

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE MATHÉMATIQUES ET
D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE
N° :



DOMAINE : MATHÉMATIQUES ET
D'INFORMATIQUE
FILIERE : INFORMATIQUE
OPTION : INFORMATIQUE
DECISIONNELLE SE OPTIMISATION

**Mémoire présenté pour l'obtention
Du diplôme de Master Académique**

**Par : Saadi Abdelhakim
Mohammedi Menad**

Intitulé

**Un EAI (Enterprise Application Intégration)
pour le rassemblement des sources financières
Etude de cas : chambre de l'agriculture de M'sila**

Soutenu devant le jury composé de :

Dr.Mouhoub N	Université de M'sila	Président
Dr.Loucif H	Université de M'sila	Examinateur

Encadré par : dr. brahimi mahmoud

Année universitaire : 2020/2021

Remerciement

Nous tenons avant tout à remercier notre Bon Dieu tout Puissant de m'avoir donné le Courage, la force et la volonté pour achever ce modeste travail.

Notre encadreur, monsieur brahimi mahmoud pour nous avoir encadrés, dirigé et conseillé, et pour avoir mis à notre disposition tous les moyens nécessaires à l'accomplissement de ce travail.

Nous tenons à remercier tout le personnel de l'établissement la chambre agricole,

Nos remerciements s'adressent aussi aux membres de jury, pour avoir bien voulu accepter de juger notre modeste travail. Nos reconnaissances vont également à l'ensemble des enseignants de l'université du M'sila

Dédicaces

*Je dédie ce travail en signe de reconnaissances et de
gratitudes à mes chers parents pour leur soutien
immense qui n'ont pas cessé de m'apporter ainsi que les
conseils qu'ils m'ont prodigué sans les quels j'avoue je ne
serais pas ce que je suis aujourd'hui.*

A ma petite famille et A la grande famille SAADI.

A notre encadreur monsieur BRAHIMI MAHMOUD.

A tous mes amis surtout :HOCIN MENGULATI ,DJELAL

BENLATRACH.DJABER,SOHYBE, YAKOUB

A mon binôme : MOHAMEDI MENAD.

A toute la promotion IDO2 2020-2021.

SAADI ABDELHAKIM

Dédicaces

*Je dédie ce travail en signe de reconnaissances et de
gratitudes à mes chers parents pour leur soutien
immense qui n'ont pas cessé de m'apporter ainsi que les
conseils qu'ils m'ont prodigué sans les quels j'avoue je ne
serais pas ce que je suis aujourd'hui.*

A ma femme qui a été toujours avec moi

A mes enfants Abdellah et El baraa

A la grande famille MOHAMEDI

A Mon frère Thamer

A notre encadreur monsieur BRAHIMI MAHMOUD.

A tous mes amis surtout : BELKHINECH.R,MECHIKI

DJABER ET MEGUELATI HOCINE .

A mon binôme : SAADI ABDELHAKIM.

A toute la promotion IDO2 2020-2021.

MOHAMEDI MENAD

1 Introduction générale.....	01
------------------------------	----

Premier chapitre : présentation de l'organisme d'accueil

1 Introduction.....	02
2 Définition de l'établissement	02
2.1 Attributions de l'établissement	02
2.2 Organisation et fonctionnement	03
2.2.1 Le conseil de la chambre	03
2.2.2 Le conseil administration	04
2.3 Présentation des services	05
2.4 Organigramme du personnel de la Chambre d'agriculture	06
2.5 Aspect logiciel	07
2.5.1 Les Problèmes existants	07
2.5.2 Application pour la gestion d'investissements agricoles.....	07
2.5.3 Application pour la caisse.....	08
2.5.4 Application pour la gestion d'affiliation et de radiation.....	08
2.6 Solutions	09
3 Conclusion	09

Deuxième chapitre Intégration des Applications d'Entreprise (EAI)

1 Introduction.....	10
2 Enterprise Application Intégration (EAI).....	10
2.1 Fonctionnalités	10
2.2 Composants	11
2.3 Avantages.....	11
3 Types d'EAI	11
3.1 Intégration point à point.....	11
3.2 Intégration des concentrateurs et des rayons.....	11
3.3 Bus Intégration	12
3.4 Middleware	12
3.5 Micro services.....	12
4 Typologie d'applications d'entreprise	13
4.1 Les applications compaquetées (Packaged Application).....	13

4.2 Les bases de données patrimoniales (Legacy databases).....	14
4.3 Les applications spécifiques (Custom applications).....	14
4.4 Les applications Web	14
5 Niveaux D'intégration	15
5.1 Intégration des données.....	16
5.2 Intégration de l'interface utilisateur (présentation).....	16
5.3 Intégration du processus métier.....	16
6 Défis d'intégration des applications	16
6.1 Niveau du support technologique.....	16
6.2 Restrictions technologiques et administratives.....	17
6.3 Aptitude de l'intégration avec les autres systèmes.....	17
6.4 Difficulté de compréhension des systèmes.....	17
7. Technologies d'intégration d'applications	17
7.1 Techniques basées sur les réseaux ad hoc.....	18
7.2 Techniques d'unification et de standardisation.....	19
7.3 Techniques basées sur les intergiciels.....	20
7.4 Techniques basées sur l'EAI	21
7.5 Techniques basées sur le BPM.....	23
7.6 Techniques basées sur l'ESB.....	23
8 Conclusion.....	25

Troisième chapitre : CONCEPTION ET REALISATION

1. Introduction	26
2. Présentation d'UML.....	26
2.1 Les briques de base d' UML	27
2.2 Les diagrammes dans UML	28
2.2.1. Les diagrammes des classes».....	28
2.2.2 Le diagramme d'objets	28
2.2.3 Le diagramme des cas d'utilisation	28
2.2.4 Le diagramme de séquence et le diagramme de collaboration.....	28
2.2.5 Le diagramme d'états-transitions.....	28
2.2.6 Le diagramme d'activités.....	29
2.2.7 Le diagramme de composants.....	29

Sommaire

4.3.1 Description détaillée des classes d'objet et d'association	49
4.3.2 Diagramme de classe générale.....	51
5 Implémentation et outils de développement	53
5.1 Environnement logiciel.....	53
5.2 Delphi XE2.....	53
5.2.1 Propriétés de Delphi XE2	53
5.3 Microsoft SQL Server (alias MSSQL)	54
5.4 Microsoft Access 2003	54
6. Présentation des interfaces.....	55
6.1 Interface de login	55
6.2 Fenêtre principal après connexion.....	55
6.3 L'interface agriculteur	56
6.4 l'interface pour la gestion d'affiliation et de radiation.....	57
6.5 L'interface de caisse	57
7 Conclusion Générale.....	58

Liste des figures

Figure1 : Organigramme du service d'accueil	06
Figure 2. Application pour la gestion d'investissements agricoles.....	07
Figure 3. Application de caisse.....	08
Figure 4 Application pour la gestion d'affiliation et de radiation.....	08
Figure5 : Architecture de EAI	13
Figure 6: Dimensions de l'intégration d'application.....	15
Figure 7: Différentes techniques d'intégration syntaxique.....	18
Figure 8 : Principe de technique ad hoc	19
Figure 9 : Principe des intergiciels.....	20
Figure 10 : Principe des techniques d'EAI.....	22
Figure 11 : Architecture type d'un ESB	24
Figure 12 : Diagramme de cas d'utilisation pour le caissier	36
Figure 13 : Diagramme de cas d'utilisation pour le service investissement.....	37
Figure14 : Diagramme de cas d'utilisation pour le service d'affiliation et de radiation.....	38
Figure15 : Diagramme de cas d'utilisation pour l'Administrateur.....	39
Figure.16: Diagramme de séquence << Ajouter abonnement >>.....	40
Figure17 : Diagramme de séquence << Modifier abonnement >>.....	41
Figure18 : Diagramme de séquence << Supprimer abonnement >>.....	42
Figure19: Diagramme de séquence << gestion de Renouvellement carte agriculture >>.....	43
Figure20: Diagramme de séquence << nouveaux demande carte d'agriculture>>.....	44
Figure21: Diagramme de séquence << gestion Investissement >>.....	45
Figure22:: Diagramme de séquence << gestion radiation >>.....	46
Figure23 : Diagramme de séquence << gestion d'utilisateur >>.....	47
Figure24 : Diagramme de classe générale	51
Figure25 : Interface d'authentification.....	54
Figure26: L'interface Fenêtre principal après connexion	54
Figure27 : L'interface agriculteur.....	56
Figure28 : gestion d'affiliation et de radiation.....	57
Figure29 : L'interface de caisse.....	57

Liste des tableaux

Tableau1 : Identification des cas d'utilisation de chaque acteur.....	32
Tableau2 : Description détaillée des classes d'objet et d'association	49

Introduction Générale

Dans l'ère de l'information, l'intégration des applications d'entreprise (EAI pour Enterprise Application Intégration en Anglais) est devenue un cadre nécessaire pour la cohésion de l'entreprise. Elle permet à des applications hétérogènes de gérer leurs échanges au sein de la même entreprise ou à travers des entreprises éparpillées dans le cas de coopération inter-entreprises dans le cadre de l'intégration des affaires (business intégration).

Dans cette perspective, la chambre de l'agriculture aujourd'hui veut se bénéficier de cette technologie et d'utiliser les données de l'ensemble de ses infrastructures logicielles dans différents outils en parallèle tout en assurant une qualité et une fiabilité à toutes épreuves. De ce fait, nous allons essayer à travers ce projet d'intégrer les applications existantes dans la chambre de l'agriculture afin de garantir les points suivants :

- Restituer l'information à la bonne personne et au bon moment sous le format approprié.
- Rehausser l'interrelation des systèmes d'information avec la stratégie, l'organisation et la culture de l'entreprise
- Echanges Inter-applicatifs de données en permettant à plusieurs logiciels de communiquer entre eux et de gérer les différents échanges
- Créez un réseau local pour connecter les bases de données dispersées.
- Rassembler, router et synchroniser les données issues des différentes applications, ce qui permet d'éviter d'interfacer directement les applicatifs les uns avec les autres et créer ainsi une architecture complexe dans le cas de l'établissement de la chambre de l'agriculture.

Pour atteindre ces objectifs ce mémoire sera organisé comme suit : Le premier chapitre sera consacré à la présentation du domaine de l'EAI (Enterprise Application Intégration) afin de comprendre la situation de notre travail. Dans le deuxième chapitre, nous allons présenter la chambre de l'agriculture de la wilaya de M'sila avec ses applications que nous voulons les intégrer. Le travail de conception et de réalisation de cette tâche d'intégration sera présenté dans le troisième chapitre. Finalement nous clôturerons ce travail par une conclusion avec quelques suggestions permettant aux futurs développeurs d'améliorer le travail proposé.

CHAPITRE 1 PRESENTATION DE L'ORGANISME D'ACCUEIL

1 Introduction :

L'étude préalable est l'étape initiale de la conception d'un système d'information, elle est indispensable dans tout projet informatique, elle a pour but :

1. Définition de l'établissement
2. L'analyse du fonctionnement du système existant.
3. L'étude des postes de travail.
4. L'étude des documents et des procédures aux critiques et aux propositions pour améliorer le fonctionnement du système existant. Cette étape nous permet de spécifier les problèmes et les difficultés posées et tenir compte tous les éléments techniques et organisationnels de la structure.

2 Définition de l'établissement :

La chambre de l'agriculture est un établissement public à caractère industriel et commercial, placé sous la tutelle du Ministère de l'Agriculture.

La chambre de l'agriculture est dotée de la personnalité morale et de l'autonomie financière. Elle est le forum de représentation des intérêts de la profession agricole et constitue le partenaire privilégié des autorités administratives et techniques dans les domaines intéressant le développement agricole. [1]

2.1 Attributions de l'établissement :

La chambre de l'agriculture s'occupe de :

- ✓ Organisation et développement des formes de concertation, de coordination et d'information entre les adhérents et entre ceux-ci et les institutions publiques intervenant dans les sphères de la production, du financement, d'approvisionnement, de la distribution et de la transformation
- ✓ Représentation de ses adhérents auprès des pouvoirs publics pour toutes les matières en relation avec les missions de la chambre
- ✓ Contribution à l'élaboration et à la mise en œuvre de la politique nationale de développement des activités agricoles et à leur diversification.

- ✓ Défense des intérêts professionnels et sociaux de ses membres.
- ✓ Promouvoir la création de toute structure susceptible d'améliorer les performances des producteurs agricoles au plan de la production et de favoriser la fourniture des services dont ils ont besoin. [1]
- ✓ Organisation des foires, expositions, concours, et de faciliter la diffusion de l'information scientifique technique et économique en direction de ses membres
- ✓ La chambre de l'agriculture est obligatoirement consultée sur tout projet ayant des incidences sur les intérêts de ses membres en matière notamment d'aménagement de l'espace rural .Elle agit en tant qu'organe de consultation et force de proposition.

2.2 Organisation et fonctionnement :

La chambre de l'agriculture est organisée comme suit :

2.2.1 Le conseil de la chambre :

- Composition :

- ✓ Six (06) membres élus par bureau de wilaya d'associations professionnelles de producteurs agricoles.
- ✓ Membre élu par filière d'activité représentant les prestataires de services liés à l'agriculture.

- Fonctionnement

Le conseil de la chambre se réunit trimestriellement en session ordinaire sur convocation de son président.

Il peut se réunir en session extraordinaire sur convocation du président ou à la demande des deux tiers (2/3) au moins de ses membres.

- Prérogatives

Le conseil de la chambre oriente, suit et facilite la mise en œuvre du programme d'action de la chambre adopté par l'assemblée générale.

- ✓ Il élit les membres représentant la profession au conseil d'administration de la chambre.
- ✓ Il propose tout aménagement à l'organisation et au fonctionnement général de la chambre.
- ✓ Il peut désigner des correspondants sur toute l'étendue de la circonscription territoriale de la chambre, qui seront chargés d'éclairer la chambre sur la situation de

l'agriculture de leur ressort, ou de constituer des représentants de celle-ci pour des missions ponctuelles.

2.2.2 Le conseil administration :

✓ Composition :

- Neuf (09) membres élus par le conseil de la chambre parmi ses membres et dont deux tiers doivent être obligatoirement des producteurs.
- Deux (02) membres représentant le Ministre de l'agriculture.
- Deux (02) membres représentant le Ministre de l'économie.

✓ Fonctionnement :

Le conseil d'administration se réunit en session ordinaire une fois par mois.

Il peut se réunir en réunion extraordinaire sur convocation de son président, à la demande du Ministre de l'agriculture ou à la demande de la majorité de ses membres.

✓ Prérogatives :

Le conseil d'administration délibère notamment sur :

- Le programme et le bilan d'activité de la chambre
- Les projets de budgets et les comptes d'exploitations prévisionnels de la chambre.
- L'organigramme et le projet de règlement intérieur de la chambre.
- Les projets de contrats, accords et conventions conformément à la réglementation en vigueur.
- Les études effectuées ou à effectuer dans le cadre de la mission générale de la chambre.
- Les propositions de relations étrangères.
- L'acceptation des dons et legs conformément aux lois et règlements en vigueur.
- Les projets d'acquisition, d'aliénation et d'échange de biens immeubles dans le cadre de la législation en vigueur.
- L'approbation du rapport annuel d'activité ainsi que les comptes de gestion clos présentés par le secrétaire général.

✓ Le président :

- ✓ Il est élu par l'assemblée générale.
- ✓ Il convoque et préside l'assemblée générale.
- ✓ Il est président du conseil de la chambre et du conseil d'administration

✓ Le secrétaire général :

- Le secrétaire général assure la gestion et le fonctionnement de la chambre.
- Il assure ses pouvoirs sous le contrôle du conseil d'administration organe délibérant.
- Il est ordonnateur du budget de la chambre.
- Il représente la chambre en justice et dans les actes de la vie active.
- Il réalise les études nécessaires aux choix de la chambre en matière de planification d'investissement et de relations commerciales.

2.3 Présentation des services

✓ **Présentation et rôle du service caisse :**

- traitement des impayés annuels
- Renouvellement des cartes d'agriculteur
- Délivrance de la carte magnétique d'agriculteur

L'Organisation du service de la caisse est fondée sur les principes directeurs suivants :

- Le principe de l'exploitation: Il consiste à servir rapidement la clientèle et la renseigner avec exactitude.
- Le principe de sécurité : Il consiste à exercer des contrôles fréquents et à prendre des mesures de sécurité, telle que la vérification

✓ **Service d'affiliation et de radiation :**

L'affiliation : l'attestation d'affiliation est utilisée par l'agriculteur pour accéder à l'assurance sociale.

La radiation : se fait sur demande de l'agriculteur.

✓ **Service de Gestion des investissements Agricoles :**

- Gestion des investissements agricoles
- Réception des dossiers de cartes agricoles

2.4 Organigramme du personnel de la Chambre d'agriculture :

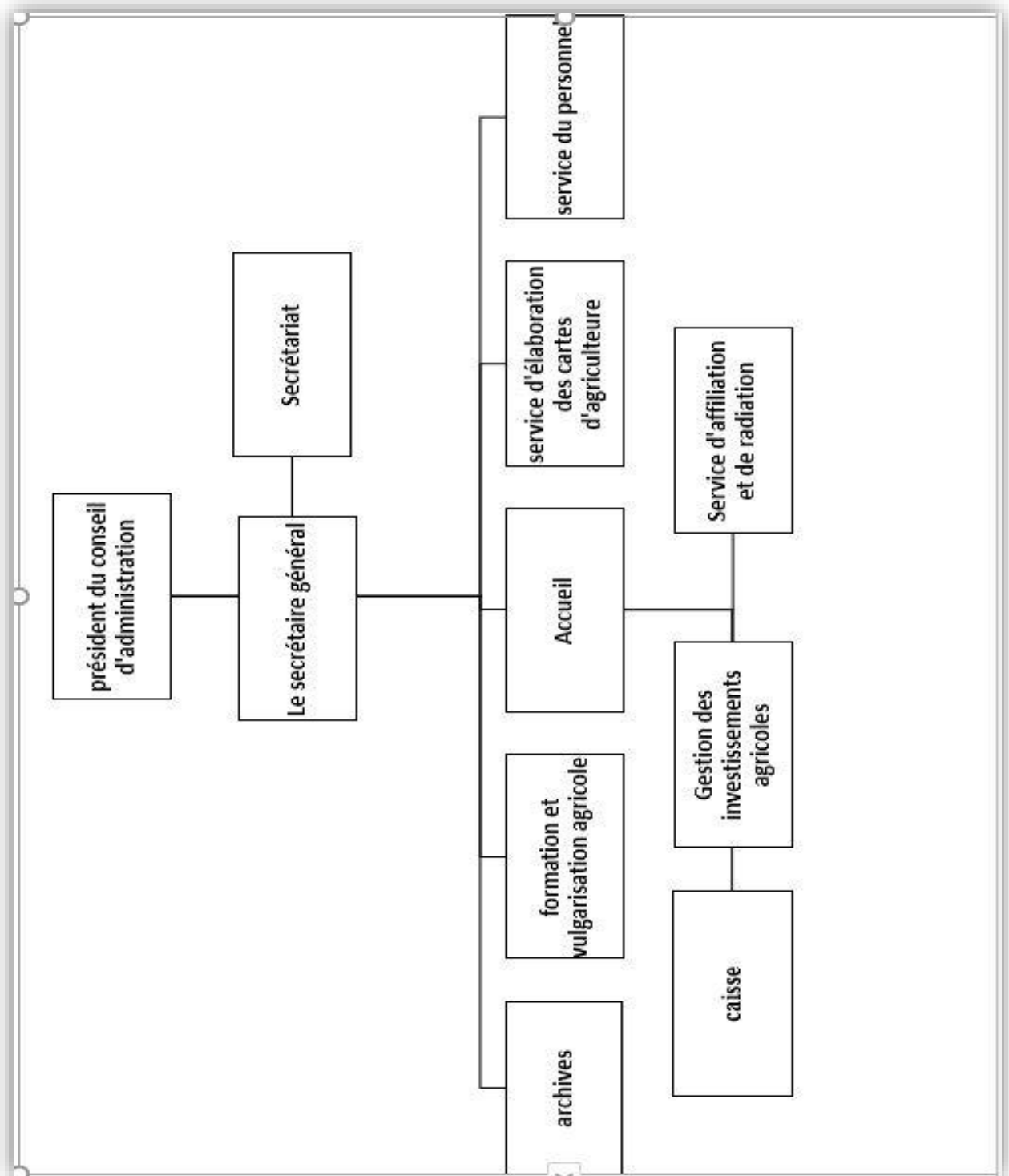


Figure 1. Organigramme du service d'accueil

2.5 Aspect logiciel:

Il existe dans la chambre de l'agriculture trois applications (voir figure 9, 10 et 11) :

- ✓ Une application pour les opérations menées dans la caisse
- ✓ Une application pour gérer les investissements agricoles
- ✓ Une application pour gérer le service d'affiliation et de radiation
- ✓ Les trois applications développées par Delphi/ Langage de programmation Object Pascal.
- ✓ La base de données utilisée est de type Access.

2.5.1 Les Problèmes existants:

- La base de données contient une seule table
- Les données s'écrasée
- Les données se répètent
- Les informations et les processus ne se partagent pas avec d'autres utilisateurs
- Perte du temps pour rechercher les informations d'un appareil à un autre

2.5.2 Application pour la gestion d'investissements agricoles



Figure 2. Application pour la gestion d'investissements agricoles

2.5.3 Application pour la caisse



Figure 3. Application de caisse

2.5.4 Application pour la gestion d'affiliation et de radiation.



Figure 4. Application pour la gestion d'affiliation et de radiation

2.6 Solutions :

A travers l'EAI (Enterprise Application Intégration), la chambre de l'agriculture peut résoudre plusieurs problèmes et promouvoir plusieurs fonctionnalités dont l'essentielles sont :

- Faire communiquer et collaborer les applications d'une même entreprise. Autrement dit, elle permet à une entreprise de gérer et d'étendre ses affaires inter-applicatives.
- Relier à moindre coût les applications internes et externes.
- Favoriser l'intégration fonctionnelle plutôt que l'intégration de donnée.
- Favoriser la réutilisation
- Relier plusieurs systèmes d'informations les uns aux autres aussi bien au niveau service qu'au niveau informationnel, en permettant ainsi le partage à la fois des informations et des processus.

3 Conclusion :

Après avoir présenté la chambre de l'agriculture avec ses services et ses applications, nous pouvons constater que l'intégration de ses applications est plus que nécessaire pour remédier aux insuffisances techniques et assurer plus de perfection fonctionnelle. Dans ce qui suit, nous allons aborder notre travail conceptuel et technique concernant l'intégration des applications existantes dans la chambre de l'agriculture.

CHAPITRE 2

INTEGRATION DES APPLICATIONS D'ENTREPRISE (EAI)

1 Introduction

Avec l'évolution de l'informatique est notamment des techniques réseaux, les entreprises cherchent actuellement à intégrer leurs applications isolées afin d'assurer plus de cohésion entre ces dernières et d'éviter les problèmes de redondances et de sécurité. Dès ce chapitre nous allons présenter le domaine d'intégration des applications d'entreprises ou Enterprise Application Intégration (EAI) avec ses spécificités, ses techniques et ses avantages.

2 Enterprise Application Intégration (EAI) :

Enterprise Application Intégration (EAI) pour le terme consacré en anglais, est aussi appelé Echanges Inter-applicatifs de Données en français. Ce terme désigne une architecture SI qui permet à plusieurs logiciels de communiquer entre eux et de gérer les différents échanges. On parle aussi de gestion des flux inter-applicatifs. Par raccourci, les outils logiciels middleware utilisés pour réaliser ces architectures EAI sont eux aussi appelés (logiciels EAI) [2].

Un logiciel EAI permet de se connecter à tout type de sources de données, d'en extraire des données, de manipuler ces données, de structurer ces données puis de les déposer dans toutes autres structures de données. L'exécution de ces manipulations et la synchronisation de données est programmable et déclenchée par tout type d'évènements.

2.1 Fonctionnalités

Les fonctionnalités de l'EAI sont triples :

- La connexion aux applications,
- La conversion des informations dans un langage commun,
- Le transport des informations, de l'application émettrice à l'application réceptrice.

2.2 Composants :

Pour que l'EAI fonctionne parfaitement il doit intégrer les éléments suivants [3]:

- Un référentiel des objets métier de l'entreprise (ou du process),
- Un moteur de gestion de règles,
- Des connecteurs applicatifs permettant l'interface avec les applications et les données de l'entreprise,
- Un système de transport des informations.

2.3 Avantages :

Les avantages de l'EAI sont multiples à savoir :

- Rationalisation de l'échange de données
- Réduction des coûts de main-d'oeuvre
- Economies de temps
- Réduction des erreurs humaines

3 Types d'EAI :

Il existe cinq types de base d'EAI qui rationalisent différentes formes d'intégration, à savoir [4]

3.1 Intégration point à point :

Comme l'un des modèles originaux, l'intégration point à point ne gère que les interactions simples entre deux sources de données. Dans ce type, un script est utilisé pour extraire, traduire et transférer des informations d'une application à une autre. Bien qu'il soit possible de gérer plusieurs systèmes, les connexions peuvent devenir ingérables et difficiles à entretenir.

3.2 Intégration des concentrateurs et des rayons :

L'intégration des rayons et des concentrateurs crée une base centralisée à laquelle d'autres applications peuvent se connecter, tout comme le concentrateur et les rayons d'une

roue. En établissant une base de données qui consolide toutes les données, les interconnexions entre les sous-systèmes sont éliminées. Bien que cela simplifie la maintenance, les développeurs de logiciels doivent tout de même vérifier la programmation pour s'assurer que les informations sont acheminées vers le bon emplacement.

3.3 Bus Intégration :

L'intégration Bus est une version avancée du modèle de concentrateur en rayons, car il peut fonctionner avec une interférence humaine limitée ou nulle. En utilisant un ensemble de normes et de protocoles, l'intégration de bus permet la transmission automatique des données tant que les applications sont conformes aux politiques.

3.4 Middleware :

Middleware est un outil logiciel qui s'installe entre les interfaces d'application et les systèmes d'exploitation pour traduire et prendre en charge la communication. Ce modèle simplifie la consolidation des entrées de données sur plusieurs systèmes avec différents scripts et formats. L'emplacement central de cet outil donne également aux développeurs la flexibilité quant au nombre d'applications pouvant être connectées. Il existe des versions simples de middleware qui relient deux ou trois systèmes, et des modèles cloud plus complexes qui intègrent virtuellement plusieurs logiciels disparates.

3.5 Micro services

Micro service est l'architecture standardisée des applications d'entreprise modernes basées sur le cloud. Les entreprises qui utilisent l'intégration basée sur le Web peuvent collecter des métadonnées en accédant à partir de sources internes et externes via des interfaces de programmation d'applications (API). Ces services étendent le partage d'informations au-delà des secteurs d'activité grâce au développement de logiciels complexes.

Une intégration complète des applications d'entreprise peut optimiser la communication, l'efficacité des rapports et les analyses basées sur les données. Toutefois, les entreprises doivent d'abord décrire leurs objectifs afin de déterminer quel modèle d'intégration améliorera leurs systèmes existants.

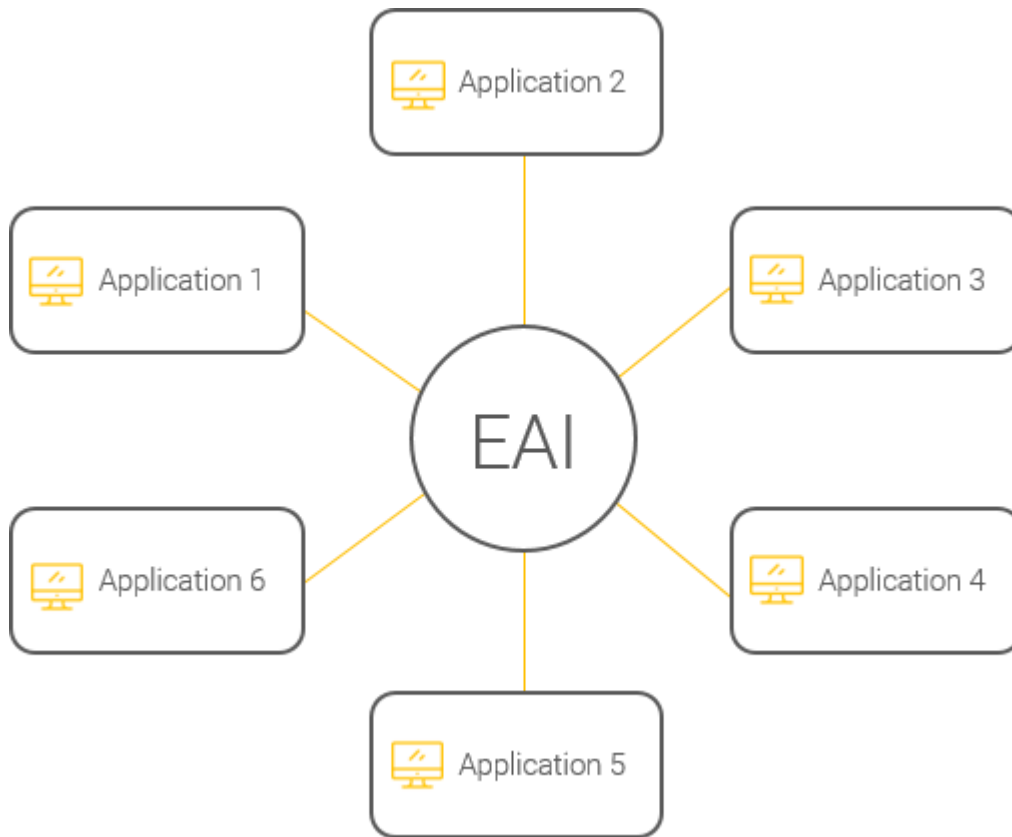


Figure5 : Architecture de EAI [5]

4 Typologie d’applications d’entreprise :

Les systèmes d’information et les applications dans une même entreprise sont très variés, nous distinguons dans la littérature plusieurs manières de classification des applications d’entreprise, selon le critère de classification choisi on distingue les applications selon leur nature : progiciels ou applications spécifiques [6], ainsi que leur style architectural : application 1-tiers ; 2-tiers ou application 3-tiers. Il a aussi dans [7] classifié les applications selon le mode d’exécution des événements en applications batch, applications transactionnelles, applications client/serveur, applications web, applications fil de l’eau et progiciels. Nous distinguons dans cette section, selon la nature des applications [8] :

4.1 Les applications compactées (Packaged Application) :

Ces applications contiennent un grand nombre de fonctions utiles pour le bon déroulement des activités de l’entreprise, comme : la gestion de stocks, de production, et des ressources humaines...etc. Exemple de ce type d’application : les progiciels intégrés

ERP (Enterprise Resource Planning), CRM (Customer Relationship Management) et les applications logistiques (SCM : Supply Chain Management).

4.2 Les bases de données patrimoniales (Legacy databases) :

Ce type d'application permet la gestion et le contrôle des données critiques pour les processus d'affaires de l'entreprise.

4.3 Les applications spécifiques (Custom applications) :

Sont les applications spécialisées développées par l'entreprise 'sur mesure' pour satisfaire ses besoins d'affaires, en utilisant différents langages de programmation, comme : C/C++, COBOL, ... etc. Les applications patrimoniales sont un exemple typique de ce genre d'applications.

4.4 Les applications Web :

Sont une forme particulière d'applications transactionnelles exécutées sur des technologies Web [9].

Dans le contexte de l'intégration, ces applications sont caractérisées par les propriétés suivantes :

- **Autonomie** : dans la mesure où chaque application peut être conçue et exécutée indépendamment des autres.
- **Distribution** : cela signifie que les applications sont très souvent réparties physiquement sur le réseau de l'entreprise.
- **Dynamisme** : dans la mesure où les applications doivent être capables d'évoluer de façon dynamique, pour faire face aux changements d'ordre organisationnel, fonctionnel et technique que subit l'entreprise.
- **Hétérogénéité** : cela signifie que les composants autonomes peuvent être développés en utilisant des approches et des méthodologies différentes[10], ce qui donne lieu à l'hétérogénéité des applications au niveau technique (les différences liées aux matériels et aux plateformes), au niveau syntaxique (les différences liées aux formats de données, aux signatures des fonctions d'une

applications) et au niveau sémantique (les différences liées au sens associé aux données et aux fonctions d'une application).

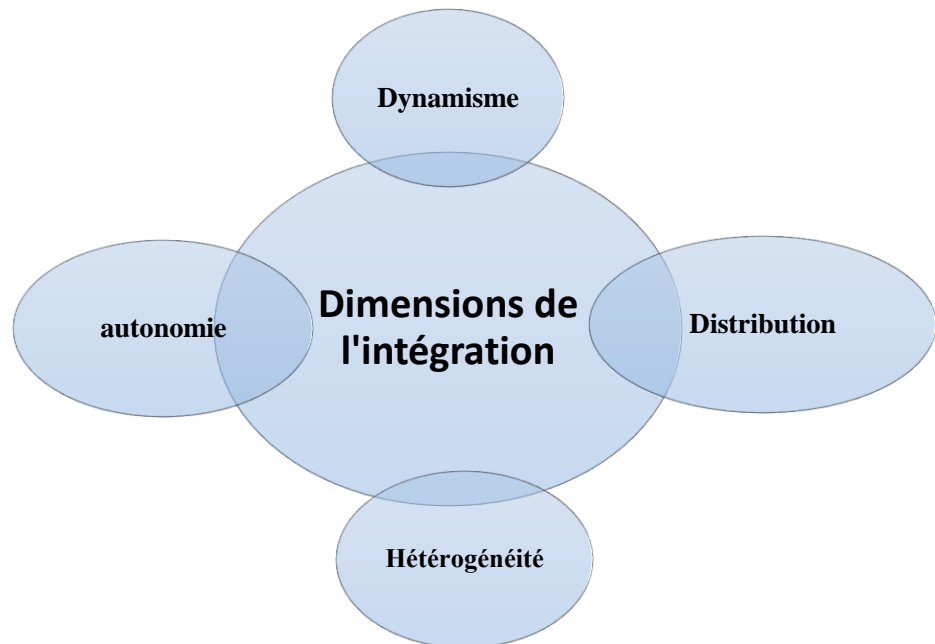


Figure 6: Dimensions de l'intégration

5 Niveaux D'intégration

L'intégration peut concerner un ou plusieurs niveaux de système d'information et des applications de l'entreprise. Linthicum [11] définit deux niveaux d'intégration : l'intégration au niveau des données et l'intégration au niveau des traitements et processus métiers. Ce dernière type peut être à son tour représenté par le niveau d'interfaces d'applications, le niveau des méthodes et le niveau des interfaces utilisateurs. De son côté Janarthanan et al [12] définit les trois niveaux d'intégration : données, interface utilisateur et méthode et considère le niveau API (Application Programming Interface) de comme étant le niveau d'application. Kang et Seth montrent la nécessité de choix d'une méthode d'intégration qui peut être réalisée à travers quatre niveaux d'intégration d'applications : intégration par données, par interface utilisateurs, par applications et par méthodes [13].[14].

En fait, toutes ces classifications d'intégration sont très similaires, et comme nous pouvons le remarquer, les approches d'intégration par données, intégration par interfaces

utilisateurs et intégration par méthodes sont communes à toutes les taxonomies présentées ci-dessus.

5.1 Intégration des données :

C'est la manière la plus répandue pour intégrer les applications de l'entreprise. [12] [7] L'intégration à ce niveau se réalise par la migration des données d'une base de données à une autre sans avoir besoin d'introduire des changements sur la logique des applications de l'entreprise. Ceci peut être décrit tant qu'extraction de l'information à partir d'une base de données, avec un traitement éventuel, la transformation et l'injection de cette information dans une autre base de données. Ce type d'intégration est utilisé dans le cas où les applications à intégrer n'ont pas une interface des utilisateurs et des API (Application Programming Interface) .

5.2 Intégration de l'interface utilisateur (présentation) :

Cette approche est aussi appelée screen-scraping [12] qui permet de relier la logique d'intégration dans l'interface des utilisateurs. L'intégration à ce niveau se fait par le développement d'une interface commune (partagée) pour exposer les différentes applications isolées en intra et inter-entreprises [7]

5.3 Intégration du processus métier :

L'intégration des applications par processus est une approche très intéressante, elle permet le partage de la logique d'affaires, ainsi que l'intégration de nouvelles applications à travers les applications existantes. [12] ,Cela peut être fait pour créer une nouvelle interface ou intégrer une nouvelle application. Afin de relier et intégrer les applications dans un processus commun.

6 Défis d'intégration des applications :

L'intégration des systèmes d'information et des applications dans les entreprises est très difficile, à cause de plusieurs problèmes situés à plusieurs niveaux [8]:

6.1 Niveau du support technologique :

Les systèmes d'information des entreprises sont différents dans le niveau d'avancement technologique. Nous prenons par exemple la différence dans le support fourni pour la gestion des transactions et la sécurité. Plusieurs systèmes d'information

sont des primitifs et n'offrent pas de supports pour l'accès transactionnel, et il existe d'autres plus avancés qui permettent un accès transactionnel aux ressources

6.2 Restrictions technologiques et administratives :

Plusieurs systèmes d'information imposent des restrictions technologiques et administratives spécifiques aux utilisateurs. Par exemple l'utilisation des systèmes patrimoniaux est très compliquée (l'ajout d'un nouveau client à une base de données d'un système patrimonial est très difficile). Donc, les entreprises qui possèdent ce type de système doivent l'adapter avec les nouveaux systèmes, comme il faut les intégrer avec les différentes applications de l'entreprise.

6.3 Aptitude de l'intégration avec les autres systèmes :

Les systèmes d'information diffèrent dans leurs modèles de programmation et APIs, ce qui rend leur intégration très difficile, parce que chacun d'eux a été développé dans son propre environnement, et qui ne peut pas être conforme avec les autres systèmes, vu qu'ils n'ont pas les mêmes objectifs.

6.4 Difficulté de compréhension des systèmes :

Avant que le développeur démarre l'intégration des applications, il faut d'abord qu'il comprenne tous les détails de bas niveau de programmation des systèmes d'information. Si nous avons, par exemple, un système d'information et que son API client est défini dans une bibliothèque C, et que cette dernière a défini des applications clients utilisées pour gérer les transactions et qu'elle a effectué l'accès transactionnel au système d'information, alors il faudra que le développeur étudie aussi cette bibliothèque, pour pouvoir utiliser le client API de ce système.

7 Technologies d'intégration d'applications :

L'intégration des applications constitue un domaine prometteur en constante évolution. [7] Il existe principalement sur le marché plusieurs techniques d'intégration qui permettent d'adresser l'intégration syntaxique, et qui peuvent être mise en œuvre au sein des entreprises ou plus communément au sein des organisations.

es technologies sont aussi diverses que variées, nous pouvons dans cette partie distinguer six grandes classes, qui sont selon [6]: les techniques réseau ad-hoc de communication ; les techniques d'unification et de standardisation ; les techniques basées sur les intergiciels de communication (Middleware) ; les techniques basées sur l'EAI (Enterprise Application Intégration); les techniques basées sur le

moteur BPM (Business Process Management) ; et les techniques basées sur l’ESB (Enterprise Service Bus). La figure est un méta modèle UML permettant de résumer les principales technologies d’intégration syntaxique. Le méta modèle est un diagramme de classe. Chaque classe représente une catégorie de technique.

Après avoir présenté le principe de chaque technologie, nous allons terminer cette section par une synthèse des avantages et des inconvénients de chacune des technologies étudiées, avec une comparaison entre elles.

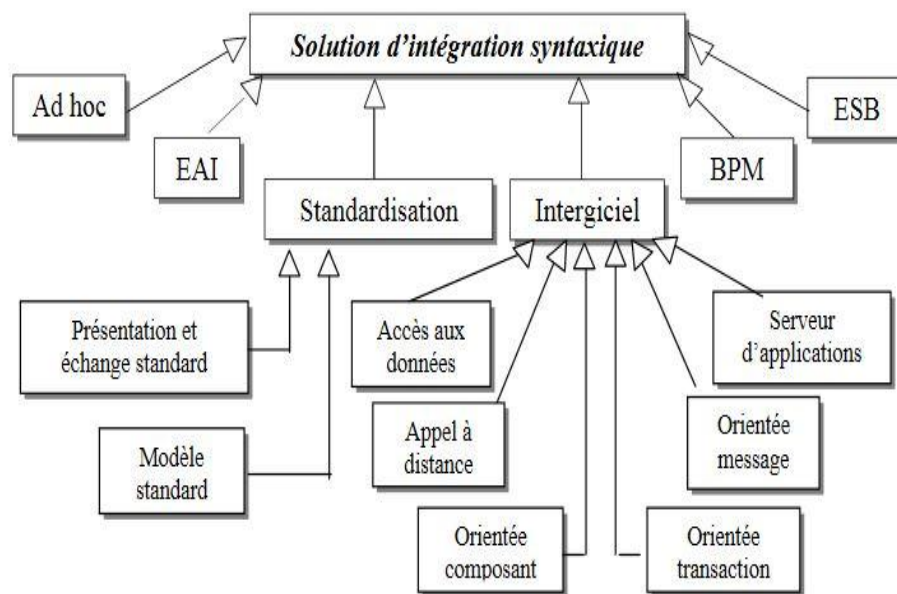


Figure 7 : Différentes techniques d’intégration syntaxique [6]

7.1 Techniques basées sur les réseaux ad hoc :

Ces techniques sont utilisées, dont le besoin principal, est de faire communiquer les différentes applications qui envahissent l’entreprise pour : supprimer les saisies multiples, consolider les informations, et réduire les tâches administratives [20]. Ces technologies se basent principalement sur certains middlewares invoqués de manière programmatique pour définir des passerelles entre les applications [6], comme le RPC (Remonte Procédure Call) qui consiste à mettre en œuvre des appels à distance entre les procédures des applications. Dans cette approche, les applications sont connectées directement les unes aux autres, et communiquent en face à face en n’utilisant pas de structures évoluées d’intégration intermédiaires, ce qui donne lieu à une architecture accidentelle difficile à maintenir. (Voir figure.8)

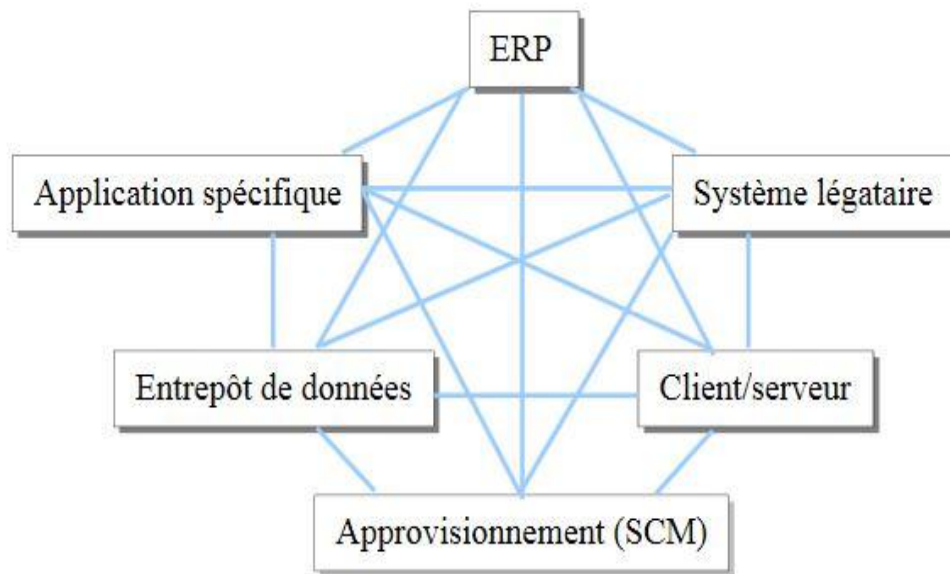


Figure 8 : Principe de technique ad hoc [6]

7.2 Techniques d'unification et de standardisation :

Dans l'entreprise, on utilise une multitude de langages qui permettent de représenter les modèles de données, les modèles de processus, ou encore d'échange de données. Une des solutions très simple pour réaliser l'intégration entre applications, est d'unifier la manière de présenter et d'échanger les données en utilisant des standards de représentation [6] tels que :

UML (Unifie Modeling Language) et XSD (XML Schema Definition) qui sont généralement utilisés pour décrire les schémas de données et des bases de données, EDI (Exchange Data Interface), XML (eXtensible Markup Language) et ebXML (e-business XML) pour l'échange de données intra et inter-entreprises. Des standards des modèles de processus et des applications sont aussi utilisés, comme : UEML (UnifiedEnterprise Modelling Language) qui est un langage de modélisation d'entreprise [15] et BPML (Business Process Modeling Language) qui est un langage unifié de modélisation de processus métiers de l'entreprise [6].

Les techniques de standardisation offrent, en effet, un moyen d'entente aux partenaires, qui présentent leurs entreprises selon des formalismes de modélisation incompatibles et qui ne couvrent pas les mêmes vues.

Ces techniques permettent aussi de standardiser les outils, méthodes et stratégies pour faciliter l'intégration optimale des applications hétérogènes. Cependant elles n'ont pas

réussi à s'imposer à grande échelle, car l'échange standardisé s'effectue sur un mode point à point.

7.3 Techniques basées sur les intergiciels :

Les inconvénients des techniques de communication point à point et le besoin des entreprises à améliorer leurs réactivités sur le marché, ont poussé l'entreprise à utiliser des systèmes intermédiaires appelés : intergiciels. [16] Les intergiciels (en Anglais middleware) sont des logiciels qui jouent le rôle d'un intermédiaire dédié pour la gestion des communications inter applications. Ils se chargent de la communication, du routage des messages entre les applications, de la transformation des données, la gestion des transactions, et le partage des connexions [11].

La figure 5 illustre le principe des intergiciels, qui permettent de masquer toute l'hétérogénéité technique liée aux plateformes (langage de programmation, réseau, système d'exploitation) [6].

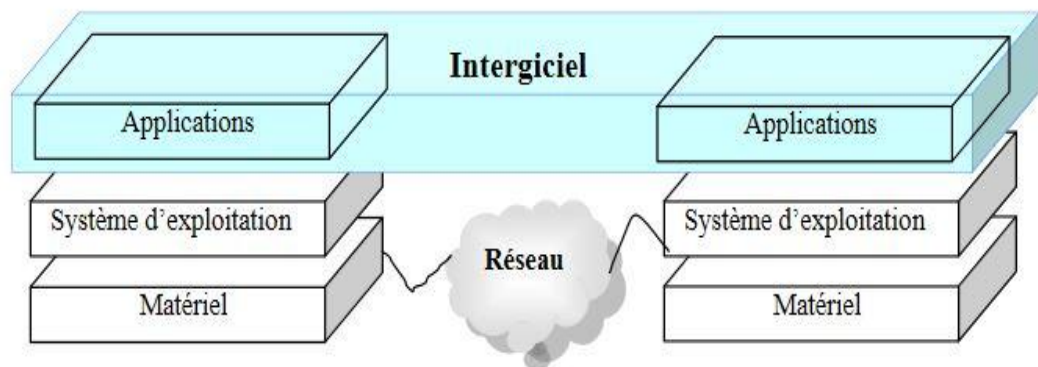


Figure 9 : Principe des intergiciels [6]

La notion de middleware est aujourd'hui plus évoluée, elle inclue d'autres services de haut niveau, tels que la gestion des processus, la sécurité, le management des échanges (administration et exploitation), la qualité de services (performances, disponibilité, etc.) [3]. Il existe plusieurs types de middlewares, parmi lesquels nous citons

- Les middlewares d'accès aux bases de données : permettant d'accéder à des données de manière transparente, quel que soit le type de la source de données (DB2, Oracle, ...), exemple : ODBC et JDBC.
- Les middlewares d'appel de procédures à distance : permettant d'accéder et d'exécuter des programmes (procédures) sur un site distant à travers le réseau de manière transparente, en utilisant une communication synchrone.

- Les middlewares orientés messages : permettant l'échange des messages en utilisant une communication asynchrone.
- Les middlewares orientés composants : ce type d'intergiciel offre une solution d'intégration distribuée, en exploitant les objets distribués et le principe d'appel à des méthodes distantes. Un exemple de ce type : CORBA (Common Object Request Broker Architecture) et COM/DCOM (Component Object Model).
- Les middlewares orientés transactions : permettent principalement de gérer les accès aux bases de données de façon à garantir l'intégrité des transactions qui y sont effectuées, donc à assurer la consistance des données partagées.
- Les serveurs d'applications : ce type d'intergiciel fournit un ensemble de services d'exécution aux composants déployés (applications). Ces services d'exécution sont cachés dans des composants d'applications du modèle de programmation simplifié. Les services fournis incluent : le support de transaction, le mécanisme de sécurité, l'accès aux bases de données, la messagerie asynchrone et la communication distribuée

Un exemple d'un serveur d'applications la plateforme Java 2 Edition Entreprise (J2EE). Qui utilise les connecteurs JCA (Java Connector Architecture) pour se connecter aux applications distribuées [8]

7.4 Techniques basées sur l'EAI :

Ce sont des techniques où chaque application se connecte de manière indépendante et unique à un « système central », sans avoir de connaissances à priori de typologie de l'application ou du système destinataire (voir figure10) [17]. Ce système est à l'origine, un progiciel EAI (Enterprise Application Integration), qui joue le rôle d'un médiateur. Il est responsable de plusieurs fonctions : interfacier les applications généralement par l'utilisation des connecteurs propriétaires difficilement maintenables, transformer les messages à un format pivot pour le "mapper" ensuite au format propriétaire attendu par les applications, rediriger (Routing) le message vers une application destinataire en fonction de son contexte, et contrôler l'exécution de processus métiers d'un système d'information de l'entreprise[18]

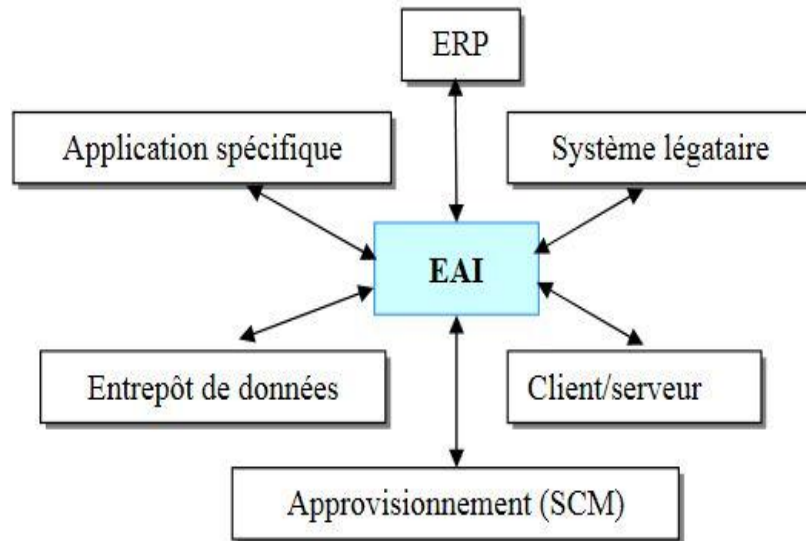


Figure 10 : Principe des techniques d'EAI [6]

L'architecture type de l'EAI permettant d'assurer ces fonctionnalités, est représentée par l'architecture en couche suivante où chaque couche est responsable d'un service d'intégration :

- La couche connexion des connecteurs : assure le transport des messages depuis l'EAI aux applications et vice versa, c'est-à-dire, l'extraction depuis les applications productrices ou la restitution aux applications consommatrices, L'EAI utilise typiquement MOM (Message Oriented Middleware) comme un moyen fondamental de transport de messages entre les différentes applications et la couche de transformation et de routage, et le HTTP, SOAP, SMTP, IIOP et RMI comme protocole de transport
- La couche transformation et interprétation des données : cette couche fournit des services permettant la conversion des données d'un format à un autre, par exemple le moteur XSL qui effectue des transformations des messages XML
- La couche de routage des informations : fournit des moteurs d'intégration pouvant déterminer le routage des messages intelligemment basé sur des règles prédéfinies et envoyer les messages à la cible nécessaire
- La couche modélisation de processus métier : fournit les outils nécessaires à la modélisation des processus métiers sous une forme permettant leurs automatisations, en organisant les enchaînements des tâches entre les applications

7.5 Techniques basées sur le BPM :

Les outils basés sur la gestion de processus métiers (GPM27) ne sont pas nouveaux. Ils sont déjà utilisés et vus incorporés dans les outils EAI. Ceci ne veut pas dire que ces derniers ne peuvent pas être considérés comme des outils d'intégration à part entière[7]. Les techniques de GPM sont des solutions basées sur la notion de processus métier. Elles ont été proposées pour répondre à la fois à un besoin fort des entreprises de bien formaliser, gérer et contrôler le processus métier de leurs systèmes d'information dans leur globalité, ainsi qu'un besoin d'intégration avec les divers composants et progiciels, qui sont utilisés tout au long du cycle de vie d'un produit, d'une relation avec un client ou d'une relation avec un fournisseur

6.6 Techniques basées sur l'ESB :

Les faiblesses des solutions basées sur les outils d'EAI, ainsi que l'émergence de nouvelles approches orientées services comme les services web et SOA ont modifiés le paysage de l'intégration [17], De nouvelles technologies appelées l'ESB (Enterprise Service Bus : en français Bus de services d'entreprise) ont été défini en 2003 par le Gartner Group [19], dont l'objectif principal est de définir un nouveau type de middleware d'intégration d'applications. la technologie ESB est centrée sur la notion de bus, qui permet d'assurer une intégration distribuée des différents services métiers connectés (les fournisseurs et les consommateurs de services) [15].

« L'ESB est une nouvelle architecture qui exploite les services web et les normes des services web dite « WS. *30 », le middleware de la messagerie, le routage intelligent et transformation. Les ESB agissent comme des composants légers, intégration omniprésente par laquelle les services logiciels et les composants d'application émergent » [13] (Voir figure.11)

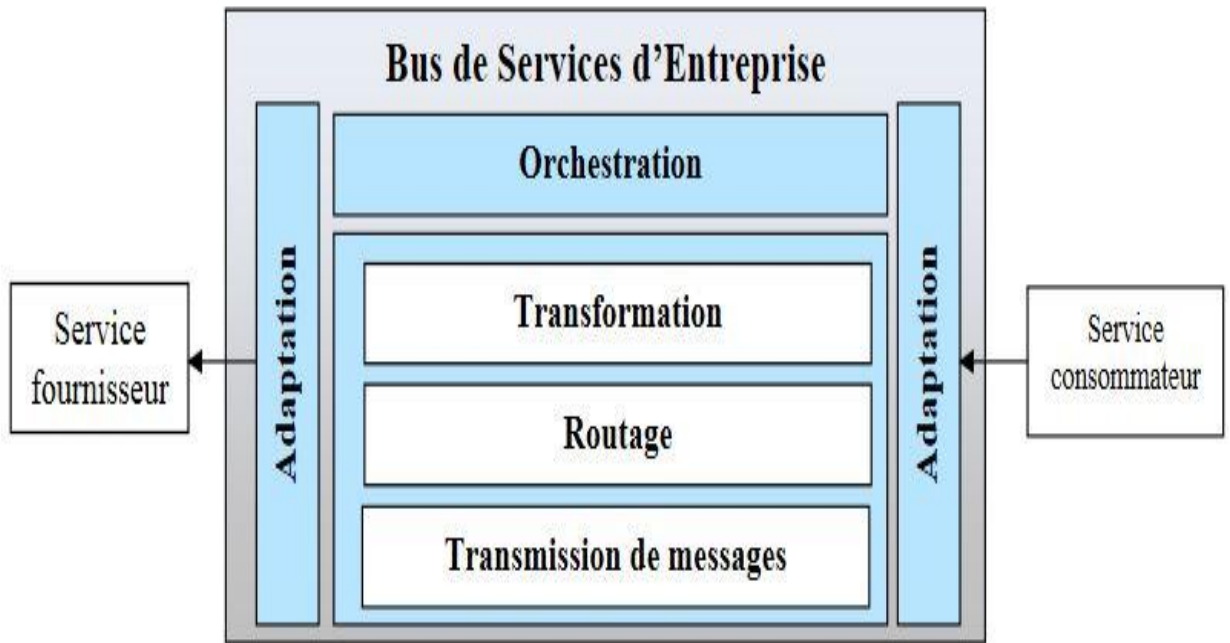


Figure 11 : Architecture type d'un ESB[13]

La figure.11 représente les principales fonctionnalités d'un ESB, et les composants permettant de les fournir : [20]

- Transmission de messages : une demande d'un service (message) est transmise en mode asynchrone au fournisseur de service.
- Routage : permet d'envoyer la demande de service au fournisseur de service nécessaire, en utilisant des règles prédéfinies.
- Transformation : permet de transformer le format de message à un autre format.
- Adaptateur : généralement dans l'ESB les applications utilisent le standard SOAP pour l'échange de messages (un format standard). Cependant, il existe des applications qui ne supportent pas le protocole SOAP, ceci nécessite l'utilisation des adaptateurs pour transformer le message
- Orchestration : un moteur d'orchestration permet de gérer et contrôler les flux de contrôle d'un service à un autre, constituant les processus métiers de l'entreprise

Conclusion

Nous avons présenté dans ce chapitre le domaine de l'EAI qui constitue une obligation pour plusieurs entreprises pour bien mener leurs tâches. Notre objectif est de réaliser une intégration entre les applications dispatchées de la chambre d'agriculture de la wilaya de M'Sila afin de bien superviser les différentes activités des administrateurs et des agriculteurs. Dans ce qui suit, nous allons présenter notre organisme d'accueil avec la phase conceptuelle concernant la modélisation de l'intégration.

CHAPITRE 3

CONCEPTION ET REALISATION

1 Introduction :

Notre travail consiste à développer une application qui permettra la communication et la collaboration de plusieurs applications au sein de la chambre agricole. De ce fait, nous commencerons par présenter les fonctionnalités de cette application ensuite nous consacrerons une large partie pour la phase d'analyse et de conception en élaborant les différents diagrammes UML de cette dernière. Par conséquent, la réalisation sera entamée après cette phase en présentant les différents outils utilisés avec le résultat final de notre système.

2 Présentation d'UML :

UML est un langage de modélisation très complet, qui couvre de nombreux aspects du développement des logiciels, comme les exigences, l'architecture, les structures et les comportements. Depuis sa normalisation, en 1997, UML a fortement évolué, passant d'un langage peu formel, principalement destiné à la documentation, à un langage suffisamment précis pour que des applications puissent être générées à partir des modèles. Cette évolution vers une plus grande précision a cependant créé une césure entre les tenants du « tout-modèle », qui demandent toujours plus de formalisme, et les développeurs, qui apprécient UML pour sa capacité à capturer en quelques dessins les grandes lignes d'une application. Le mieux étant l'ennemi du bien, pour satisfaire les uns, il a fallu complexifier UML au-delà du besoin des autres. En pratique, l'effort de formalisation et d'abstraction requis par une utilisation complète du langage de modélisation peut souvent s'avérer contre-productif lorsque l'écriture de code est l'objectif immédiat. Dans ce chapitre, on présente une approche de développement de logiciels dans laquelle modélisation et programmation se complètent harmonieusement. Cette démarche permet aux développeurs de bénéficier tout de suite d'une large part des avantages de la modélisation avec UML, tout en restant dans le monde de la programmation [21].

2.1 Les briques de base d'UML :

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage. UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis l'expression de besoin jusqu'au codage. Dans ce cadre, un concept appartenant aux exigences des utilisateurs projette sa réalité dans le modèle de conception et dans le codage. Le fil tendu entre les différentes étapes de construction permet alors de remonter du code aux besoins et d'en comprendre les tenants et les aboutissants. En d'autres termes, on peut retrouver la nécessité d'un bloc de code en se référant à son origine dans le modèle des besoins. [22]

- La terminologie d'UML inclut trois sortes de briques :

- Les éléments.

- Les relations.

- Les diagrammes.

Les éléments sont les abstractions essentielles à un modèle. Les relations constituent les liens entre ces éléments et les diagrammes les regroupent en des ensembles dignes d'intérêt.

2.2 Les diagrammes dans UML :

UML définit neuf sortes de diagrammes pour représenter les différents points de vue de modélisation. Un diagramme donne à l'utilisateur un moyen de visualiser et de manipuler des éléments de modélisation. Les différents types de diagrammes d'UML sont présentés dans l'extrait du méta modèle suivant [21] :

2.2.1. Les diagrammes des classes :

Il représente un ensemble de classes, d'interfaces et de collaborations, ainsi que leurs relations. C'est le diagramme le plus fréquent dans la modélisation des systèmes orientés objet. Il présente la vue de conception statique d'un système

2.2.2 Le diagramme d'objets :

Il représente un ensemble d'objets et leurs relations. C'est probablement le diagramme le moins utilisé d'UML.

2.2.3 Le diagramme des cas d'utilisation :

Il représente un ensemble de cas d'utilisation et d'acteurs et leurs relations. Il est particulièrement important dans l'organisation et la modélisation des comportements d'un système.

2.2.4 Le diagramme de séquence et le diagramme de collaboration sont :

Deux types de diagrammes d'interaction. Les diagrammes d'interaction représentent une interaction, c'est-à-dire un ensemble d'objets et leurs relations, y compris les systèmes. Les diagrammes de séquence sont des diagrammes d'interaction qui mettent l'accent sur le classement chronologique des messages. Alors que les diagrammes de collaboration sont des diagrammes d'interaction qui mettent l'accent sur l'organisation structurelle des objets qui envoient et reçoivent des messages [21].

2.2.5 Le diagramme d'états-transitions :

C'est un automate à états finis, composé d'états, de transitions, d'événements et d'activités. Il présente la vue dynamique d'un système, il est particulièrement important dans la modélisation du comportement d'une interface, d'une classe ou d'une collaboration.

2.2.6 Le diagramme d'activités :

Il décrit la succession des activités au sein d'un système. Il présente la vue dynamique d'un système, il est particulièrement important dans la modélisation de la fonction d'un système et met l'accent sur le flot de contrôle entre les objets.

2.2.7 Le diagramme de composants :

Il représente l'organisation et les dépendances au sein d'un ensemble de composants. Il présente la vue d'implémentation statique d'un système.

2.2.8 Le diagramme de déploiement :

Il représente la configuration des nœuds de processus en phase d'exécution ainsi que les composants qui y résident.

Dans ce chapitre, nous travaillerons avec le diagramme des classes, les diagrammes des cas d'utilisation et les diagrammes de séquence.

2.3 Les règles d'UML :

Les briques de base d'UML ne peuvent être assemblées au hasard. Comme tout langage, UML possède un certain nombre de règles qui précisent à quoi doit ressembler un modèle correctement mis en forme. Un modèle correctement mis en forme est un modèle sémantiquement cohérent. ML possède des règles sémantiques pour [23] :

- Les noms : la manière de désigner les éléments, les relations et les diagrammes.
- Le contexte: l'environnement qui donne une signification bien précise à un nom.
- La visibilité :la manière dont ces noms peuvent être vus et utilisés par d'autres.
- L'intégrité : la manière dont les objets établissent des relations correctes et cohérentes entre eux.
- L'exécution : les conséquences de l'exécution ou de la simulation d'un modèle dynamique.

2.4 Les mécanismes généraux d'UML :

UML a quatre mécanismes généraux qui s'appliquent d'une manière cohérente à l'ensemble du langage, le rendant plus simple à utiliser :

- Les spécifications
- . -Les décorations

. -Les distinctions communes.

-Les mécanismes d'extensibilités

2.4.1 Les spécifications :

UML est plus qu'un simple langage graphique. En fait, derrière chaque élément de sa notation graphique ; il existe une spécification qui fournit un énoncé textuel de la syntaxe et de la sémantique de cette brique de base, par exemple, derrière une icône de classe se trouve une spécification qui précise l'ensemble des attributs et des comportements mobilisés par cette classe [23].

2.4.2 Les décorations :

La plupart des éléments d'UML possèdent une notation graphique unique et directe qui fournit une représentation visuelle des aspects les plus importants de ces éléments. La notation d'une classe peut inclure d'autres informations, comme la visibilité de ses attributs et ses opérations, ou encore indiquer si elle est abstraite ou non. La plupart de ces informations peuvent être représentées par des décorations graphiques ou textuelles ajoutées au rectangle de base de la classe.

2.4.3 Les distinctions communes : Dans la modélisation des systèmes orientés objets, il y'a une distinction entre classe et objet. Une classe est une abstraction alors qu'un objet est la manifestation concrète de cette abstraction. En UML on peut modéliser aussi bien les classes que les objets. En UML on distingue un objet en soulignant son nom et en utilisant le même symbole que pour sa classe. En UML l'interface et l'implémentation sont séparées. Une interface définit un contrat, et l'implémentation représente la réalisation concrète de ce contrat respectant fidèlement la sémantique complète de l'implémentation.

2.4.4 Les mécanismes d'extensibilité :

Les concepteurs d'UML ont souhaité donné aux techniques de modélisation un caractère extensible. Cette volonté se traduit par les mécanismes d'extensibilités qui sont un moyen d'étendre UML de manière contrôlée à l'aide des stéréotypes, étiquettes et contraintes. A-
Stéréotypes : un stéréotype est une extension du vocabulaire d'UML, il permet d'associer une

nouvelle signification à un élément du modèle ; on représente un stéréotype par une chaîne de caractères entre guillemets («») [21].

2.5 Etiquettes :

Une étiquette est une extension des propriétés d'un élément, elle permet la description d'une nouvelle propriété d'un élément du modèle. On la représente par une chaîne de caractères entre chevrons (<>).
c-Contraintes : une contrainte est une extension de la sémantique d'UML. Elle prescrit une règle que le modèle doit vérifier pour être qualifié de <bien formé>. Elle est représentée par une chaîne de caractères entre accolades ({}). Pour conclure cette partie, on dira qu'UML possède plusieurs facettes. C'est une norme, un langage de modélisation objet, un support de communication, un cadre méthodologique. UML est tout cela à la fois. Il comble une lacune importante des technologies objet. Il permet d'exprimer et d'élaborer des modèles objet, indépendamment de tout langage de programmation. Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation des solutions. L'aspect formel de sa notation limite les ambiguïtés et les incompréhensions. Et enfin, son indépendance par rapport aux langages de programmation, aux domaines d'application et aux processus, en fait un langage universel [21].

3 Analyse et conception du cas étudié :

Cette partie a pour objectif la spécification de manière claire la gestion de notre établissement. Pour ce faire, il est nécessaire de déterminer globalement ce qui se trouve dans le champ de l'application. On s'intéressera dans cette phase à la spécification des besoins [22].

3.1 Spécification des besoins :

Définition : C'est une étape primordiale au début de chaque démarche de développement. Son but est de veiller à développer un logiciel adéquat, sa finalité est la description générale des fonctionnalités du système.

Notre système doit répondre aux exigences suivantes :

- Récupérer les informations de chaque entité à partir de son matricule pour mettre à jour la base de données de l'application.

- Modification des informations à propos de l'agriculteur et des autres Investissements agricoles
- L'accessibilité à tout moment et efficacement à toutes les données
- Permettre à l'administrateur de gérer la base de données
- Partager l'application dans un réseau local

3.2 Identification des acteurs :

Définition : Un acteur représente un ensemble de rôles joué par des entités externes (utilisateur humain, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. Dans le cas de notre application, nous avons quatre acteurs qui interagissent avec le système [23] :

- Administrateur : la personne chargée de l'administration de la base de données.
- Le caissier : Renouvellement des cartes agricoles et traitement des impayés
- Chef service d'affiliation et de radiation : gérer le service d'affiliation et de radiation
- Chef service investissement :
 - ✓ Gérer les investissements agricoles
 - ✓ Gérer la réception des dossiers de cartes agriculteur

3.3 Identification des cas d'utilisation :

Un cas d'utilisation représente un ensemble de séquences qui sont réalisées par le système en réponse à une action d'un acteur et qui produisent un résultat observable. Les cas d'utilisation décrivent ce que le système devra faire sans spécifier comment le faire.

- Le tableau suivant récapitule les cas d'utilisation de chaque acteur de notre application [21].

Désignation	Tache
Le caissier	- s'identifier - Renouveler l'abonnement annuel des cartes d'agriculteur

	<ul style="list-style-type: none"> - Renouveler les cartes d'agriculteur et traiter les impayés - validation des nouvelles demandes de cartes d'agriculteur - faire le bilan quotidien - Délivrance de la carte magnétique d'agriculteur
chef service investissement	<ul style="list-style-type: none"> - s'identifier - Gestion des investissements agricoles - Réception et vérification des dossiers des cartes d'agriculteur -classement des dossiers des cartes d'agriculteur
chef service d'affiliation et de radiation	<ul style="list-style-type: none"> -s'identifier - gérer les opérations d'affiliation et de radiation
Administrateur	<ul style="list-style-type: none"> - s'identifier - Accéder à l'interface administrateur - gérer la base de données et les utilisateurs

Tableau1 : Identification des cas d'utilisation de chaque acteur

4 La conception

Le processus de conception de notre projet se caractérise par deux niveaux : le niveau applicatif et le niveau données. Le niveau applicatif s'appuie essentiellement sur quelques diagrammes de l'extension du langage de modélisation UML. A cet effet nous avons adopté la démarche suivante :

- Après l'identification des différents acteurs ainsi que les cas d'utilisation qui sont mis en œuvre par ces acteurs, le diagramme de cas d'utilisation est élaboré
- Chaque cas d'utilisation se traduit par un ou plusieurs scénarios. Chaque scénario fait l'objet d'une description sous forme graphique à l'aide d'un diagramme de séquence et un diagramme d'activité
- Une identification des classes est fournie par la synthèse des diagrammes de séquence, ainsi que le diagramme de classe sera élaboré

Le niveau donné concerne l'organisation conceptuelle, logique et physique des données manipulées. Durant la partie analyse nous avons pu identifier les données nécessaires et indispensables au bon fonctionnement de l'application et à travers la conception du niveau applicatif nous allons dégager les classes significatives, dès lors on peut élaborer la conception de la base de données.

4.1 Le niveau applicatif :

4.1.1 Le diagramme des cas d'utilisation :

Lors de la phase d'analyse nous avons pu identifier les acteurs ainsi que les cas d'utilisation associés à ces derniers. Ce qui nous donne l'opportunité d'élaborer le diagramme des cas d'utilisation

Définition : Les diagrammes de cas d'utilisation permettent de représenter un ensemble de cas d'utilisation, d'acteurs et leurs relations. Ils présentent la vue statique des cas d'utilisation d'un système et sont particulièrement importants dans l'organisation et la modélisation des comportements d'un système.

-La relation d'inclusion (include) : elle indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. Cette relation permet de décomposer des comportements et de définir les comportements partageables entre plusieurs cas d'utilisations

-La relation d'extension (Extend) : elle indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à des conditions [21].

4.1.2 Cas d'utilisation relatifs au caissier :

- Page d'accueil
- S'authentifier
- Interface caissier

- Mise à jour Renouveler l'abonnement des cartes d'agricultures :
 - * Ajouter un abonnement.
 - * Modifier un abonnement.
 - * Supprimer un abonnement.

- Mise à jour des renouvellements des cartes d'agriculteurs :
 - * Ajouter. Renouvellement
 - * Modifier. Renouvellement
 - * Supprimer Renouvellement

- Validation des nouvelles demandes des cartes des agriculteurs :
 - * Ajouter. Nouvelle demande carte
 - * Modifier. Nouvelle demande carte
 - * Supprimer Nouvelle demande carte

- Délivrance de la carte magnétique d'agriculteur :
 - * Ajouter. Carte magnétique délivrée
 - * Modifier. Carte magnétique délivrée
 - * Supprimer carte magnétique délivrée

4.1.3 présentation du diagramme de cas d'utilisation pour le caissier :

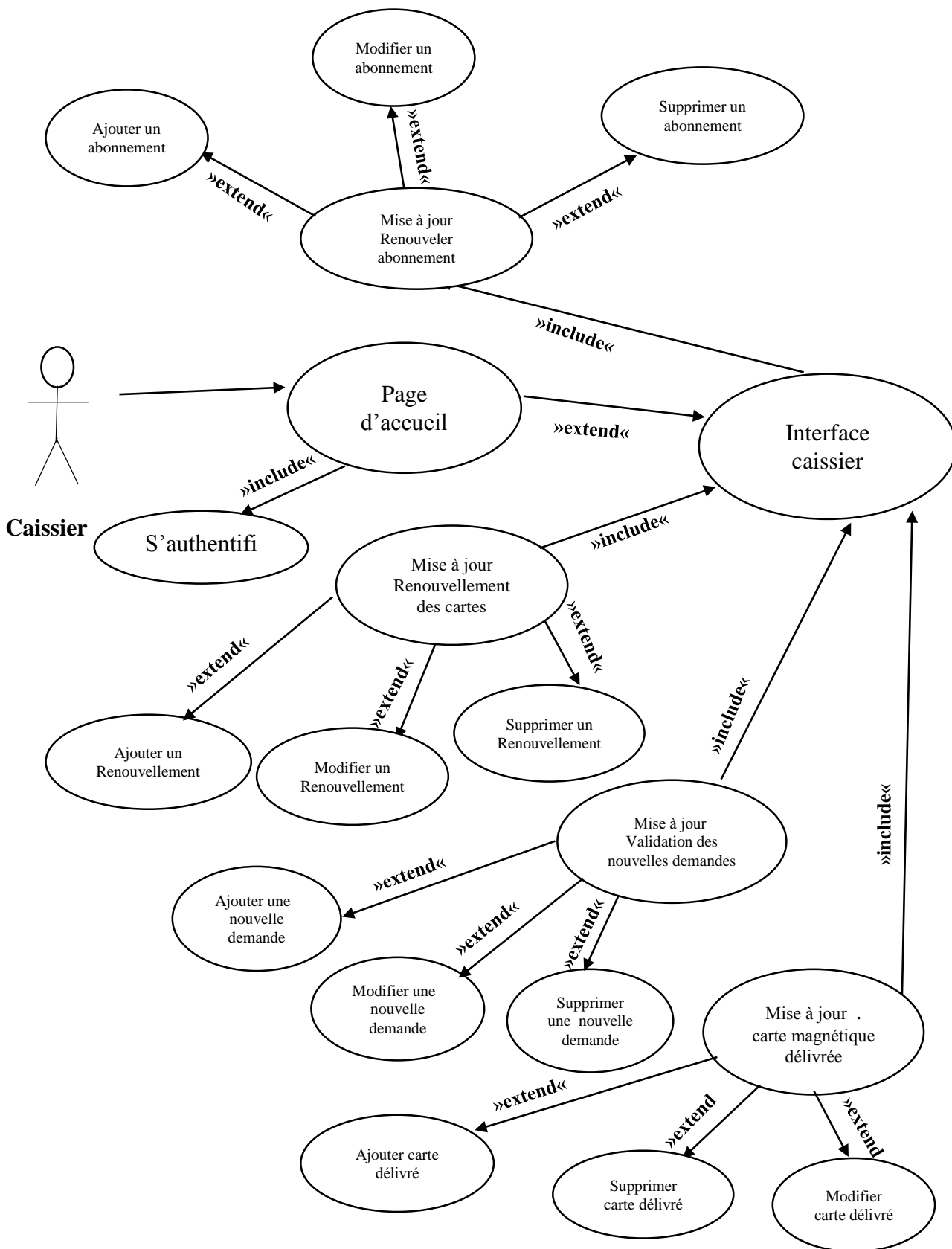


Figure12 : Diagramme de cas d'utilisation pour le caissier

4.1.4 présentation du diagramme de cas d'utilisation pour la gestion des investissements

agricoles :

- Page d'accueil
- S'authentifier
- Interface
- Mise à jour des investissements agricoles
 - Ajouter. Investissements
 - Modifier. Investissements
 - Supprimer investissements

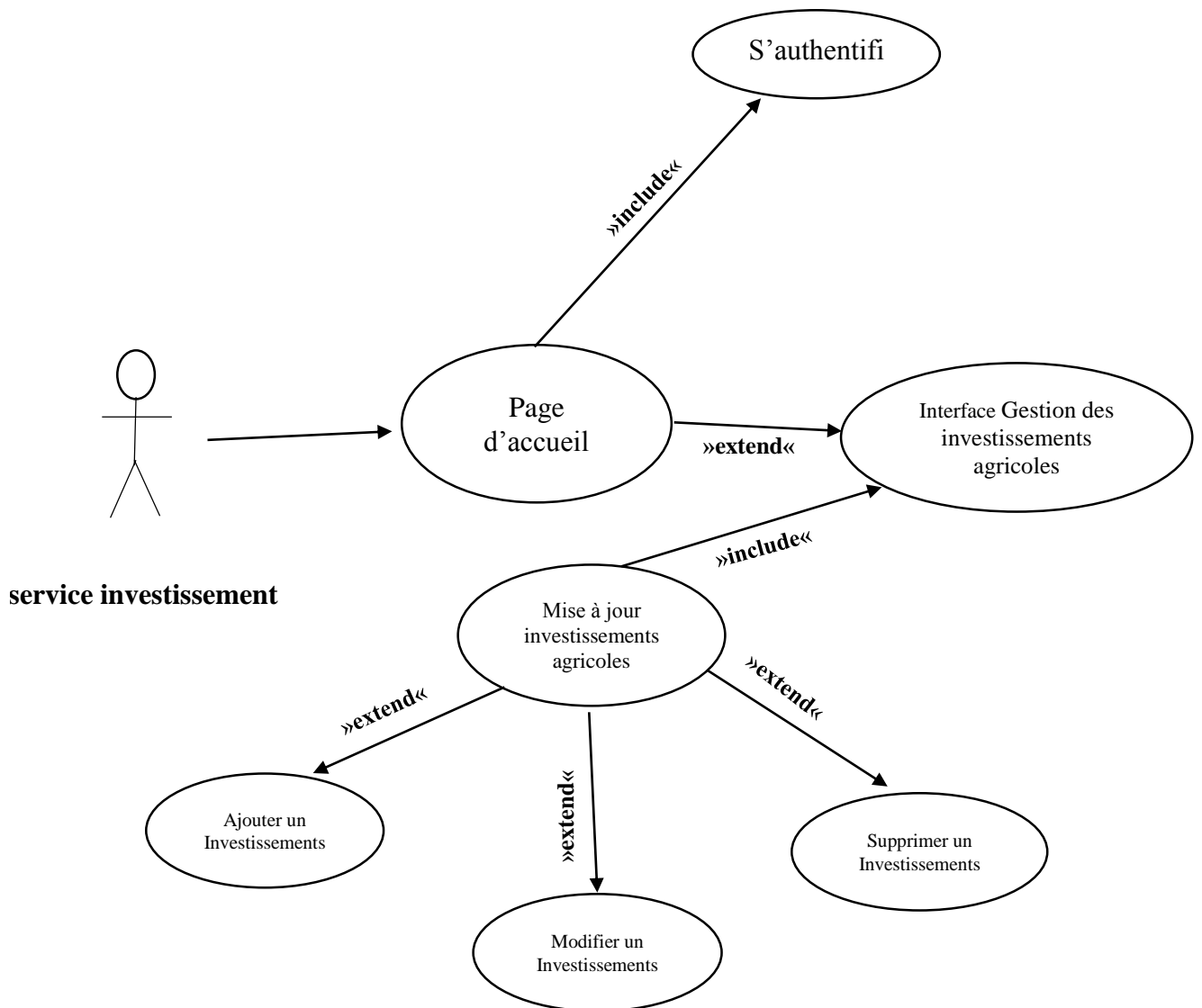


Figure 13 : Diagramme de cas d'utilisation pour le service

4.1.5 présentation du diagramme de cas d'utilisation pour le service d'affiliation et de radiation :

- Page d'accueil
- S'authentifier
- Interface
- Mise à jour des radiations des agriculteurs
 - Ajouter. Radiation
 - Modifier. Radiation
 - Supprimer radiation

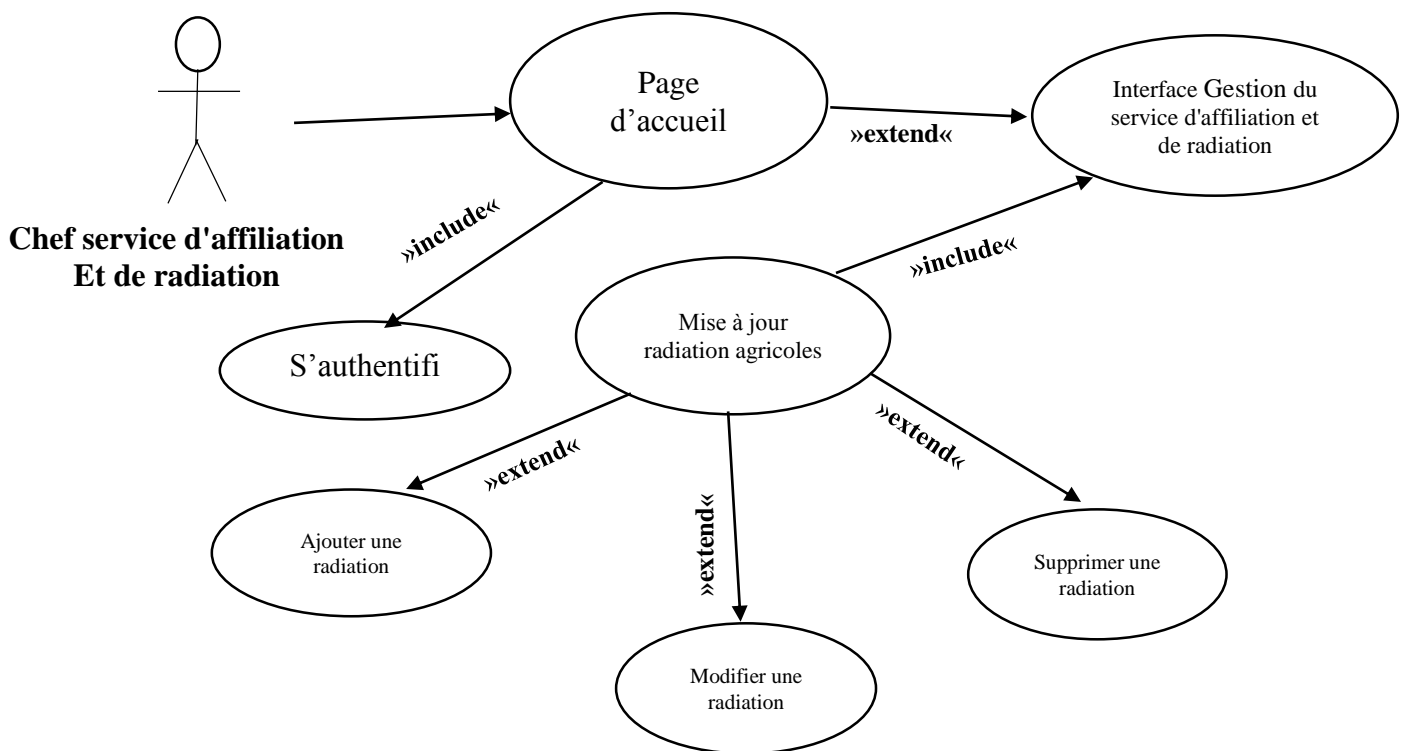


Figure14 : Diagramme de cas d'utilisation pour le service d'affiliation et de radiation

4.1.6 Présentation du diagramme de cas d'utilisation pour l'administrateur :

- Page d'accueil
- S'authentifier
- Interface
- Gérer la base de données

Mise à jour des utilisateurs

- Ajouter utilisateur
- Modifier utilisateur
- Supprimer utilisateur

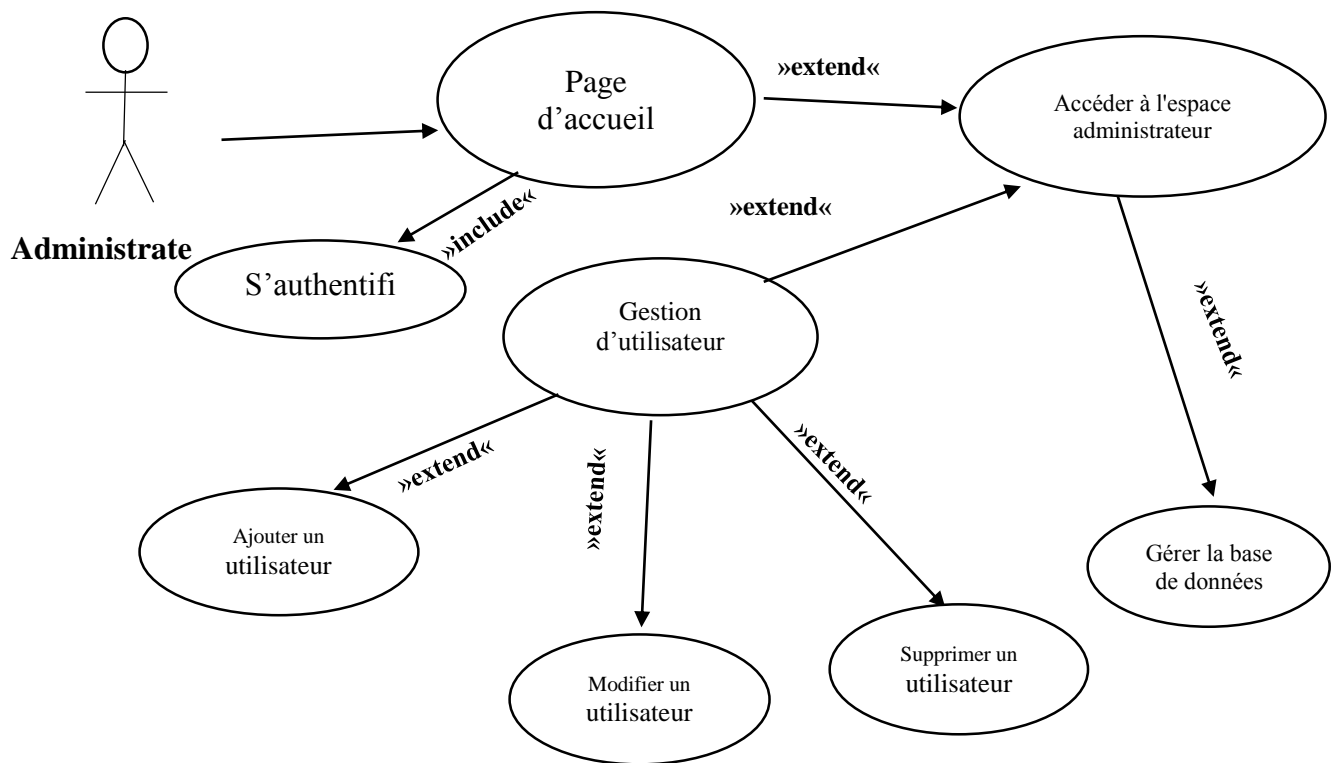


Figure15 : Diagramme de cas d'utilisation pour l'Administrateur

4.2 Le diagramme de séquence :

Le diagramme de séquence est une variante du diagramme de collaboration, permet de visualiser les messages par une lecture de haut en bas. L'axe vertical représente le temps, l'axe horizontal les objets qui collaborent. Une ligne verticale en pointillé est attachée à chaque objet et représente sa durée de vie.

4.2.1 Diagramme de séquence Caissier (Ajouter un abonnement) :

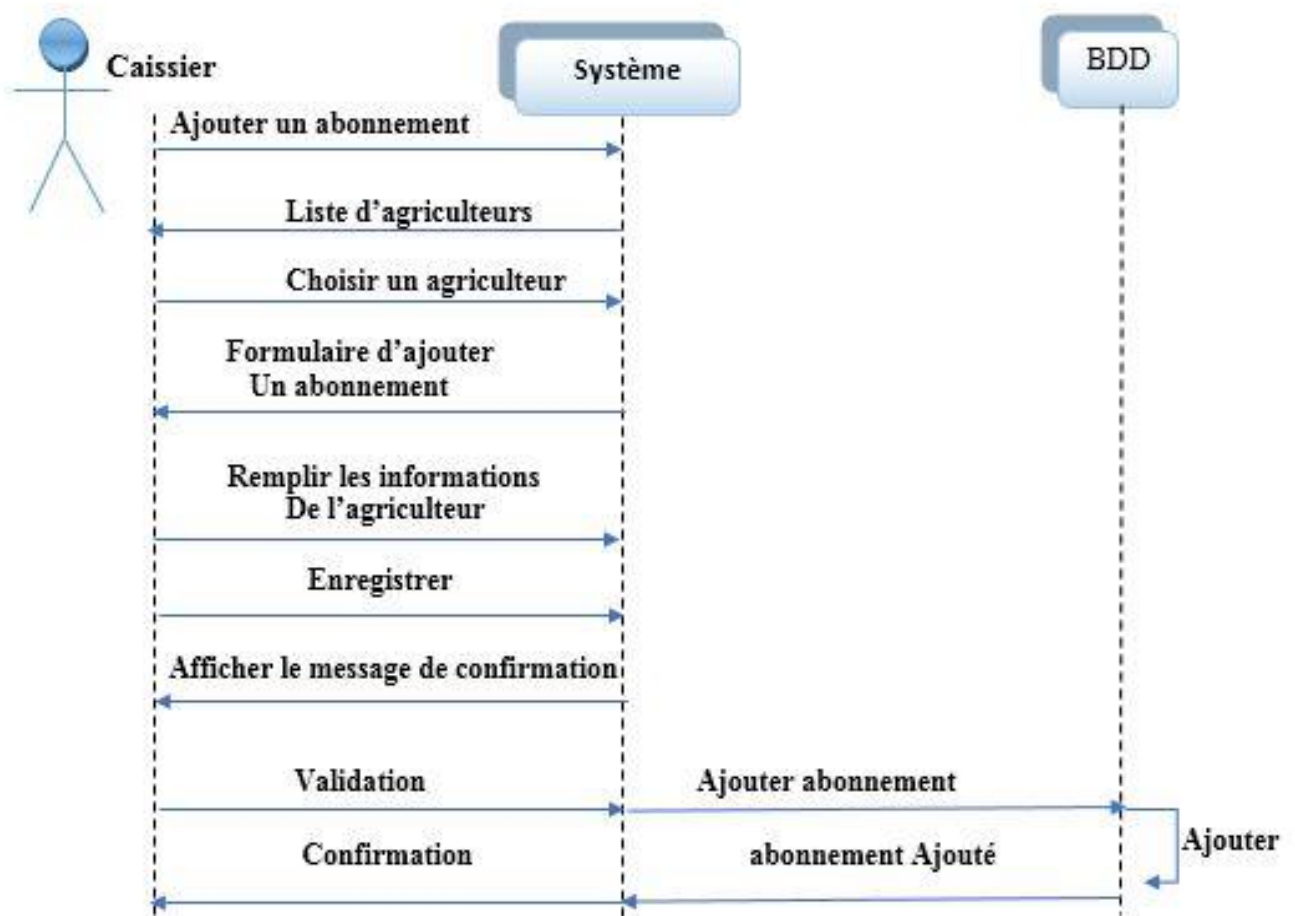


Figure.16 : Diagramme de séquence << Ajouter abonnement >>

4.2.2 Diagramme de séquence Caissier (Modifier un abonnement) :

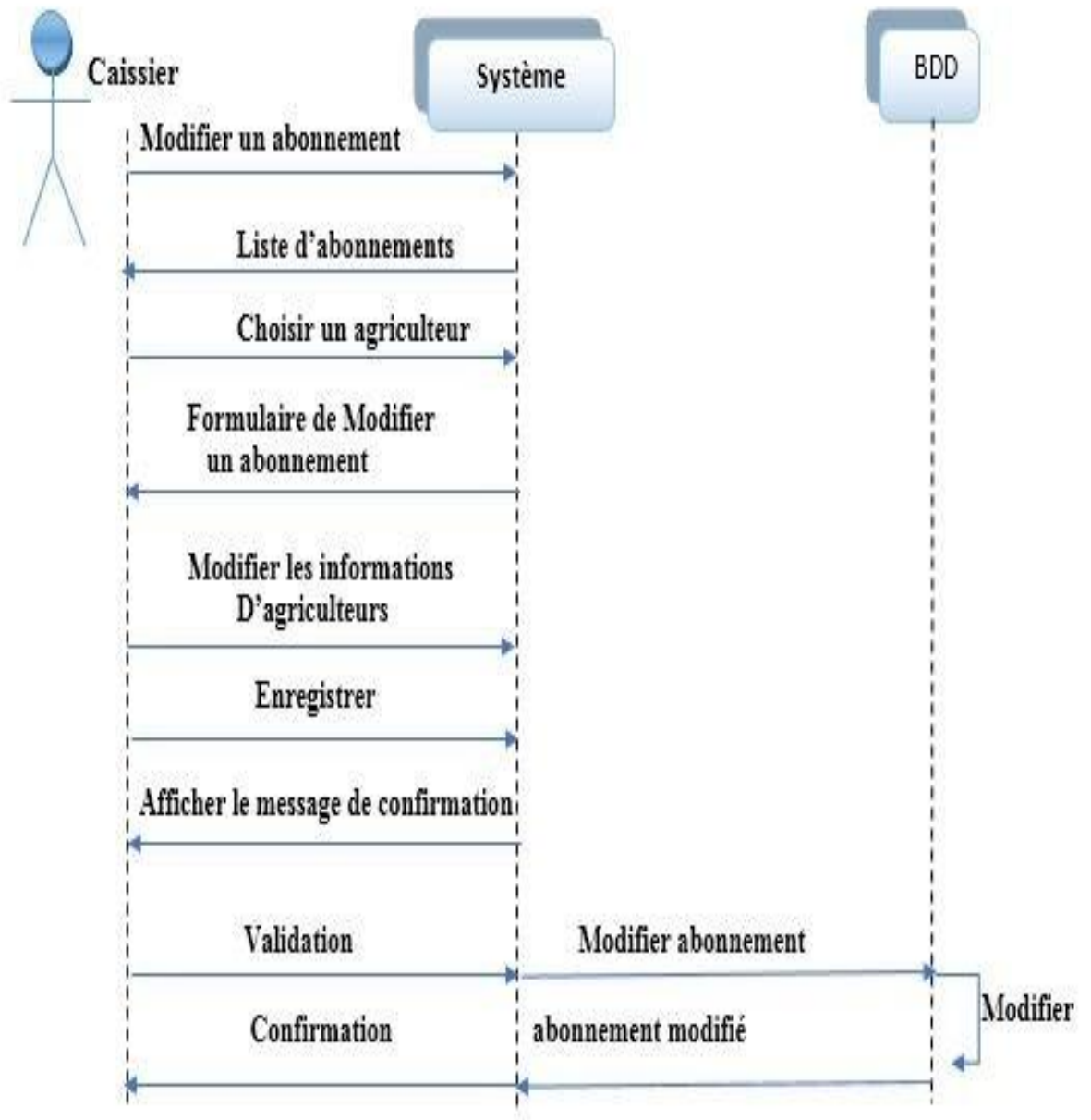


Figure17. : Diagramme de séquence << Modifier abonnement >>

4.2.3 Diagramme de séquence Caissier (Supprimer un abonnement) :

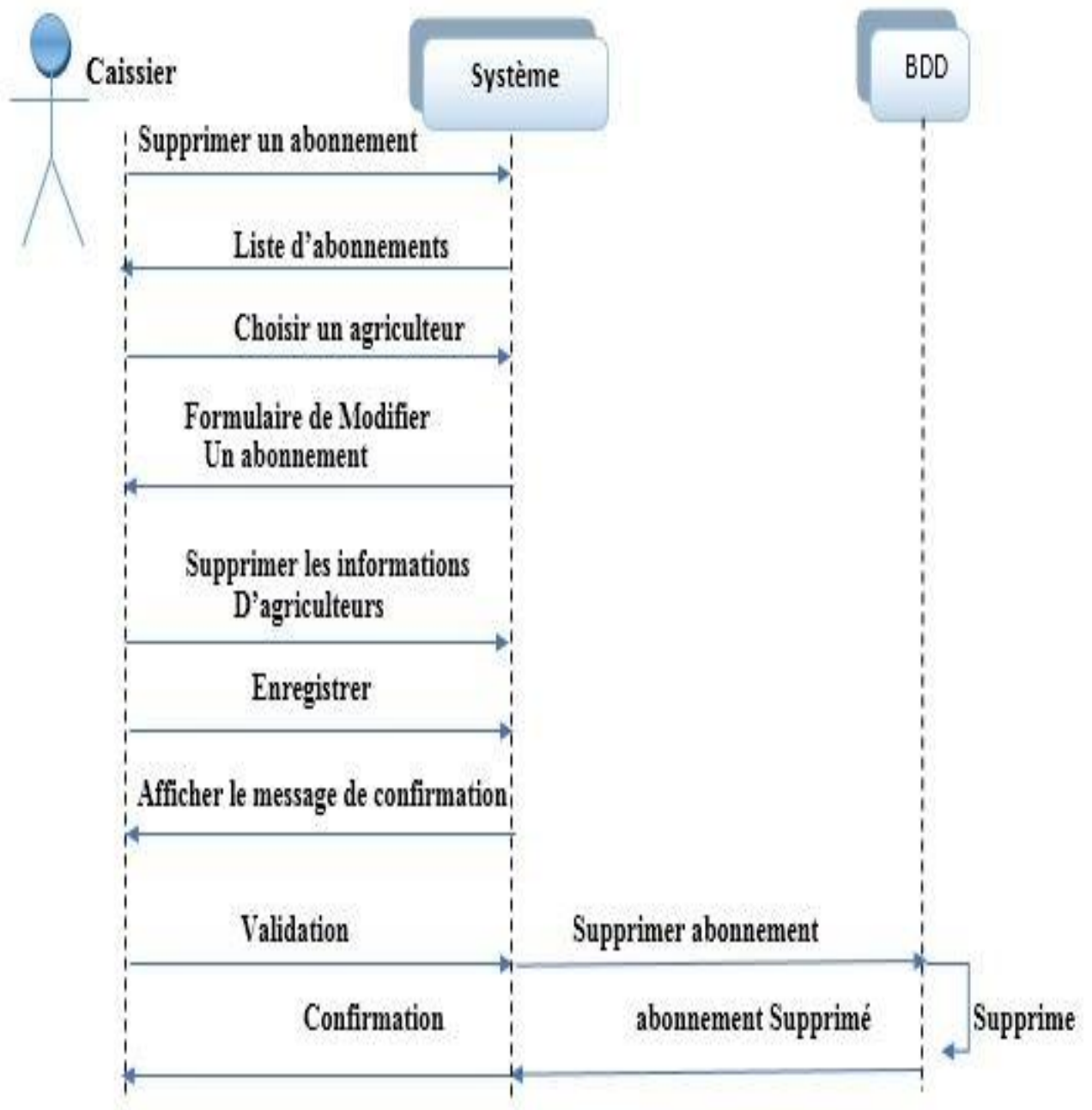


Figure.18 : Diagramme de séquence << Supprimer abonnement >>

4.2.4 Diagramme de séquence Caissier (Renouvellement carte agriculture) :

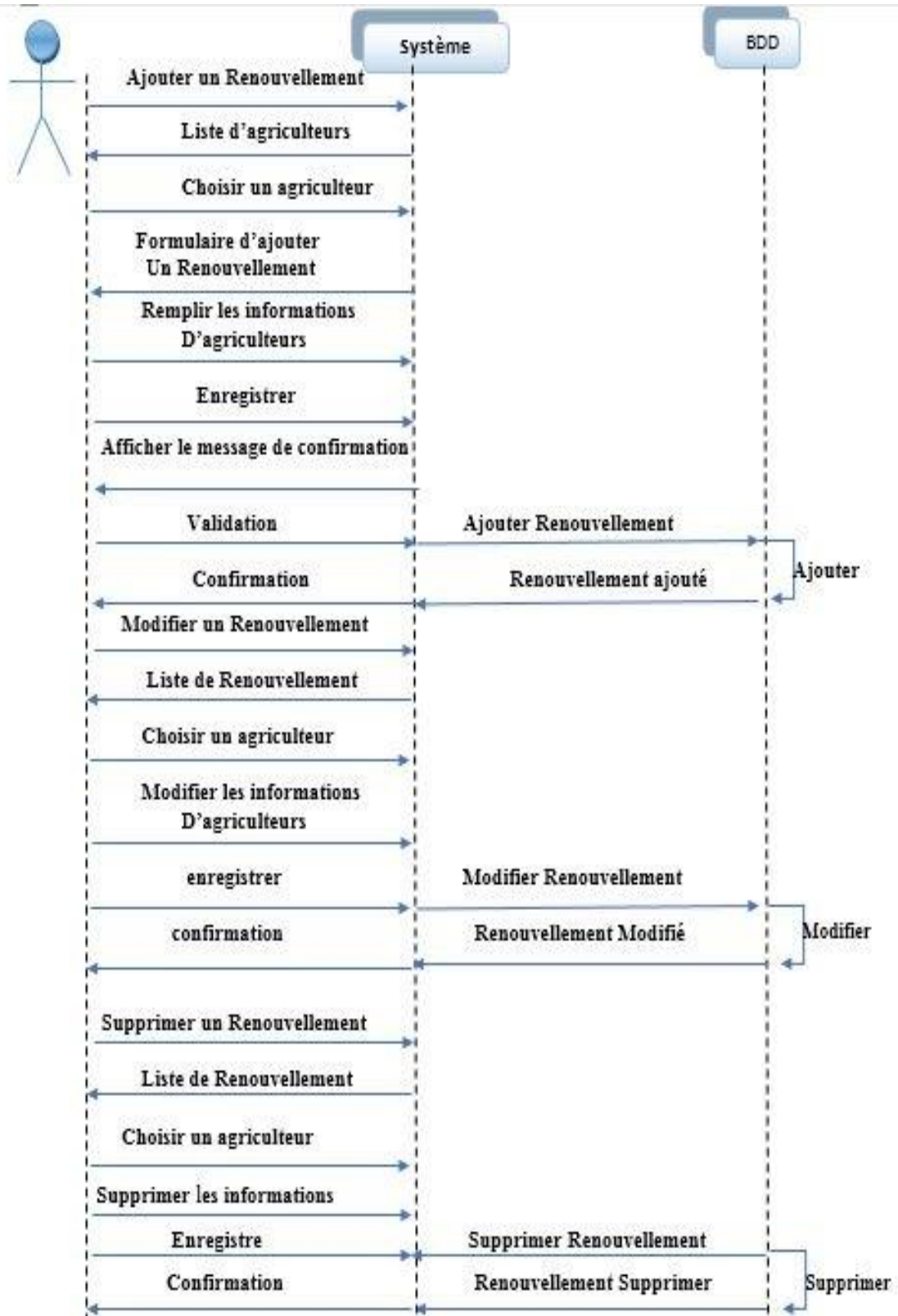


Figure19 : Diagramme de séquence << gestion de Renouvellement carte agriculture >>

4.2.5 Diagramme de séquence <<. Nouvelle demande carte >>

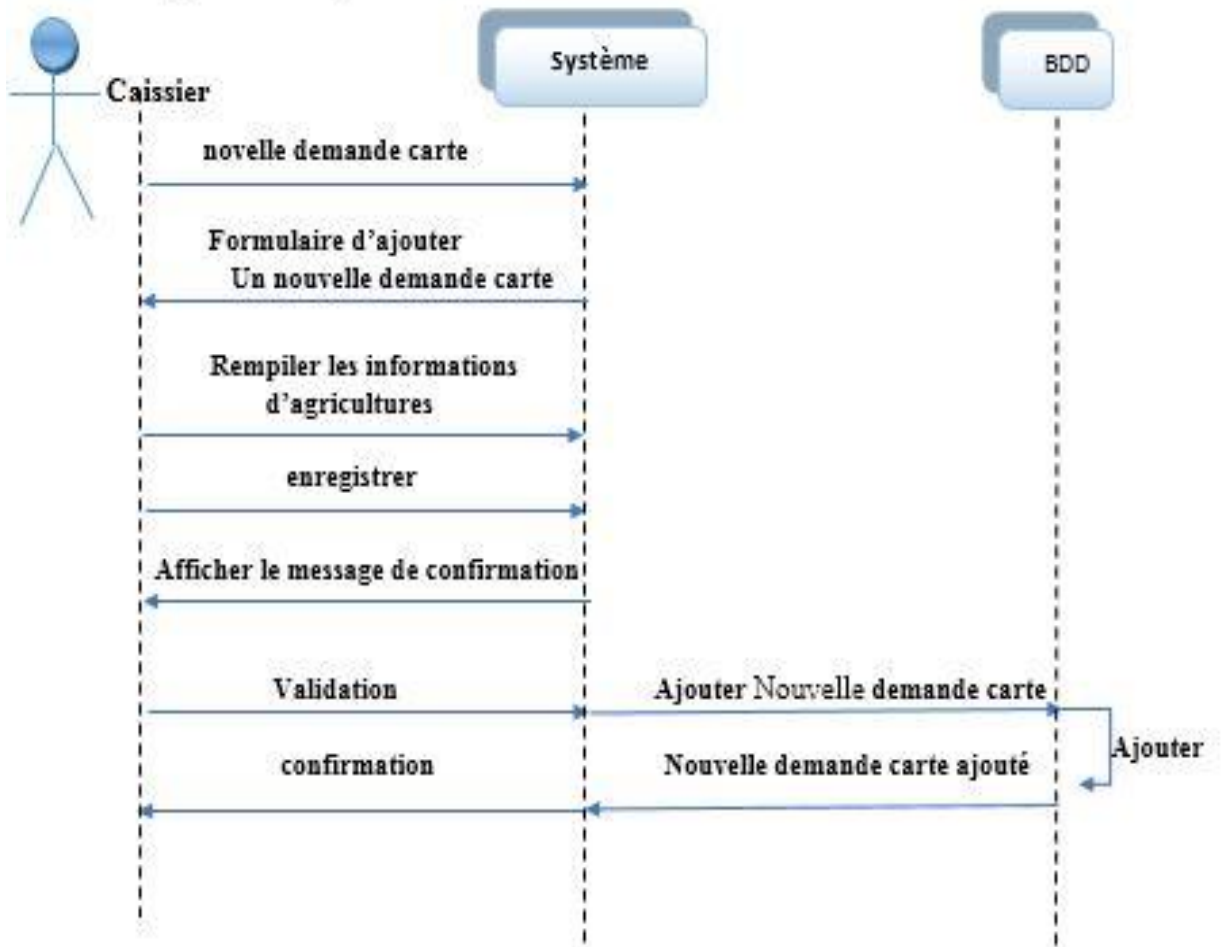


Figure20 : Diagramme de séquence <<. Nouvelle demande carte d'agriculture>>

:

4.2.6 Diagramme de séquence << Gestion des Investissement >>

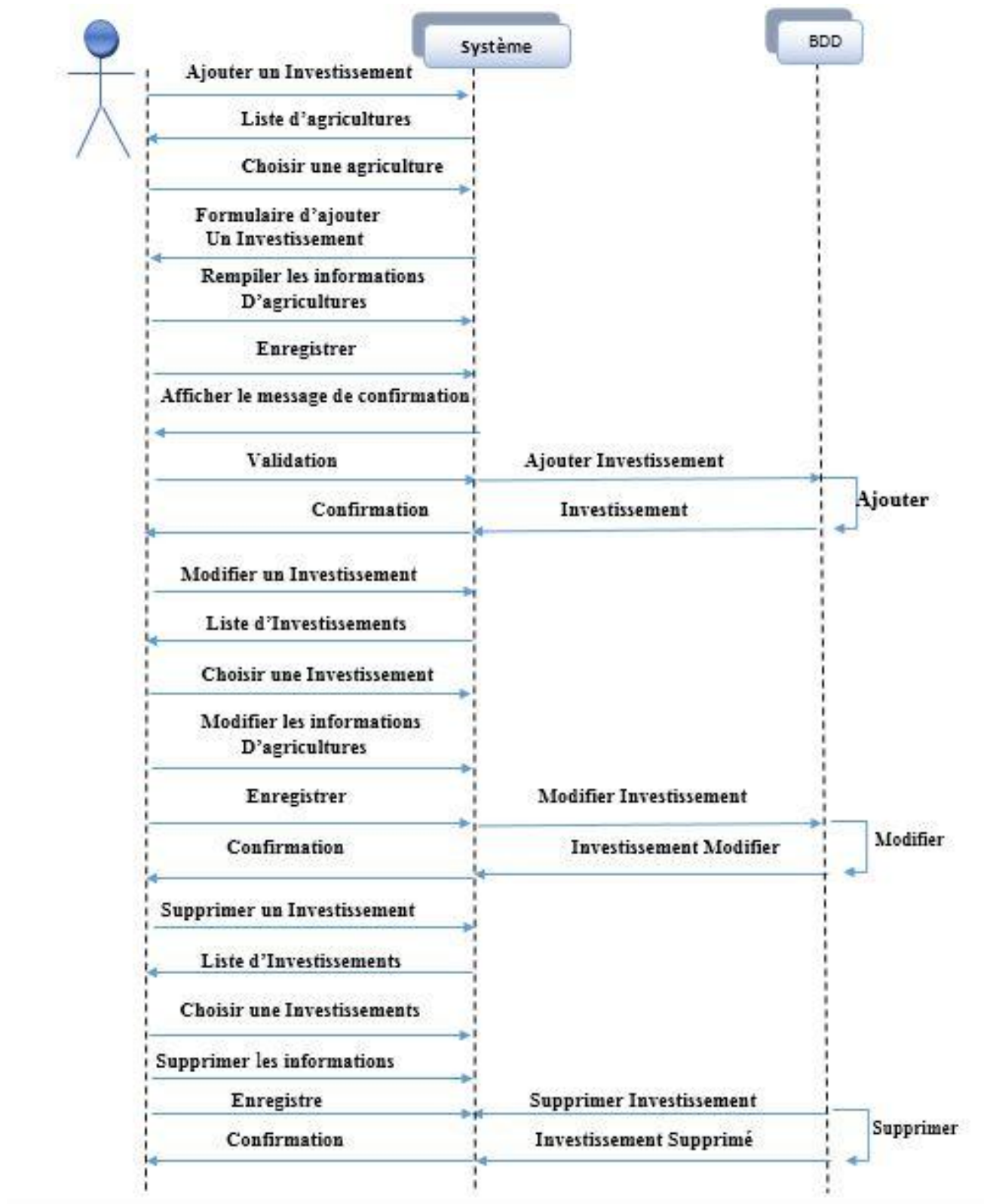


Figure21. : Diagramme de séquence << gestion de Investissement >>

4.2.7 Diagramme de séquence << gestion de radiation >> :

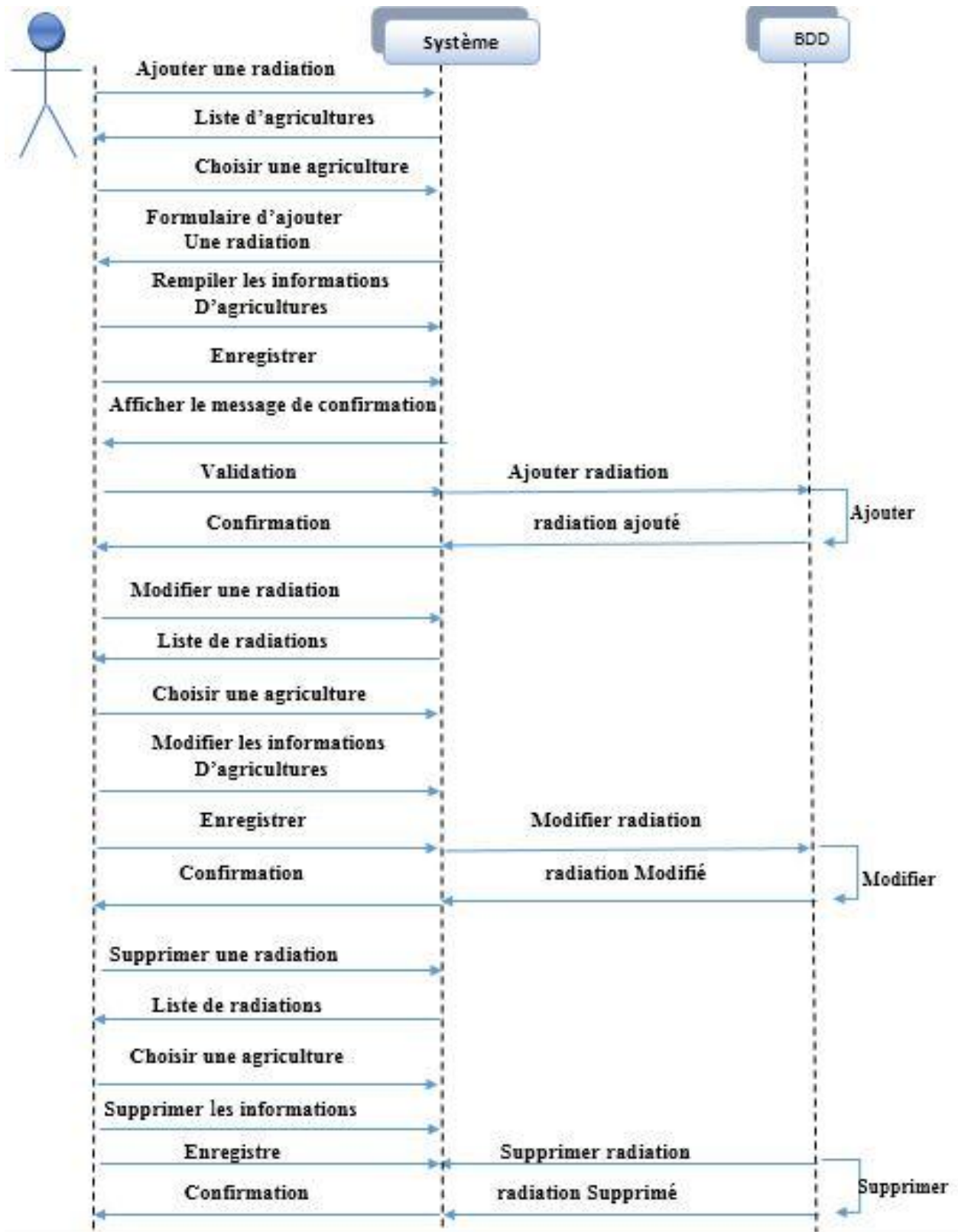


Figure22 : Diagramme de séquence << gestion de radiation >>

4.2.8 Diagramme de séquence << gestion d'utilisateur >>

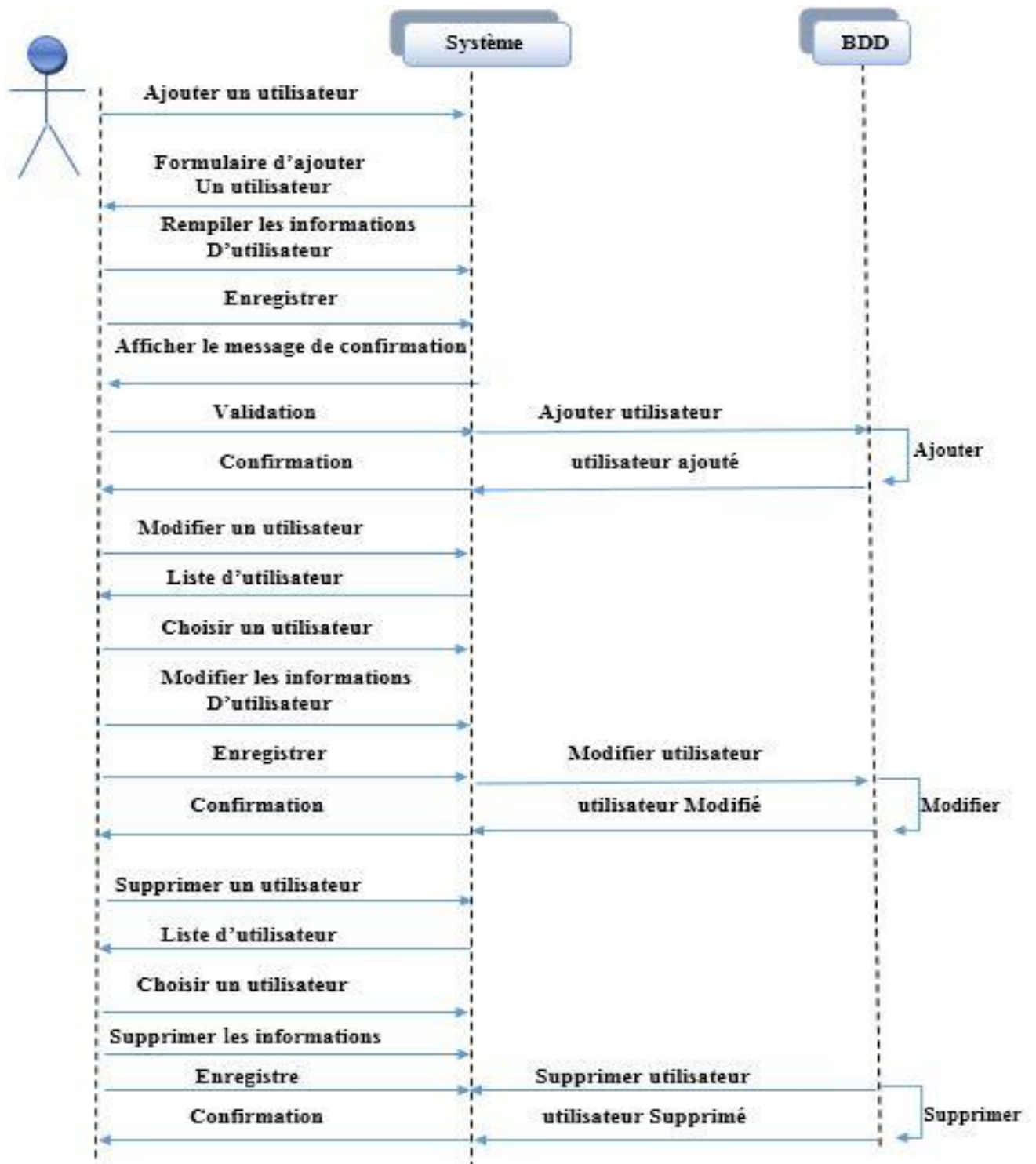


Figure23 : Diagramme de séquence << gestion d'utilisateur >>

4.3 Diagrammes des classes :

Identification des classes d'objets et d'association Cette phase doit déterminer le modèle statique qui consiste en la détermination des classes d'objet, ses attributs et ses méthodes, les associations entre les classes puis la représentation du diagramme de classe qui donne une illustration schématisée des différentes classes et leurs relations

4.3.1 Description détaillée des classes d'objet et d'association :

Classe	Attributs principaux	Méthodes
Table_agriculteur	Num_wilaya Num_nationale Nom_prénom Nom_père Nom_mère Num_télé Num_pv Date_pv Date_naissanc Jugement Lieu_naissance Adresse Type_fonction Activite_fonction Nom_latin Remerq Sex Date_sorti activ_prenc	Les impressions () Ajouter une agriculteur () Modifier une agriculteur () Supprimer une agriculteur ()
Table_investi	Code_inves Num_nationale Zone_ferme Commune_ferme Daira_ferme Date_sorti Pièce_justificative Superficie_globale Superficie_desservie Superficie_cadastrer Superficie_occuper Céréale Aliments Légumes Chambre_plastique Arbre_olive Arbre_fruité Culture_industrielle Bovins Ovins	Les impressions () Ajouter une investissement () Modifier une investissement () Supprimer une investissement ()

	<p>Caprin Équidés Dromadaires Autre_anim Poule_pondeuse Poule_chaire Lapins Dindes Apiculteur Autre_petianim Forage Puits Basin Vallées Barrage Matr_irrigation Autre Matrial_agri Permanen Contractuel Remerq2 Remarque Dont_brebis</p>	
Table_galizi	<p>Num_nationale pay_norml pay_rotar Anneé_cor Date_galizi Zone_ferme Total_galiz Code_galizi remerq</p>	<p>Les impressions () Ajouter une galizi () Modifier une galizi () Supprimer une galizi ()</p>
Table_radiation	<p>Code_radiation Num_radiation_com Date_radiation_com Num_sorti Num_nationale Remarque date_sorti</p>	<p>Les impressions () Ajouter une radiation () Modifier une radiation () Supprimer une radiation ()</p>
Table_renovlle	<p>Code_renovlle Num_nationale Date_renovlle Pay_norml Remerq zone_ferme</p>	<p>Les impressions () Ajouter une renouvellement () Modifier une renouvellement () Supprimer une renouvellement ()</p>

Tableau2 : Description détaillée des classes d'objet et d'association

4.3.2 Diagramme de classe générale :

Parmi les diagrammes le plus importants dans L'UML les diagrammes de classe Le diagramme de classe présente un ensemble de classeurs.il décrit les classes et leurs relations, il peut également décrire les regroupements de classes en paquetage, les interfaces et les objets, les classes qui participent à une collaboration ou qui réalisent une case d'utilisation, etc. [23]

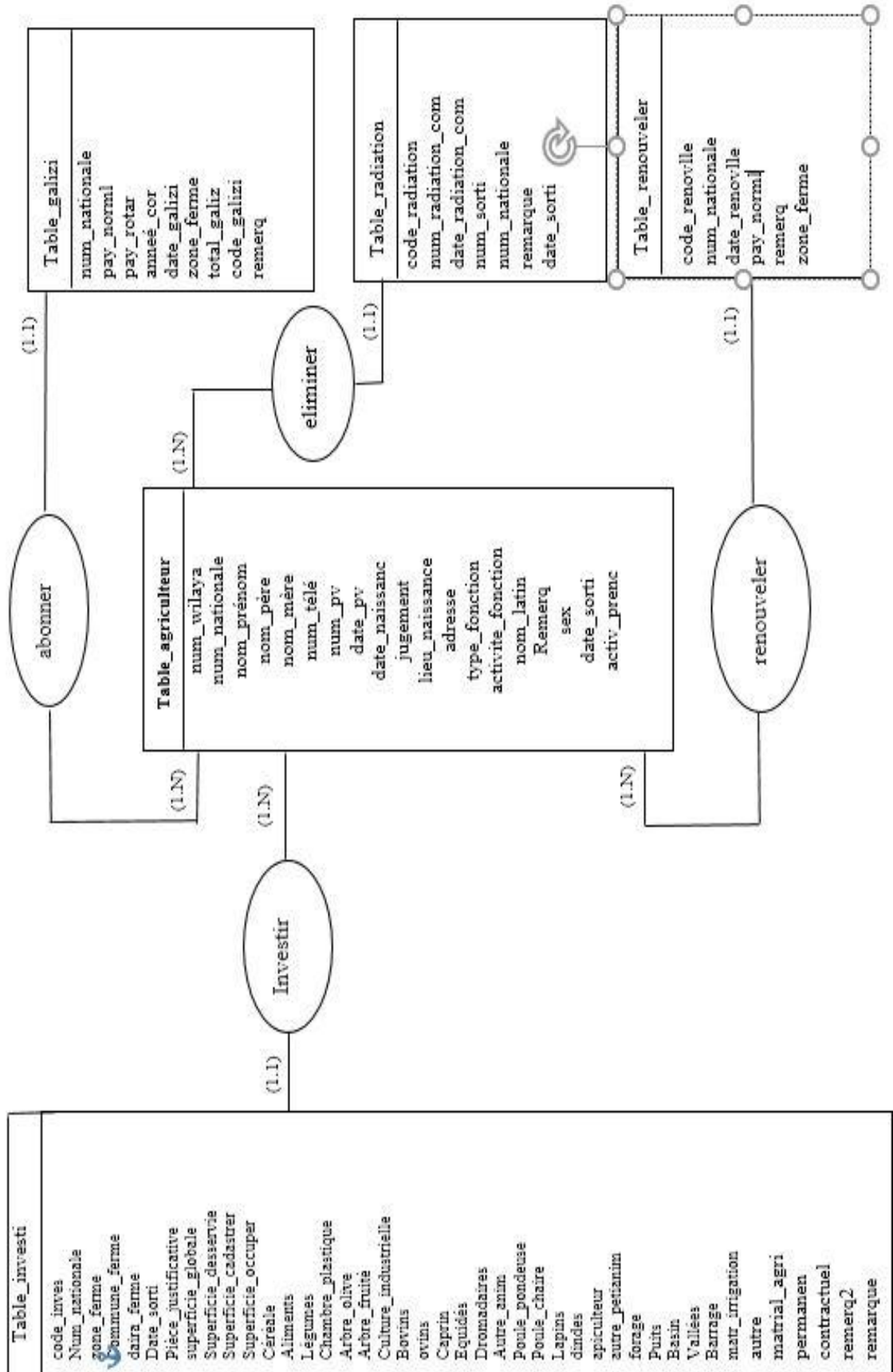


Figure24 : Diagramme de classe générale

5 Implémentation et outils de développement :

Etant la dernière étape dans le cycle de développement des systèmes d'informations, la réalisation consiste à couronner les phases précédentes, donnant un aspect tangible aux suggestions présentées lors de l'étude de l'existant et est forme concrète à la conception. Afin de présenter le prototype réalisé, en utilisant les prises d'écrans (screen prints) pour figurer le travail fait et d'illustrer la grande et principale fonctionnalité

5.1 Environnement logiciel :

Le long de la phase de développement, nous avons utilisé l'environnement logiciel suivant :

Système d'exploitation : Microsoft Windows 7

Outil de développement : Delphi xe2

Data base server : sql server 2008 R2 pour la gestion de la base de données

5.2 Delphi XE2 :

De nombreux utilisateurs de Delphi se demandent s'ils vont trouver des raisons impérieuses de migrer vers Delphi XE2. Ici, ils sont : une pléthore de nouvelles fonctions connexes à la productivité des développeurs sans précédent, visant tous à votre capacité à créer des applications de qualité supérieure avec des performances améliorées. Cet article donne quelques bonnes raisons de migrer, avec un aperçu de toutes les nouvelles fonctionnalités ajoutées à l'article de presse Delphi depuis la version 7, où Delphi est un IDE pour construire des applications Windows 32 bits et est devenu un IDE pour Windows 64 bits, Mac et le développement IOS [25].

Le tableau suivant donne un aperçu rapide des nouveaux hauts niveaux, les capacités de chacune des versions de produits Delphi depuis Delphi 7. Ce document fournit les informations techniques approfondies pour toutes ces fonctionnalités et bien plus encore.

5.2.1 Propriétés de Delphi XE2

- Plate-forme FireMonkey
- le développement 64 bits, Mac et IOS de Windows
- Styles VCL
- appui RTL pour Windows et Mac
- DataSnap surveillance et de contrôle
- Connecteurs DataSnap
- les pilotes dB Express pour Mac, Windows 64 bits et nouveau pilote dB Xpress ODBC

- Couverture support de l'API pour Amazon
- assistant du gestionnaire de déploiement et la plate-forme

5.3 Microsoft SQL Server (alias MSSQL) :

est un système de gestion de base de données développé par la société Microsoft. Il permet donc par son interface de créer et lister ses tables, en dessiner les diagrammes, y exécuter du code SQL appelé Transact-SQL, en visualisant son plan d'exécution, et de sauvegarder et lancer des procédures stockées et triggers.

Ceci en fait une alternative compétitive avec MySQL et Oracle.

5.4 Microsoft Access 2003 :

Microsoft Access, en abrégé MS Access ou Access tout court, est une SGBDR (Système de gestion de Bases de données relationnelles), cela signifie que MS Access peut mémoriser les informations et les relations (les liens) existant entre ces informations. Access permet de stocker les données dans les tables puis de définir les relations existant entre ces tables. De plus, vous pouvez créer des requêtes (des critères de recherche) qui permettent de sélectionner une partie d'informations stockées dans des tables de la base de données en question, est qui dépendent aux critères que vous spécifiez dans votre requête. Access permet également de créer des formulaires et des états pour afficher modifier, ou analyser les informations provenant directement ou celles qui répondent à des critères donnés si vous spécifiez la requête correspondant. En conclusion une base de données MS Access est une ensemble d'informations relatives à un sujet par exemple la gestion de personnel. [24]

6 Présentation des interfaces :

6.1 Interface de login :

Cette interface donne la main aux utilisateurs pour accéder aux différentes interfaces après vérification le nom d'utilisateur et mot de passe.



Figure25 : Interface d'authentification

6.2 Fenêtre principal après connexion :

- Cette interface nous permet accéder aux les différentes interfaces de l'application nous permet l'accès aux différentes options tels que (gestion d'utilisateur, radiation renouvellement.et abonnement carte d'agriculture, gestion investissement) L'interface Fenêtre principal après connexion



Figure26 :L'interface Fenêtre principal après

6.3 L'interface agriculteur :

L'interface agriculteur comporte quatre boutons qui sont (nouveau, supprimer, modifier et valider).
 Qu'afficher nous toutes les informations sur les agriculteurs de toutes les applications

سجل الفلاح

الرقم الولائي 1 رقم المحضر 1 النشاط الرئيسي رقم الهاتف 661526990 تاريخ تحديد الهوية 2021-05-20

الرقم الوطني 6703 تاريخ المحضر 1998-04-14 نوع النشاط 03 الجنس تاريخ المصادقة 21/05/2021

الرمز البلدي 9 طبيعة النشاط 01C تاريخ تجديد البطاقة 21/05/2021 تاريخ الشطب

الاسم واللقب اسم الأب اسم ولقب الأم

مكان الازدياد حكم مكان الازدياد العنوان

الاسم بالفرنسية

ملاحظة

إضافة

تعديل

حفظ

إلغاء

Figure27 : L'interface agriculteur

6.4 L'interface pour la gestion d'affiliation et de radiation :



Figure28 : gestion d'affiliation et de radiation

6.5 L'interface de caisse :



figure29 : L'interface de caisse

Conclusion Générale

Dans ce projet, nous avons proposé un travail d'intégration entre les différentes applications actives au niveau de la chambre agricole de la wilaya de M'sila. L'intégration assure plus de cohérence avec une omniprésence de l'information au niveau de toutes les applications utilisées.

Dans la phase conceptuelle de ce travail, le langage UML a été exploité avec différents diagramme. En plus, l'environnement DELPHI XE avec MS ACCESS et SQL server ont fournis les fondements pratiques pour mettre en œuvre ce projet.

Le système d'information que nous avons élaboré, atteint nos objectifs tracés au début de ce travail et nous espérons que ses objectifs en permettant à l'utilisateur d'évaluer avec plus de souplesse l'exécution de ses taches avec plus de fluidité dans la circulation de l'information et moins de contraintes dans le traitement automatique. Certes que ce système reste perfectible mais il pourrait constituer une base de réflexion pour la promotion de solution informatique.

Pour notre part, la confrontation de nos connaissances théoriques avec la réalité du terrain nous a permis de renforcer notre conviction quant à la pertinence du choix informatique.

Enfin, comme futures suggestions nous proposons à enrichir notre base de données par la prise en charge de d'autres activités telles que numérisation du secteur agricole. En plus nous proposons aussi d'améliorer cette application par l'accélération et la multiplication des accès par plusieurs postes utilisateurs.

Bibliographie

Un site web :

[1] <http://www.filaha.net> consulté le : 28/04/2021

[2] <https://www.ediservices.com> le : 29/04/2021

[3] <https://fr.wikipedia.org> le : 29/04/2021

[4] <https://anyconnector.com> le : 02/05/2021

[5] <https://www.axysweb.com> le : 05/05/2021

Un ouvrage :

[6] Saïd Izza, Intégration des systèmes d'information industriels : une approche flexible basée sur les services sémantiques, thèse de doctorat de l'école supérieure nationale des Mines de Saint-Étienne, novembre 2006.

[7] S. Izza, L. Vincent, P. Burlat, H. Solignac et P. Lebrun, Intégration d'applications : Etat de l'art et perspectives, JIE 2'04, les 2 emes Journées d'Informatique pour l'Entreprise, université Saad Dahleb, Blida (Algérie) 2004

[8] R. Sharma, B. Stearns, T. Ng, J2EE Connector Architecture and Enterprise Application Integration, Publisher: Addison Wesley, First Edition December 01, 2001. ISBN: 0-201-77580-8.

[9] Fabien Cleuet, Vincent Goujon, Henry Peyret, EAI et urbanisation : comment réussir ? l'Association Française d'Adit et de conseil Informatique (AFAI), Mai 2001

[10] Yuehui Chen, Ju Yang, Yong Zhang and Jiwen Dong, Evolving Additive Tree Models for System Identification, International Journal Of Computational Cognition, VOL. 3, NO. 2, JUNE 2005.

[11] David S. Linthicum, Enterprise Application Integration, Ed: Addison Wesley, 2001 ISBN: 0-201-61583-5.

[12] S. Janarthanan, A. Ekambaram, M.M. Kusum, N.K. Nair, Integrating heterogeneous systems using Enterprise Application Integration, research & analysis report, pp (1- 20).

[13] Manoj Seth, Web services- A Fit for EAI, p1, p2.
http://www.developer.com/tech/article.php/10923_1489501.

[14] Abraham Kang, Enterprise application integration using J2EE, 2002.

[15] Jihed Touzi, Aide à la conception de système d'information collaboratif support de l'interopérabilité des entreprises, thèse de doctorat de l'école des Mines d'Albi Carmaux, Novembre 2007.

[16] Philippe Courty, L'évolution des besoins et des solutions d'intégrations, CJP 2005.

Bibliographie

[17] EBM Websourcing : Nouvelles technologies pour l'intégration : les ESB, white paper, janvier 2006

[18] Tanguy Crusson, Business Process Management, de modélisation à l'exécution : Positionnement par apport aux Architectures Orientées Services, INTALIO, 2003.

[19] David A. Chappell, Enterprise Service Bus, Publisher: O'Reilly Media, First Edition June 2004. ISBN: 0-596-00675-6.

[20] Kiran Kanetkar, A Roadmap to building an ESB, Integration for Enterprise, Boston, May 2006.

[21] P-A.Muller, Modélisation objet avec UML, Eyrolles, Paris, 1997.

[22] O.Sigaud, Introduction à la Modélisation orientée objet avec UML , Paris , 2006.

[23] Uml2.pratique de la modélisation 2eme édition Benoit Charroux,Aomar Osmani, Yann Htierry_mieg

[24] Access XP par le pratique (Chahide Khichane).

[25] Delphi 7.pdf

Un mémoire ou une thèse :

[26] B.Ilham,H.M.Siham, conception et realisation d'un system de gestion de stockd'une unite de production(Les Moulins du Hodna M'sila) Ingenieur Université de M'sila , Algerie ,201

Résumé

ملخص

ان تطور تكنولوجيا المعلومات سهل تنظيم العمل. يضمن تكامل تطبيقات المؤسسة (EAI) اتساقا أكبر مع المعلومات المنتشرة عبر جميع التطبيقات المستخدمة.

الهدف من مشروعنا هو تطوير نظام معلومات لتحقيق التكامل بين التطبيقات النشطة المختلفة على مستوى الغرفة الفلاحية لولاية المسيلة عن طريق (EAI) للقيام بذلك ، استخدمنا لغة النمذجة UML 2.0 ، Delphi xe2 ، DBMS ACCESS ، sql server. ،

كلمات مفتاحية : EAI . SI . API . UML . الغرفة الفلاحية

Abstract

Information technology development facilitates the organization of work. Enterprise Application integration (EAI) ensures greater consistency with pervasive information across all applications used. The Objective of our project is to develop an information system to realize the integration between the different active applications at the level of the agricultural chamber of the wilaya of M'sila by the EAI. To do this, we used the modeling language UML 2.0, Delphi xe2, DBMS ACCESS, sql server.

Keywords : EAI, SI, API, UML, the agricultural chamber

Résumé

Le développement technologique de l'information facilite l'organisation du travail. L'intégration des applications d'entreprise (EAI) assure plus de cohérence avec une omniprésence de l'information au niveau de toutes les applications utilisées.

Le but de notre projet est de développer un système d'information pour réaliser l'intégration entre les différentes applications actives au niveau de la chambre agricole de la wilaya de M'sila par l'EAI. Pour ce faire, nous avons utilisé le langage de modélisation UML 2.0, Delphi xe2, SGBD ACCES, sql server.

Mots clés : EAI, SI, API, UML, chambre agricole