronté, sans doute, quelques problèmes techniques dans la phase de construction du système, lors de l'intégration de notre environnement de développement. Ce qui nous a conduits à s'auto-former pour résoudre ce problème et mieux manipuler l'outil.

Globalement, nous avons pu atteindre ces objectifs visés ainsi que d'enrichir nos connaissances non seulement sur le plan pratique mais aussi sur le plan théorique.

Grâce à ce stage, nous avons eu l'occasion d'apprendre à définir nos objectifs puis à mettre en place les moyens de les atteindre tout en bénéficiant de l'aide et de l'évaluation de nos encadrants. De ce fait, nous avons appris le dynamisme du travail collaboratif.

Nous souhaitons donc que ce projet soit d'une utilité importante aussi bien pour la société que pour notre avenir professionnel. Finalement, plusieurs améliorations restent envisageables dans ce travail. Tels que, l'intégration d'autres fonctionnalités comme la négociation en ligne des tarifications bancaires pour les clients professionnels et la visualisation en temps réel du cours de change dans les salles des marchés financiers des différentes banques.

Bibliography

- [1] BIAT: La BIAT en bref,
<http://www.biat.com.tn/qui-sommes-nous/presentation-generale/la-biat-en-bref.html>
- [2] Mme Olfa Mraihi. *Cours du Processus Unifié.*
- [3] Mataf: Définition Tarification Bancaire,
<https://www.mataf.net/fr/edu/glossaire/tarification-bancaire>
- [4] WEBDEV: présentation,
<http://dictionnaire.sensagent.leparisien.fr/WebDev/fr-fr/>
- [5] Présentation de HFSQL,
<https://www.pcsoft.fr/hfsql/presentation-de-hfsql.htm>
- [6] Tutorial Rational Rose,
<https://www.utc.fr/~wschon/lo19/Tutorial%20Rational%20Rose.pdf>
- [7] Adobe Photoshop,
https://fr.wikipedia.org/wiki/Adobe_Photoshop
- [8] Adobe Illustrator,
https://fr.wikipedia.org/wiki/Adobe_Illustrator
- [9] Définition benchmark,
<http://www.definitions-marketing.com/definition/benchmark/>

Annexes

Annexe 1: Définition du benchmark

Le benchmark est, dans un contexte marketing, une démarche d'observation et d'analyse des pratiques marketing utilisées et des performances atteintes par d'autres entreprises. Ces entreprises observées et analysées peuvent appartenir à la concurrence ou être des acteurs pouvant avoir des modes de fonctionnement réutilisables par l'entreprise commanditaire du benchmark. On parle alors de benchmark concurrentiel ou de benchmark sectoriel.

Le benchmark peut également avoir trait à l'observation des pratiques managériales ou être orienté vers la mesure et comparaison de performances, comme par exemple dans l'informatique.

Le benchmark marketing peut être considéré de manière simpliste comme « une pêche aux bonnes idées ». Le benchmark est plus large qu'une pratique de veille concurrentielle dans la mesure où il ne s'intéresse pas forcément qu'aux concurrents.

Le benchmark marketing peut être global ou porter sur une problématique spécifique. Le benchmark peut s'intéresser aux pratiques des concurrents, mais il peut également observer et analyser comment des acteurs non concurrents vont gérer des problématiques proches. Le benchmark peut alors s'intéresser aux acteurs vendant des cabanes de jardins ou des trampolines. [9]

Annexe 2: Créez une application grâce au RAD

Le RAD (Rapid Application Development) est une technologie permettant de créer en quelques clics une application complète (ou un site complet). Le RAD se base sur des modèles de programmation pré-établis (appelés Patterns), permettant de générer l'application voulue. Il est également possible de créer entièrement ses propres "patterns" pour générer automatiquement une application personnalisée : vous décidez du code généré ainsi que du positionnement des champs. Le RAD peut être lancé de la façon suivante : sous le volet "Projet", dans le groupe "Génération", cliquez sur "RAD Application complète". Lorsque le modèle est choisi, il suffit de suivre les différentes étapes de l'assistant.

Pour un site WEBDEV, les patterns RAD proposés par défaut sont les suivants :

- Pattern RAD AWP : Ce pattern est destiné à la génération d'un site AWP. Il propose les fonctionnalités suivantes : tables Ajax, zones répétées, génération d'états au format PDF, gestion des menus, gestion du référencement, gestion de cellules déplaçables.
- Pattern Intranet : Ce pattern est destiné à la génération d'un site Intranet/Extranet. Il propose les fonctionnalités suivantes : tables Ajax, génération d'états au format PDF, gestion des menus sous forme d'onglets.
- Pattern RAD compatible 11 : Ce pattern permet de générer une application correspondant aux applications générées par la version 11.