

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



N° Ref :...../13...

**Centre Universitaire de Mila**

**Institut des sciences et de la technologie**

**Département de Mathématiques et Informatique**

**CONCEPTION ET DÉVELOPPEMENT  
D'UNE APPLICATION DE GESTION DES  
PROJETS DE RECHERCHE AU NIVEAU DU  
CENTRE UNIVERSITAIRE DE MILA**

**Mémoire préparé En vue de l'obtention du diplôme de licence en  
Informatique**

**Préparé par :**

1-Azizi ilyes  
2-Amri fateh  
3-Boumaali abd araouf

**Encadré par :**

-Khalfi Souheila

**Filière : Informatique**

**Année universitaire : 2012/2013**

# **DEDICACES**

*A mes très chers parents qui m'ont beaucoup soutenu,  
encouragé et qui ont fait de moi ce que je suis aujourd'hui ;*

*A mes oncles et tantes, ainsi que mes cousins et cousin ;*

*A tous mes amis en particulier*

*Hmza, aissa, sadam, naamane, youcef ;*

*A tous mes amis (es) 3<sup>em</sup> année informatique en particulier*

*Fateh ; rami ; mourad ; ilyase ;*

*N'oublier pas les meilleurs amis durant 3 dernières années*

*, nasre eddine, Sami, Abd alhake.*

*A toute les familles*

*Boumaali et lamtayache;*

*A tous ceux que j'aime tant et que je n'ai pas cités ;*

*Je dédie ce mémoire ...*

***Boumaali abdaraouf***

# Dédicace

Je dédie ce modeste travail

À ceux qui m'ont donné la force, le courage et qui se sont tellement sacrifiés pour moi et qui m'ont fourni tous les conseils durant toutes les années de ma formation :

**Ma mère Khadija** l'exemple même de l'amour de l'affection de la bienveillance, que dieu vous garde.

**Mon père Noreddine** l'homme le plus merveilleux du monde,

A mon frère Zohir,

A ma sœur Meriem, son mari Zinedine et son fils abd albari

A toute ma famille sans exceptions.

A mes binômes Abd araouf , ilyes.

A tous mes amis de la ville : Bourhane, Yasine, Oussama.

Sans oublier les meilleurs amis durant les trois dernières années : Khere Eddine, Rami, Yassere, Nasre Eddine, Sami Et Abd Alhake.

*AMRI FATEH*

# Dédicace

Je dédie ce mémoire

A ma mère qui ma donné la vie et sa bienveillance suffisante pour que j'étudie bien et j'obtiens la qualification dans ma vie

A mon père qui a souffert plusieurs années pour me permettre de vivre, d'étudier pour devenir un homme respectable dans le future.

A mon frère qui étudie en 2ème année lycée

A ma sœur qui va passé le bac à qui je souhaite un bon courage

A mes amis surtout Fateh et Abderaouf qu'ils travaillent beaucoup avec moi pour accomplir ce mémoire à la fin d'étude universitaire.

*Azizi ilyes*

## **Remerciement**

*Nous tenons à remercier toujours et par cette occasion, en premier et avant tout, notre créateur <<ALLAH>> car sans sa majesté et son aide nous n'aurions pu arriver à ce stade scientifique.*

*Nous présentons nos sincères gratitudes et remerciements à notre encadreur M<sup>elle</sup> KHALFI SOUHEILA pour la grand soutien morale qu'elle nous a apportée au cours de notre projet et aussi pour son aide précieuse et ses conseils judicieux qu'elle nous a fait bénéficiers*

*Nous tenons à exprimer nos sincères remerciements à tous les personnels de l'institut de l'informatique Ainsi que tous les enseignants qui nous ont enseigné durant toutes nos années D'étude.*

*Nous devons remercier notre dirigeant de stage qui nous a donné toutes les informations nécessaires pour notre projet Mr YAHIA ABDELWAHABE*

*Enfin, Nous exprimons nos plus vifs remerciements à toute personne qui nous a aidée à élaborer ce travail de loin ou de près.*

## Introduction générale

1. Contexte de travail.....	1
2. Problématique et motivation.....	1
3. Objectif de travail.....	1
4. Organisation du mémoire .....	2

## CHAPITRE 1 LE LANGAGE UML

1.1 Introduction.....	4
1.2 Le langage de modélisation unifié (UML).....	4
1.2.1 Définition.....	4
1.2.2 Historique.....	4
1.2.3 Les avantages d'UML .....	5
1.2.4 Les Diagrammes d'UML.....	6
1.2.5 Description de quelques diagrammes.....	7
1.2.5.1 Le diagramme de cas d'utilisation.....	7
1.2.5.2 Le diagramme de classe .....	7
1.2.5.3 Le diagramme de séquences.....	8
1.2.5.4 Diagramme d'états-transitions.....	9
1.2.5.5 Diagramme d'activités.....	10
1.3 Mise en œuvre d'UML.....	11
1.3.1 Identification des besoins et spécification des fonctionnalités.....	11
1.3.1.1 Identification et représentation des besoins diagramme de cas d'utilisation.....	11
1.3.1.2 Spécification détaillée des besoins Diagrammes de séquence système..	12
1.3.1.3 Maquette de l'IHM de l'application (non couvert par UML).....	12

1.3.2 Phases d'analyse.....	13
1.3.2.1 Analyse du domaine : modèle du domaine.....	13
1.3.2.2 Diagramme de classes participantes.....	14
1.3.2.3 Diagrammes d'activités de navigation.....	15
1.3.3 Phase de conception .....	17
1.3.3.1 Diagrammes d'interaction.....	17
1.3.3.2 Diagramme de classes de conception.....	19
1.4 Conclusion .....	20

## **CHAPITRE 2 ETUDE DE L'EXISTAN**

2.1 Introduction.....	22
2.2 Présentation de l'organisme d'accueil (CENTRE UNIVERSITAIRE DE MILA).....	22
2.3 Cadre d'étude.....	24
2.4 Description des tâches effectuées au niveau du service.....	25
2.4.1 L'étude des nouveaux projets.....	25
2.4.2 L'évaluation des projets en cours de réalisation.....	25
2.4.3 Gestion des demandes de prolongation.....	25
2.5 Étude de document.....	25
2.6 Conclusion.....	30

## **CHAPITRE 3 ETUDE DE CAS**

3.1 Introduction.....	32
3.2 Identification des cas d'utilisations.....	32
3.3 Description des cas d'utilisation.....	32
3.3.1 Cas d'utilisation «authentification».....	32
3.3.2 Cas d'utilisation « ajouté projet».....	33
3.3.3 Cas d'utilisation «ajouté un chercheur».....	33
3.3.4 Cas d'utilisation «Mise à jour des projets».....	34
3.3.4.1 Modifier projet.....	34
3.3.4.2 Supprimer un projet.....	34
3.3.5 Cas d'utilisation «Mise à jour des chercheurs».....	35
3.3.5.1 Modifier chercheur.....	35
3.3.5.2 Supprimer un chercheur.....	35
3.3.6 Cas d'utilisation « la prolongation ».....	36
3.3.7 Cas d'utilisation« l'état avancé ».....	36
3.3.8 Cas d'utilisation «évaluation de l'état d'avancement »:.....	37
3.4 Diagramme de cas d'utilisation.....	38
3.5 Les diagrammes de séquence.....	39
3.5.1 S'authentifier .....	39
3.5.2 Ajouter le projet .....	40
3.4.3 Ajouter le chercheur .....	41
3.5.4 Mise a jour d'un projet .....	42
3.5.4.1 Modification d'un projet .....	43

3.5.4.2 suppression d'un projet .....	43
3.5.5 Mis a jour chercheur .....	44
3.5.5.1 Modifie chercheur .....	44
3.5.5.2 Supprimé chercheur .....	45
3.5.6 Etat d'avance .....	46
3.5.7 Prolongation .....	47
3.5.8 Évaluation de l'état d'avancement .....	48
3.6 Le diagramme de classe .....	49
3.7 Conclusion .....	50

## **CHAPITRE 4 IMPLEMENTATION**

4.1 Introduction.....	52
4.2 Les outils de développement.....	52
4.2.1 L'environnement de développement Delphi .....	52
4.2.2 Implémentation de la base de données sous Access.....	53
4.3 Le passage du diagramme de classe au modèle relationnel.....	54
4.3.1 Les règles de passage .....	54
4.3.2 La base de données.....	55
4.4 Gestion de contrôle et de sécurité.....	56
4.4.1 Contrôle.....	56
4.4.2 Sécurité.....	56
4.5 Les interfaces de l'application.....	57
4.5.1 Interface d'authentification .....	57

4.5.2 Interface du menu principale (recherche scientifique).....	58
4.5.3 Interface projet.....	59
4.5.4 Interface chercheur .....	60
4.5.5 Interface soutenance.....	61
4.5.6 Interface l'état d'avance .....	62
4.5.7 Interface communication.....	63
4.6 Conclusion.....	<b>64</b>
<b>CONCLUSION GÉNÉRALE .....</b>	<b>60</b>
<b>BIBLIOGRAPHIE.....</b>	<b>61</b>

## LISTE DES FIGURES

<b>Figure 1.1:</b> historique de l'UML.....	5
<b>Figure 1.2:</b> diagramme de cas d'utilisation.....	7
<b>Figure 1.3:</b> diagramme de classe.....	8
<b>Figure 1.4:</b> diagramme de séquence.....	9
<b>Figure 1.5:</b> Quelle méthode pour passer de l'expression des besoins au code de l'application?.....	11
<b>Figure 1.6:</b> Les besoins sont modélisés par un diagramme de cas d'utilisation.....	12
<b>Figure 1.7:</b> Les diagrammes de séquence système illustrent la description textuelle des cas d'utilisation.....	13
<b>Figure 1.8:</b> Une maquette d'IHM facilite les discussions avec les futurs utilisateurs.....	13
<b>Figure 1.9:</b> La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classes.....	14
<b>Figure 1.10:</b> Le diagramme de classes participantes effectue la jonction entre les cas d'utilisation, Le modèle du domaine et les diagrammes de conception logicielle.....	16
<b>Figure 1.11:</b> Les diagrammes d'activités de navigation représentent graphiquement les activités de navigation dans l'interface.....	17
<b>Figure 1.12:</b> Les diagrammes d'interaction permettent d'attribuer précisément les responsabilités de comportement aux classes d'analyse.....	18
<b>Figure 1.13:</b> Le système des diagrammes de séquences système, vu comme une boîte noire, est remplacé par un ensemble d'objets en collaboration.....	19
<b>Figure 1.14:</b> Chaîne complète de la démarche de modélisation du besoin jusqu'au code.....	19
<b>Figure 2.1 :</b> Organigramme du centre universitaire de Mila.....	22
<b>Figure 2.2 :</b> hiérarchie du SARV.....	23
<b>Figure 3.1 :</b> Diagramme de cas d'utilisation.....	37
<b>Figure 3.2 :</b> Diagramme de séquence du cas d'utilisation Authentification.....	38
<b>Figure 3.3 :</b> Diagramme de séquence du cas d'utilisation ajouté projet.....	39
<b>Figure 3.4 :</b> Diagramme de séquence du cas d'utilisation ajouté chercheur.....	40
<b>Figure 3.5 :</b> Diagramme de séquence du cas d'utilisation modifie un projet.....	41

<b>Figure 3.6 :</b> Diagramme de séquence du cas d'utilisation supprimé projet.....	42
<b>Figure 3.7 :</b> Diagramme de séquence du cas d'utilisation modifie chercheur.....	43
<b>Figure 3.8 :</b> Diagramme de séquence du cas d'utilisation supprimé chercheur.....	44
<b>Figure 3.9 :</b> Diagramme de séquence du cas d'utilisation état d'avance.....	45
<b>Figure 3.10 :</b> Diagramme de séquence du cas d'utilisation prolongation.....	46
<b>Figure 3.11 :</b> Diagramme de séquence du cas d'utilisation évaluation de l'état d'avancement.....	47
<b>Figure 3.12 :</b> Diagramme de Classe.....	48
<b>Figure 4.1 :</b> Interface authentification.....	56
<b>Figure 4.2 :</b> Interface menu principale (Gestion de projet de recherche).....	57
<b>Figure 4.3 :</b> Interface ajouté projet.....	58
<b>Figure 4.4 :</b> Interface chercheur.....	59
<b>Figure 4.5 :</b> Interface de la thèse.....	60
<b>Figure 4.6 :</b> Interface l'état d'avance.....	61
<b>Figure 4.7 :</b> Interface communication.....	62

## **LISTE DES TABLEAUX**

<b>Tableau 2.1: Fiche d'étude du document : contrat de travail.....</b>	<b>27</b>
<b>Tableau 2.2: Fiche d'étude du document : rapport de prime .....</b>	<b>27</b>
<b>Tableau 2.3: Fiche d'étude du document : canevas de projet.....</b>	<b>28</b>
<b>Tableau 2.4: Fiche d'étude du document : demande de prolongation.....</b>	<b>29</b>
<b>Tableau 2.5 : Fiche d'étude du document : Fiche de résumé du travail scientifique....</b>	<b>30</b>
<b>Tableau 3.1 : Les cas d'utilisations.....</b>	<b>32</b>

## LISTE DES ACRONYMES

UML	Unified Modeling Language.
UP	Unified Process.
IHM	Interface Homme-Machine.
SARV	Service des Activités de Recherche et de la Valorisation.
CUM	Centre Universitaire de Mila.
PNR	Projets Nationaux de Recherche.
CNEPRU	Comité Nationale d'Évaluation et de Programmation de la Recherche Universitaire
RAD	Rapide Application Développement.
ODBC	Open Data Base Connectivite

# *Introduction générale*

## **1 Contexte de travail**

Suite à son développement incessant, l'informatique occupe aujourd'hui une place considérable dans tous les domaines et ce grâce aux applications informatiques qui proposent des solutions à tous les problèmes de la vie, aussi bien dans des domaines professionnels que pour des applications personnelles.

La gestion des projets de recherche du est un exemple d'un système qui nécessite une automatisation. Les fonctionnalités de ce système sont nombreuses, on peut citer à titre d'exemple : la réception des projets de recherches, le suivi des états d'avancement des projets, la gestion des prolongations...Etc.

## **2 Problématique et motivation**

Au cours de notre stage au niveau du SARV du centre universitaire de Mila, nous avons constaté que la majorité des traitements se fait manuellement, ce qui a créé pas mal de problèmes :

- La difficulté de suivre la réalisation des projets de recherche.
- La perte de temps lors de la recherche des informations.
- Le risque de perdre des informations.
- L'absence d'une base de données pour sauvegarder les informations des projets.

Devant ces obstacles, il est nécessaire de construire un système d'information pour bien gérer les projets de recherche , faciliter la tâche du chef de service et assurer l'accès vif aux données.

## **3 Objectif de travail**

Notre objectif consiste alors à la conception et le développement d'une application de gestion de recherche permettant de répondre aux besoins du chef de service. Plus précisément, elle prend en charge la gestion des projets dès que le chercheur commence son projet jusqu'à la présentation d'état final.

Pour atteindre cet objectif, nous avons utilisé le langage UML (Unified Modeling Language) dans la phase d'analyse tout en étant encadrés par la démarche proposée par Laurent

AUDIBERT [1]. Pour l'implémentation, notre choix s'est porté sur l'environnement de développement DELPHI et la base de données est implémentée avec ACCESS.

#### **4 Organisation du mémoire**

Ce mémoire est structuré de la manière suivante :

- ◆ *Le chapitre I* s'intéresse à présenter le langage de modélisation UML et à décrire la démarche de développement utilisée.
- ◆ *Le chapitre II* est dédié à présenter l'organisme d'accueil qui est le centre Universitaire De Mila dans lequel nous avons effectué notre stage. Il consiste aussi à étudier les différents postes de travail et les divers documents manipulés au sein du SARV, pour y'arriver enfin à construire un cahier de charge et ce, dans le but de préparer la conception et le développement de l'application.
- ◆ *Le chapitre III* est consacré à l'analyse de l'étude de cas.
- ◆ *Le chapitre IV* sert à présenter les outils dont nous nous sommes servis pour le développement de l'application ainsi qu'une brève description de quelques interfaces de cette application.
- ◆ Ce modeste mémoire s'achève par une conclusion générale en présentant un récapitulatif de tout ce que nous avons réalisé.

# Chapitre 01

***Le langage UML***

## 1.1 Introduction

Ce chapitre présente en premier lieu les concepts de base du langage de modélisation UML (Unified Modeling Languages). Nous commençons par une description d'UML, en suite, nous citons quelques avantages qui le rendent le langage de modélisation le plus répandu. Ainsi, nous énumérons ses diagrammes avec une description de ceux les plus utilisés dans la modélisation. Cette étude permette une bonne lecture de ce qui va suivre et exprime de manière uniforme l'analyse, la conception et la réalisation d'une application informatique. Finalement, nous allons présenter la méthode de modélisation explicitant et encadrant toutes les étapes suivies pour la réalisation de notre projet.

## 1.2 Le langage de modélisation unifié (UML)

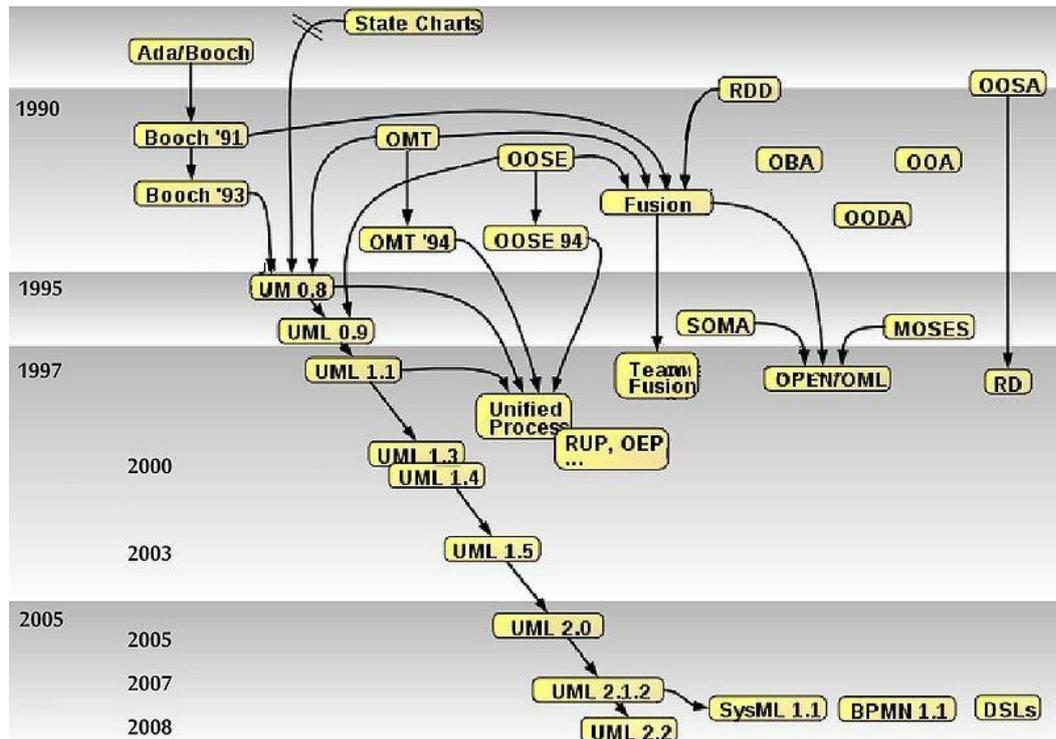
### 1.2.1 Définition

UML est une notion graphique conçue pour représenter, spécifier, construire et documenter les systèmes logiciels. Ses deux principaux objectifs sont la modélisation de système utilisant les techniques de l'orienté objet, depuis la conception jusqu'à la maintenance, et la création du langage abstrait compréhensible par l'homme et intraitable par les machines. Il permet de construire plusieurs model d'un système, chacun d'eux met en valeur des aspects différentes : fonctionnels, statiques, dynamiques, organisationnels. UML est devenu un langage incontournable dans les projets de développement.

### 1.2.2 Historique

1. Les années 1980: Utilisation de méthodes adaptées à la programmation impérative (notamment Merise)
2. Fin 80 / début 90 la programmation par objets prend de l'importance.
3. Conséquence naturelle: mise en place de méthodes orientées objet. Plus de cinquante méthodes apparaissent entre 1990 et 1995:
  - OOD : Object Oriented Design (Brooch, 1991)
  - HOOD: Hierarchical Object Oriented Design (Delattee et al. 1993)
  - OOA: Object Oriented Analysis (Schlaer, Mellor, 1992)
  - OOA/OOD: (Coad, Yourdon, 1991)
  - OMT: Object Modeling Technique (Rumbaugh, 1991)

- OOSE: Object Oriented Software Engineering (Jacobson, 1992)
- OOM: Object Oriented Merise (Bouzeghoub, Rochfeld, 1993)
- Fusion (Coleman et al. 1994)



1.1 Figure : historique de l'UML

### 1.2.3 Les avantages de UML

- ❖ Il permet de représenter l'aspect traitement du système aussi bien que l'aspect données ;
- ❖ Il n'est nullement réservé à l'orientation objets, mais se prête également à l'analyse traditionnelle ou à une approche mixte; il permet éventuellement une transition en douceur vers l'orientation objets;
- ❖ Il remplace toutes les notations existantes, y compris Yourdon (Diagrammes de flux de données et entités - associations, etc.), Merise et autres;
- ❖ Il représente un standard de modélisation, une notation. Il s'agit donc d'un outil et non d'une méthode; chacun peut utiliser la méthode la plus appropriée à ses besoins (processus, techniques, outils);
- ❖ Il s'utilise sur l'ensemble du cycle de développement de logiciels (analyse, conception, réalisation, tests, documentation);

- ❖ Il se prête à la construction de nouvelles applications, à la documentation d'applications existantes et à la définition de cahiers des charges pour la sélection de solutions du marché;
- ❖ Les principaux outils de modélisation permettent aujourd'hui de travailler avec cette notation;

#### 1.2.4 Les Diagrammes d'UML

UML dans sa version 2 s'articule autour de treize diagrammes, chacun d'entre eux est appliqué à la représentation d'un système logiciel suivant un point de vue particulier. Ces diagrammes sont regroupés dans deux grands ensembles: les diagrammes structurels et les diagrammes de comportement.

➤ **Diagrammes structurels ou diagrammes statiques**

- ◆ Diagramme de classes
- ◆ Diagramme d'objets
- ◆ Diagramme de composants
- ◆ Diagramme de déploiement
- ◆ Diagramme de paquetages
- ◆ Diagramme de structures composites

➤ **Diagrammes comportementaux ou diagrammes dynamiques**

- ◆ Diagramme de cas d'utilisation
- ◆ Diagramme d'activités
- ◆ Diagramme d'états-transitions
- ◆ Diagramme de séquence
- ◆ Diagramme de communication
- ◆ Diagramme global d'interaction
- ◆ Diagramme de temps

Ces diagrammes, d'une utilité variable selon les cas, ne sont pas nécessairement tous produits à l'occasion d'une modélisation. Les plus utiles sont les diagrammes d'activités, de cas d'utilisation, de classes, d'objets, de séquence et d'états-transitions.

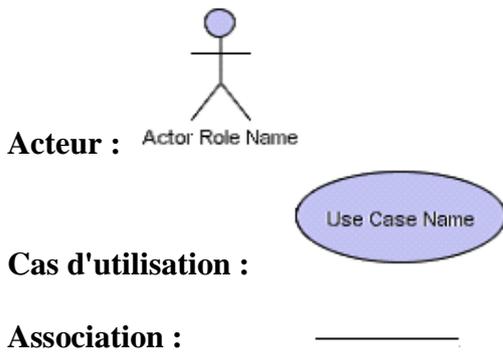
## 1.2.5 Description de quelques diagrammes

### 1.2.5.1 Le diagramme de cas d'utilisation

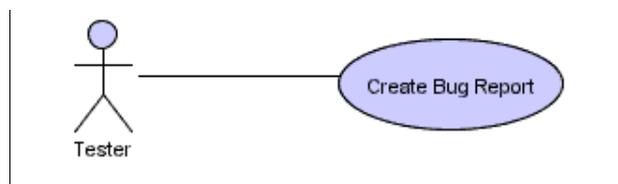
Les diagrammes des cas d'utilisation identifient les fonctionnalités fournies par le système (cas d'utilisation), les utilisateurs qui interagissent avec le système (acteurs), et les interactions entre ces derniers. C'est en effet une vue statique de l'utilisation d'un système.

Les cas d'utilisation sont utilisés dans la phase d'analyse pour définir les besoins de "haut niveau" du système. Les objectifs principaux des diagrammes des cas d'utilisation sont:

- fournir une vue de haut-niveau de ce que fait le système
- Identifier les utilisateurs ("acteurs") du système
- Déterminer des secteurs nécessitant des interfaces homme-machine(IHM).



L'image suivante montre comment ces trois éléments de base collaborent pour former un diagramme de cas d'utilisation:



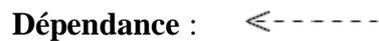
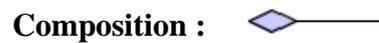
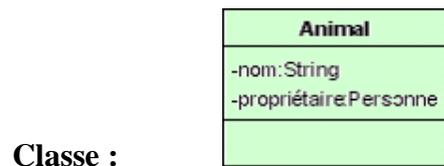
1.2 Figure :diagramme de cas d'utilisation



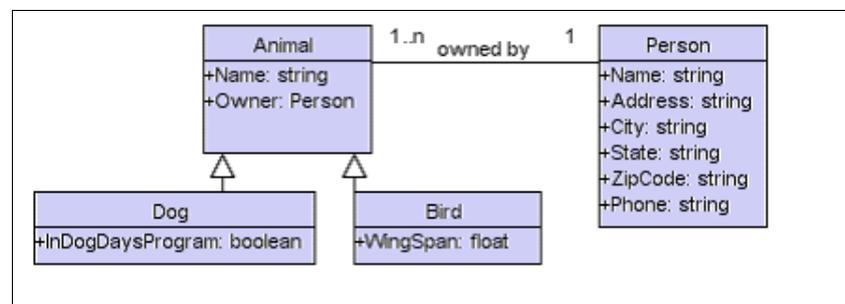
### 1.2.5.2 Le diagramme de classes

Le diagramme des classes identifie la structure de la classe d'un système, y compris les propriétés et les méthodes de chaque classe et les diverses relations, telles que la relation d'héritage par exemple.

Il représente un ensemble de classes, d'interfaces et de collaboration ainsi que leurs relations. Les diagrammes de classes représentent la vue statique d'un système.



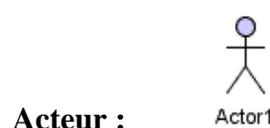
La figure suivante montre un exemple de diagramme de classe avec une relation de généralisation.

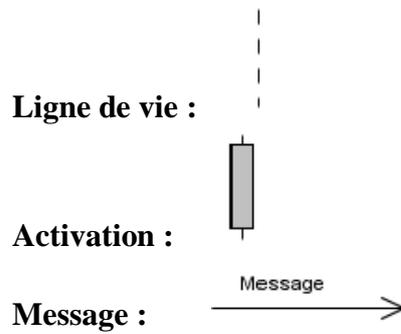


1.3 Figure : diagramme de classe

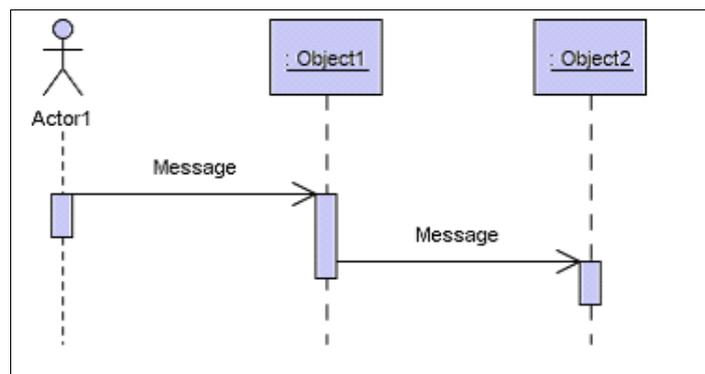
### 1.2.5.3 Le diagramme de séquences

Les diagrammes des séquences documentent les interactions à mettre en œuvre entre les classes pour réaliser un résultat, ces communications entre les classes sont reconnues comme des messages. Le diagramme des séquences énumère des objets horizontalement, et le temps verticalement. Il modélise l'exécution des différents messages en fonction du temps, donnant ainsi une vue dynamique sur le système. Les diagrammes de séquence mettent l'accent sur le classement chronologique des messages





L'exemple ci-dessous représente un diagramme des séquences qui utilise des objets par défaut (aucun nom n'est spécifié)



1.4 Figure : diagramme de séquence

#### 1.2.5.4 Diagramme d'états-transitions

Les diagrammes d'états-transitions d'UML décrivent le comportement interne d'un objet à l'aide d'un automate à états finis. Ils présentent les séquences possibles d'états et d'actions qu'une instance de classe peut traiter au cours de son cycle de vie en réaction à des événements discrets.

Un diagramme d'états-transitions rassemble et organise les états et les transitions d'un classeur donné.

**-État** : Un état représente une période dans la vie d'un objet pendant laquelle ce dernier attend un événement ou accomplit une activité, se représente graphiquement dans un diagramme d'états transitions par un rectangle aux coins arrondis.

**\_ État initial** : L'état initial est un pseudo état qui indique l'état de départ, par défaut, Lorsque le diagramme d'états-transitions, ou l'état enveloppant, est invoqué. Lorsqu'un

Objet est créé, il entre dans l'état initial.

**\_ État final** : L'état final est un pseudo état qui indique que le diagramme d'états. Transitions, ou l'état enveloppant, est terminé.

-**Événement** : Un événement de changement est généré par la satisfaction (i.e. passage de faux à vrai) d'une expression booléenne sur des valeurs d'attributs.

-**Transition** : Une transition définit la réponse d'un objet à l'occurrence d'un événement. Elle lie, généralement, deux états E1 et E2 et indique qu'un objet dans un état E1 peut entrer dans l'état E2 et exécuter certaines activités, si un événement déclencheur se produit et que la condition de garde est vérifiée.

-**Condition de garde** : La condition de garde est évaluée uniquement lorsque l'événement déclencheur se produit. Si l'expression est fautive à ce moment-là, la transition ne se déclenche pas, si elle est vraie, la transition se déclenche et ses effets se produisent.

### 1.2.5.5 Diagramme d'activités

Les diagrammes d'activités permettent de spécifier des traitements très proches des langages de programmation objet : spécifier des actions de base, structure de contrôle (condition, boucle), Ils sont donc bien adaptés à la spécification détaillée enchaînements d'actions de haut niveau, en particulier pour la description détaillée des cas d'utilisation.

-**Activité** : Une activité définit un comportement décrit par un séquençage organisé d'unités dont les éléments simples sont les actions. Une activité est un comportement (behaviorisme en anglais) et à ce titre peut être associée à des paramètres. De la gauche vers la droite, on trouve: le nœud représentant une action, un nœud objet, un nœud de décision ou de fusion, un nœud de bifurcation ou d'union, un nœud initial, un nœud final et un nœud final de flot.

-**Transition** : Le passage d'une activité vers une autre est matérialisé par une transition.

Graphiquement les transitions sont représentées par des flèches en traits pleins qui connectent les activités entre elles. Les transitions spécifient l'enchaînement des traitements et définissent le flot de contrôle.

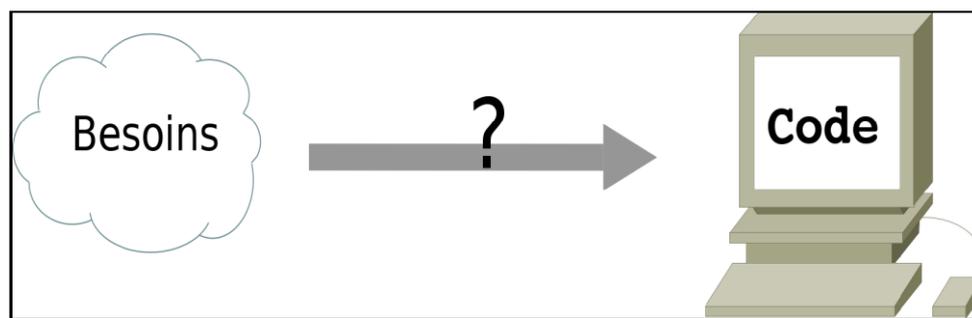
-**La synchronisation** Les flots de contrôle parallèles sont séparés ou réunis par des barres de synchronisation qui peuvent être des :

- **Débranchements**: les transitions qui partent d'un débranchement ont lieu en même temps.
- **Jonctions**: on ne franchit une jonction qu'après la réalisation de toutes les transitions qui s'y rattachent.

### 1.3 Mise en œuvre d'UML

UML n'est pas une méthode et ne propose pas une démarche de modélisation explicitant et encadrant toutes les étapes d'un projet. Il n'est qu'un langage de modélisation, ce qui justifie le besoin de faire appel à une méthode qui permet le passage de l'expression des besoins au code de l'application. Nous allons donc présenter une des méthodes proposées dans la littérature. Il s'agit d'une méthode simple et générique proposée par LAURENTAUDIBERT

Elle se situe à mi-chemin entre *UP (Unified Process)*, qui constitue un cadre général très complet de processus de développement, et *XP (eXtremeProgramming)* qui est une approche minimaliste à la mode centrée sur le code.



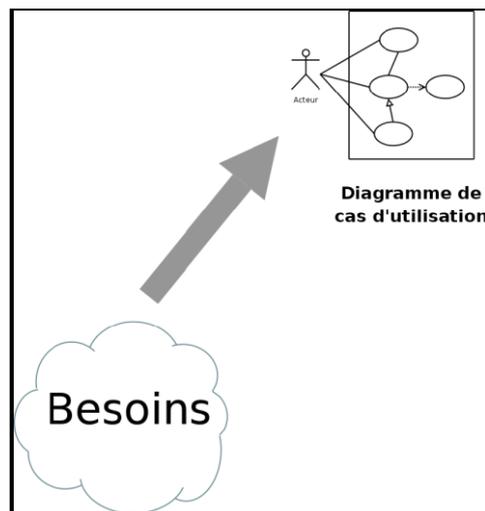
**1.5 Figure :** Quelle méthode pour passer de l'expression des besoins au code de l'application?

#### 1.3.1 Identification des besoins et spécification des fonctionnalités

##### 1.3.1.1 Identification et représentation des besoins diagramme de cas d'utilisation

Les cas d'utilisation sont utilisés tout au long du projet. Dans un premier temps, on les crée pour identifier et modéliser les besoins des utilisateurs. Ces besoins sont déterminés à partir des informations recueillies lors des rencontres entre informaticiens et utilisateurs. Durant cette étape, il faut déterminer les limites du système, identifier les acteurs et recenser les cas d'utilisation.

Les interactions entre les acteurs et le système (au sein des cas d'utilisation) seront explicitées sous forme textuelle et sous forme graphique au moyen de diagrammes de séquence.



**1.6 Figure:** Les besoins sont modélisés par un diagramme de cas d'utilisation.

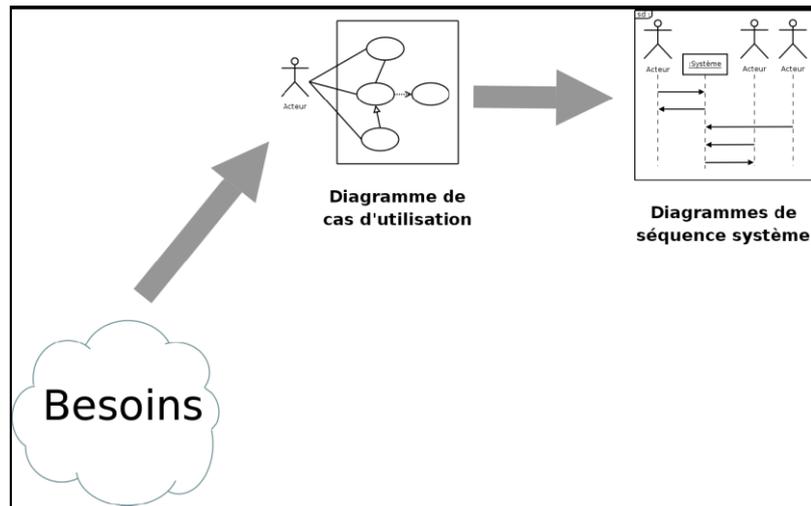
### 1.3.1.2 Spécification détaillée des besoins Diagrammes de séquence système

Dans cette étape, on cherche à détailler la description des besoins par la description textuelle des cas d'utilisation, et la production de diagrammes de séquence système illustrant cette description textuelle. Cette étape conduit souvent à mettre à jour le diagramme de cas d'utilisation puisque nous sommes toujours dans la spécification des besoins.

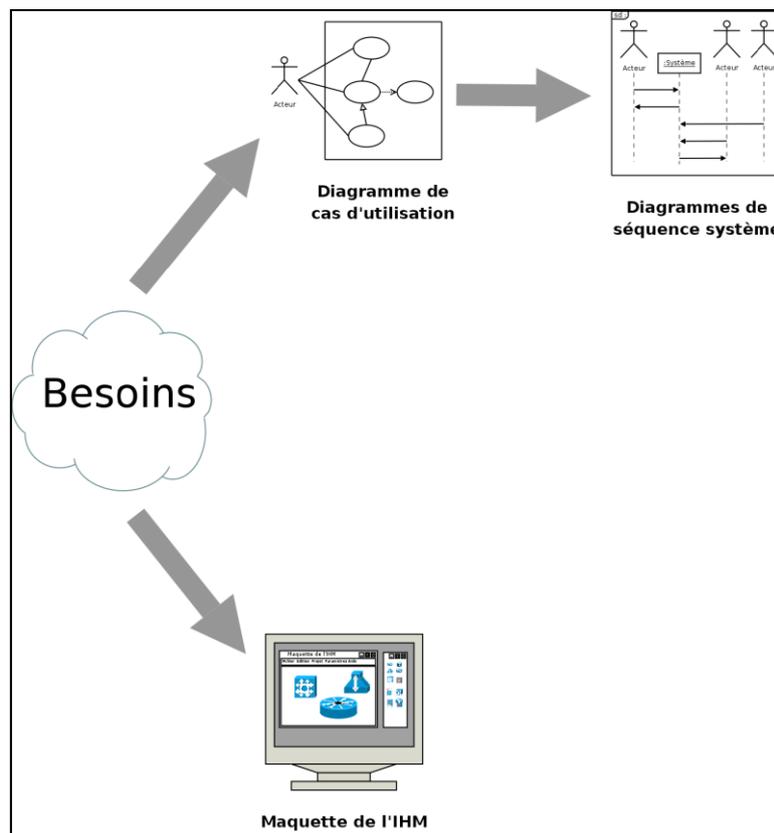
Les scénarios de la description textuelle des cas d'utilisation sont illustrés par des diagrammes de séquence système. Il faut, au minimum, représenter le scénario nominal de chacun des cas d'utilisation par un diagramme de séquence qui spécifier l'interaction entre l'acteur, ou les acteurs, et le système. Le système est ici considéré comme un tout et est représenté par une ligne de vie. Chaque acteur est également associé à une ligne de vie.

### 1.3.1.3 Maquette de l'IHM de l'application (non couvert par UML)

Une maquette d'IHM (Interface Homme-Machine) est un produit jetable permettant aux utilisateurs d'avoir une vue concrète mais non définitive de la future interface de l'application. La maquette peut très bien consister en un ensemble de dessins produits par un logiciel de présentation ou de dessin. Par la suite, la maquette pourra intégrer des fonctionnalités de navigation permettant à l'utilisateur de tester l'enchaînement des écrans ou des menus, même si les fonctionnalités restent fictives. La maquette doit être développée rapidement afin de provoquer des retours de la part des utilisateurs.



**1.7 Figure:** Les diagrammes de séquence système illustrent la description textuelle des cas d'utilisation.



**1.8 Figure:** Une maquette d'IHM facilite les discussions avec les futurs utilisateurs.

### 1.3.2 Phases d'analyse

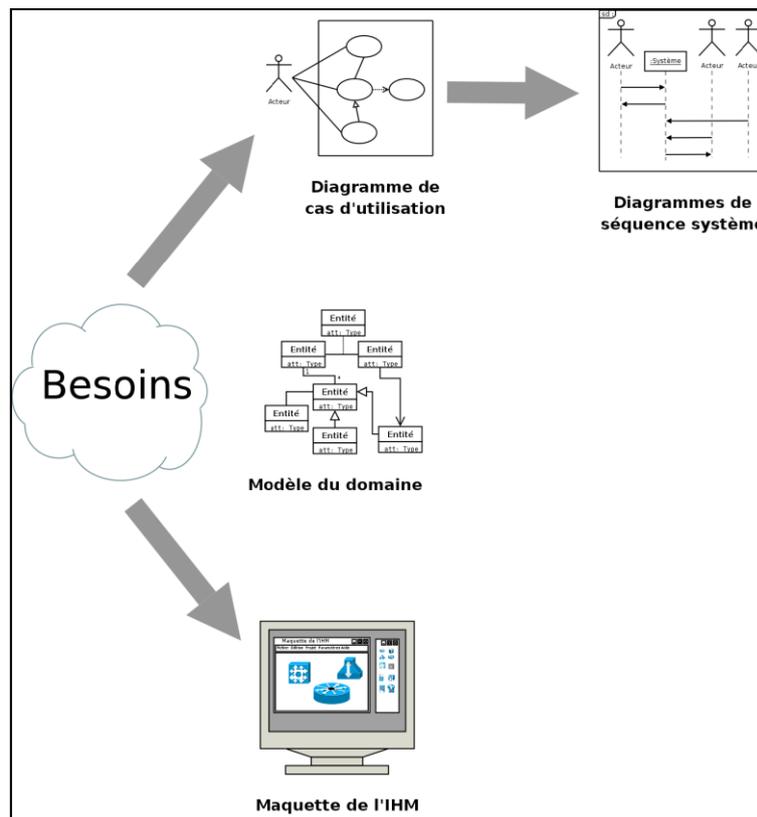
#### 1.3.2.1 Analyse du domaine : modèle du domaine

L'élaboration du modèle des classes du domaine permet d'opérer une transition vers une véritable modélisation objet. L'analyse du domaine est une étape totalement

dissociée de l'analyse des besoins. Elle peut être menée avant, en parallèle ou après cette dernière.

La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classes appelée modèle du domaine. Ce modèle doit définir les classes qui modélisent les entités ou concepts présents dans le domaine de l'application. Il s'agit donc de produire un modèle des objets du monde réel dans un domaine donné. Ces entités ou concepts peuvent être identifiés directement à partir de la connaissance du domaine ou par des entretiens avec des experts du domaine.

Les classes du modèle du domaine ne doivent pas contenir d'opération, mais seulement les attributs.



**1.9 Figure :** La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classes.

### 1.3.2.2 Diagramme de classes participantes

Le diagramme de classes participantes est particulièrement important puisqu'il effectue la jonction entre, d'une part, les cas d'utilisation, le modèle du domaine et la maquette, et d'autre part, les diagrammes de conception logicielle que sont les

diagrammes d'interaction et le diagramme de classes de conception (voir figure 1.6). Le diagramme de classes participantes modélise trois types de classes d'analyse sont :

- **Les classes de dialogues :** Les classes qui permettent les interactions entre l'IHM et les utilisateurs sont qualifiées de *dialogues*. Ces classes sont directement issues de l'analyse de la maquette. Il y a au moins un dialogue pour chaque association entre un acteur et un cas d'utilisation. En général, les dialogues vivent seulement le temps du déroulement du cas d'utilisation concerné.
- **Les classes de contrôles :** Les classes qui modélisent la cinématique de l'application sont appelées *contrôles*. Elles font la jonction entre les dialogues et les classes métier en permettant aux différentes vues de l'application de manipuler des informations détenues par un ou plusieurs objets métier. Elles contiennent les règles applicatives et les isolent à la fois des dialogues et des entités.
- **Les classes entités :** Les classes métier, qui provient directement du modèle du domaine, sont qualifiées *d'entités*. Ces classes sont généralement persistantes, c'est-à-dire qu'elles survivent à l'exécution d'un cas d'utilisation particulier et qu'elles permettent à des données et des relations d'être stockées dans des fichiers ou des bases de données. Lors de l'implémentation, ces classes peuvent ne pas se concrétiser par des classes mais par des relations, au sens des bases de données relationnelles.

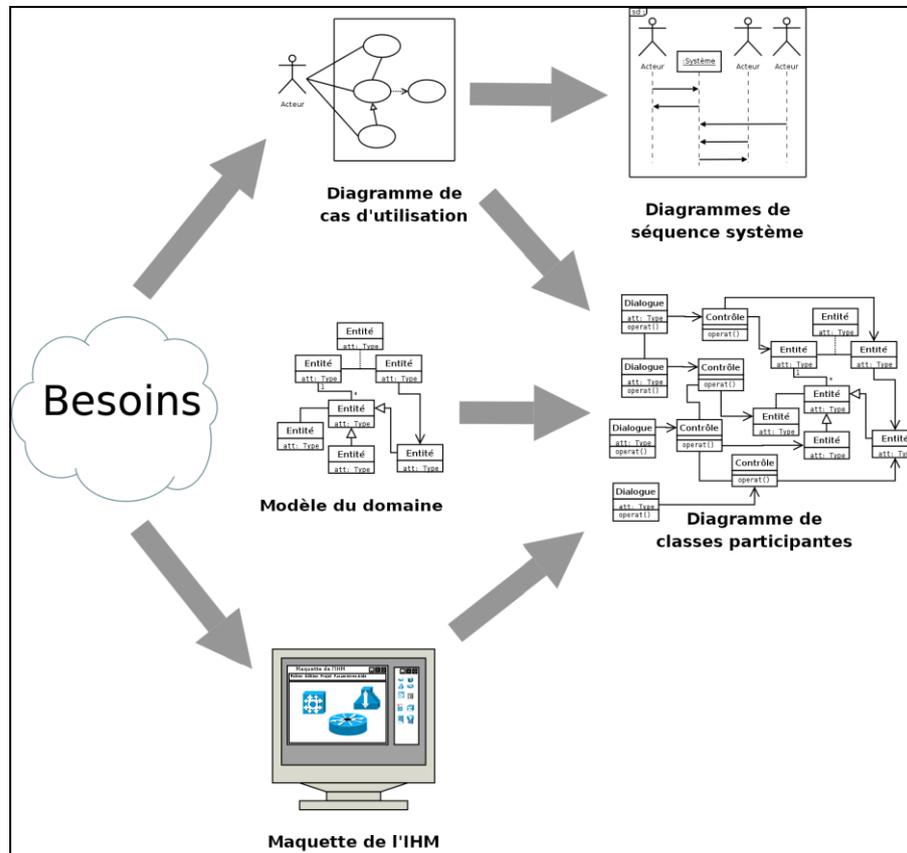
### 1.3.2.3 Diagrammes d'activités de navigation

Les IHM modernes facilitent la communication entre l'application et l'utilisateur en offrant toute une gamme de moyens d'action et de visualisation comme des menus déroulants ou contextuels, des palettes d'outils, des boîtes de dialogues, des fenêtres de visualisation, etc. Cette combinaison possible d'options d'affichage, d'interaction et de navigation aboutis aujourd'hui à des interfaces de plus en plus riches et puissantes.

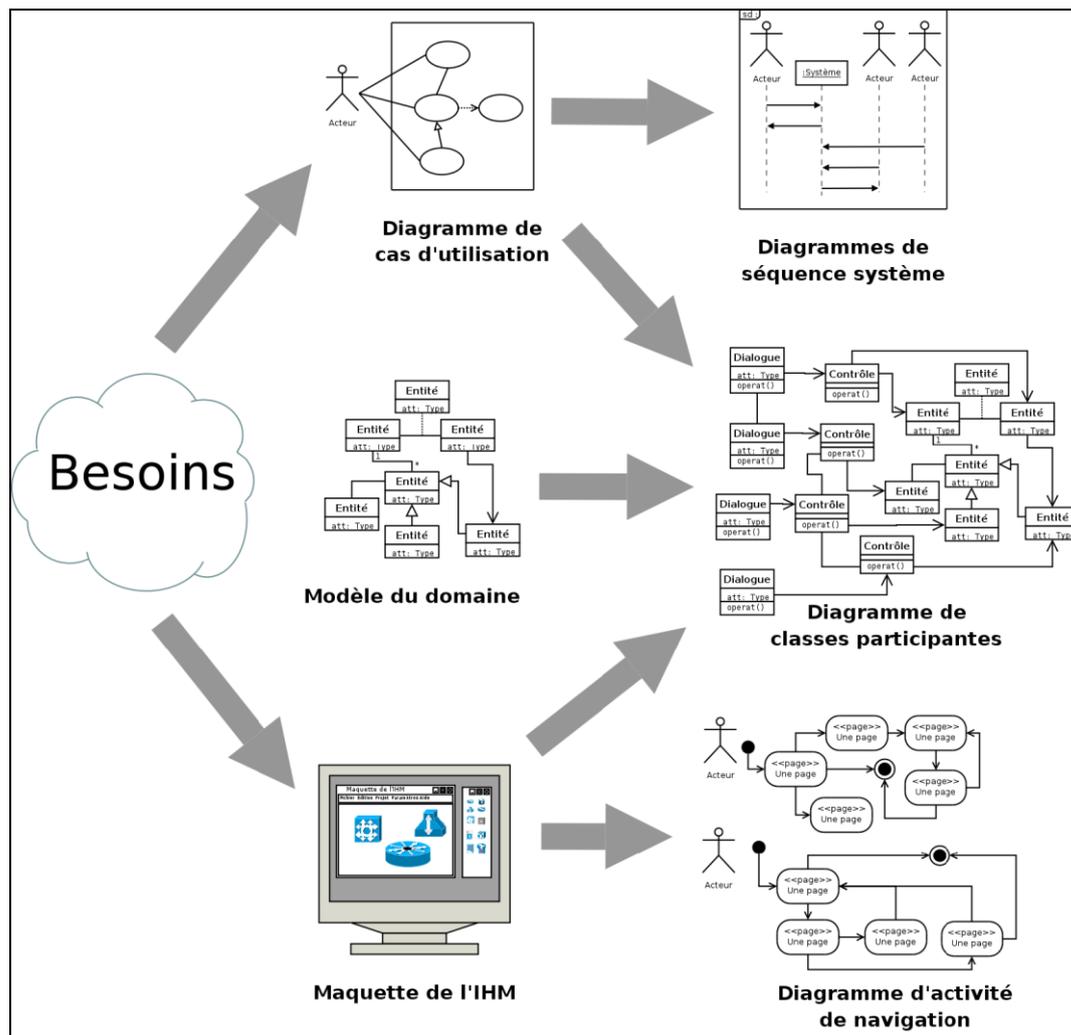
UML offre la possibilité de représenter graphiquement les activités de navigation dans l'interface en produisant des diagrammes dynamiques. On appelle ces diagrammes des diagrammes de navigation .

Les diagrammes d'activités de navigation sont à relier aux classes de dialogue du diagramme de classes participantes. Les différentes activités du diagramme de navigation peuvent être stéréotypées en fonction de leur nature : «fenêtre», «menu», «menu contextuel», «dialogue», etc.

La modélisation de la navigation à intérêt à être structurée par acteur.



**1.10 Figure :** Le diagramme de classes participantes effectue la jonction entre les cas d'utilisation, Le modèle du domaine et les diagrammes de conception logicielle.



**1.11 Figure:** Les diagrammes d'activités de navigation représentent graphiquement les activités de navigation dans l'interface.

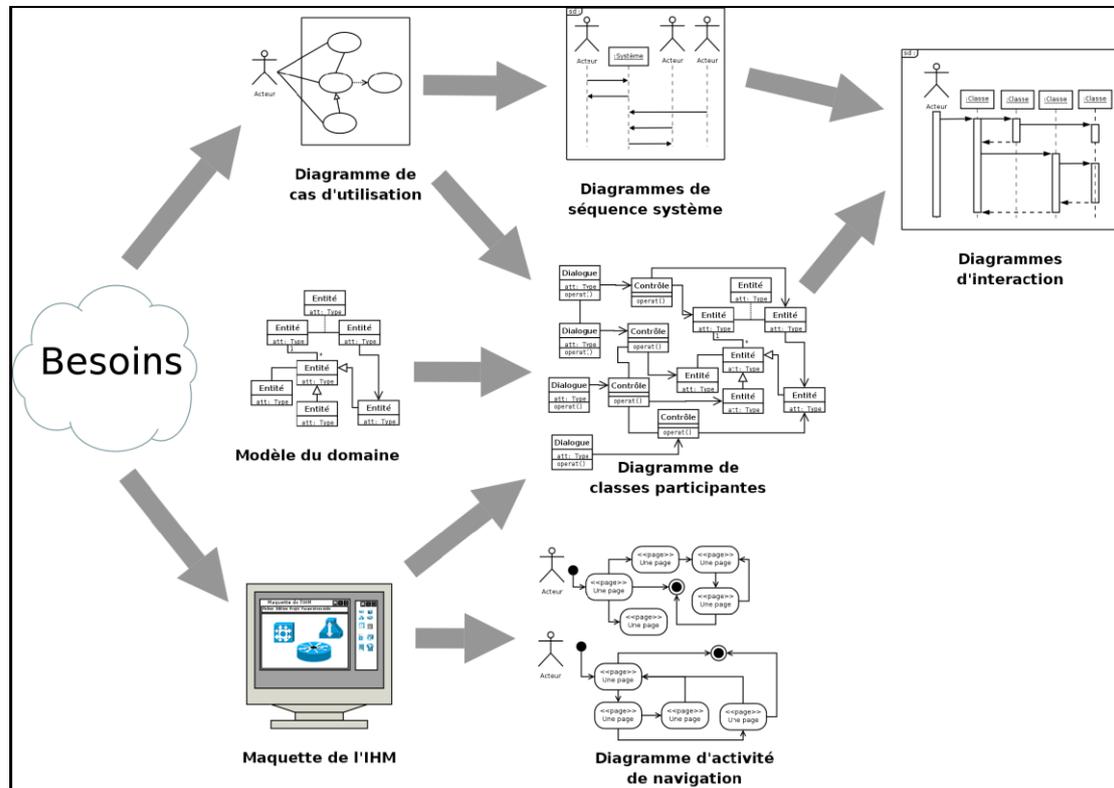
### 1.3.3 Phase de conception

#### 1.3.3.1 Diagrammes d'interaction

Maintenant, il faut attribuer précisément les responsabilités de comportement, dégagée par le diagramme de séquence système, aux classes d'analyse du diagramme de classes participantes. Les résultats de cette réflexion sont présentés sous la forme de diagrammes d'interaction UML (voir figure 1.12). Parallèlement, une première ébauche de la vue statique de conception, c'est-à-dire du diagramme de classes de conception, est construite et complétée.

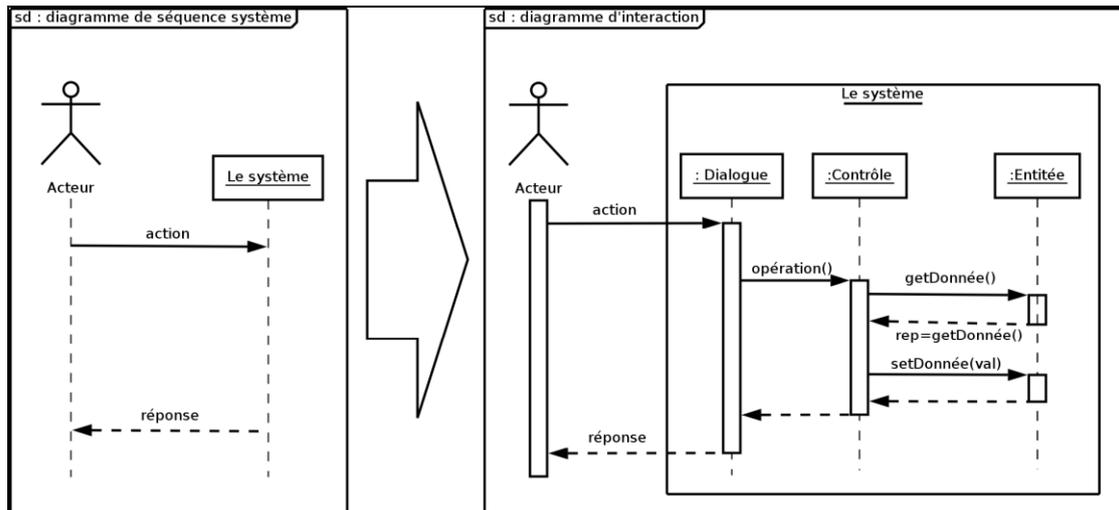
Durant cette phase, l'ébauche du diagramme de classes de conception reste indépendante des choix technologiques qui seront faits ultérieurement.

Les diagrammes d'interactions sont utiles au concepteur pour décider quelle est la classe qui va contenir chaque service ou fonction. Dans les diagrammes d'interaction, les objets communiquent en s'envoyant des messages qui invoquent des opérations sur les objets récepteurs. Il est ainsi possible de mettre en œuvre l'allocation des responsabilités à partir des diagrammes d'interaction.



**1.12 Figure:** Les diagrammes d'interaction permettent d'attribuer précisément les responsabilités de comportement aux classes d'analyse.

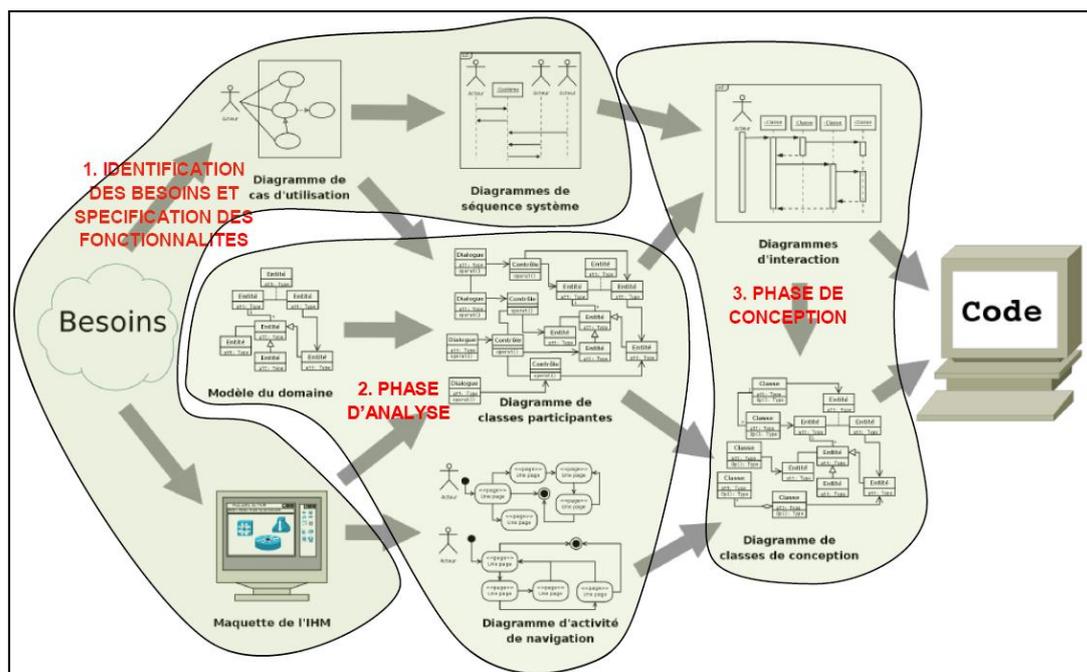
Par rapport aux diagrammes de séquences système, nous remplaçons ici le système, vu comme une boîte noire, par un ensemble d'objets en collaboration. Ces objets sont des instances des trois types de classes d'analyse du diagramme de classes participantes, à savoir des dialogues, des contrôles et des entités.



**1.13 Figure:** Le système des diagrammes de séquences système, vu comme une boîte noire, est remplacé par un ensemble d’objets en collaboration.

### 1.3.3.2 Diagramme de classes de conception

L’objectif de cette étape est de produire le diagramme de classes qui servira pour l’implémentation. Une première ébauche du diagramme de classes de conception a déjà été élaborée en parallèle du diagramme d’interaction. Il faut maintenant le compléter en précisant les opérations privées des différentes classes. Il faut prendre en comptes les choix techniques, comme le choix du langage de programmation.



**1.14 Figure:** Chaîne complète de la démarche de modélisation du besoin jusqu’au code.

## **1.4 Conclusion**

A l'issu de ce chapitre, nous nous sommes adaptés avec l'outil de modélisation UML d'une part et la méthode proposée par Laurent AUDIBERT [1] d'autre part, ce qui va nous guider pour la réalisation des prochaines étapes de notre projet.

# Chapitre 02

*Etude de l'existant*

## 2.1 Introduction

L'étude de l'existant constitue une étape indispensable dans tout projet informatique. Elle représenté un travail de terrain grâce auquel, on pourrait connaitre de prés, la situation actuel de l'organisation. A cet effet, nous procédons, par la description des différents postes de travail puis nous faisons une analyse des documents utilisés au niveau du service. Concernant note étude, elle portera sur le Service des activités de recherche et de la valorisation (SARV).

## 2.2 Présentation de l'organisme d'accueil (CENTRE UNIVERSITAIRE DE MILA)

Le Centre Universitaire de Mila est un établissement public à caractère scientifique, culturel et professionnel doté de la personnalité morale et de l'autonomie financière. Il a été crée par le Décret exécutif N°204-08 du 09 juillet 2008. Donnant ainsi naissance à l'enseignement supérieur dans la wilaya.

Il est situé à 5km du centre-ville de Mila sur la route nationale N°79 allant vers les communes de zeghaia et de ferdjioua et s'étend sur une superficie d'environ 87 hectares.

Le Centre Universitaire de Mila propose durant l'année 2009-2010 une formation supérieure dans les domaines suivants :

- Mathématiques et Informatique.
- Science de la Nature et de la Vie.
- Sciences économiques, Commerciales et Sciences de Gestion.
- Langues et Littérature Arabe.
- Anglais.

Le diagramme suivant présente l'organigramme du centre universitaire de Mila :

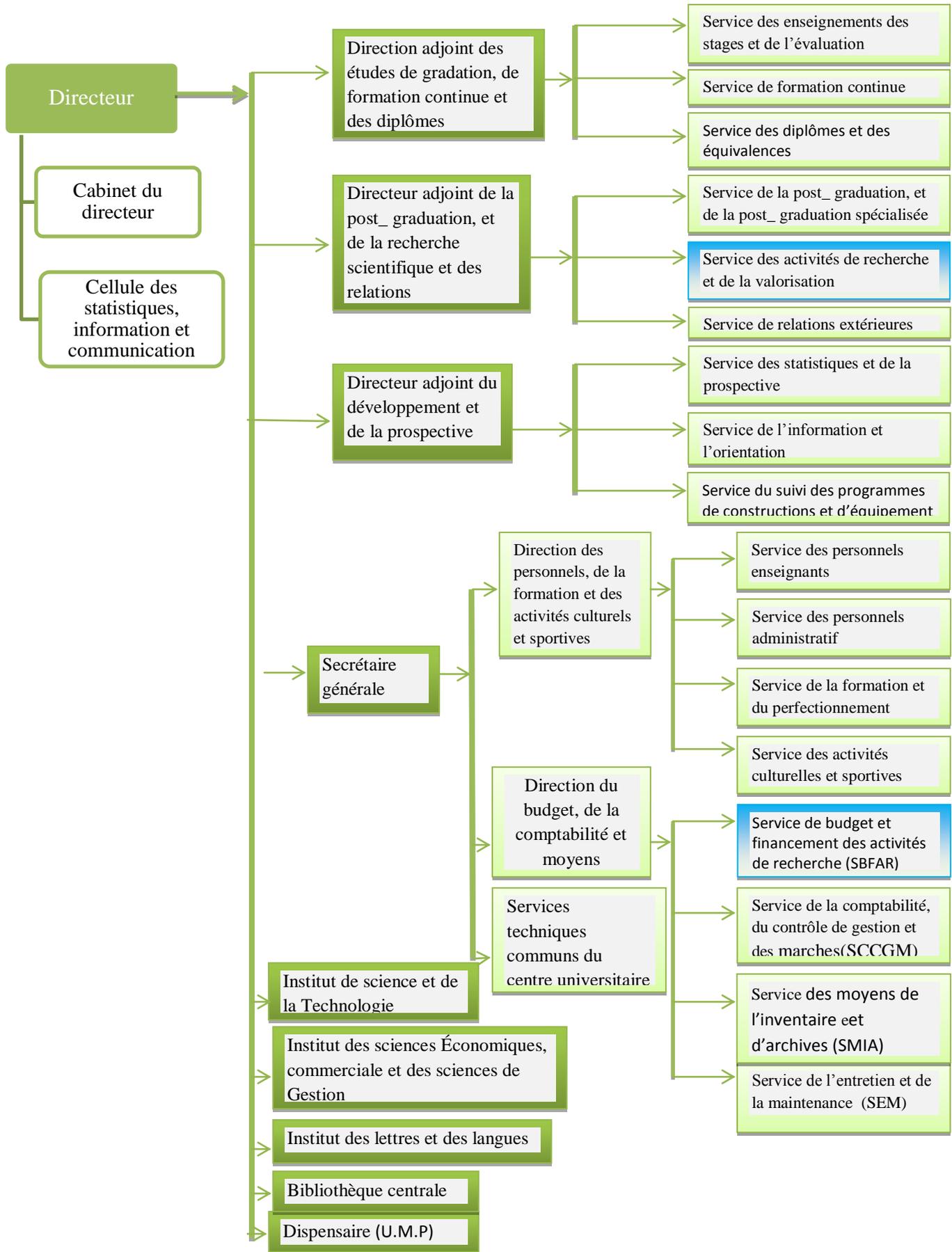
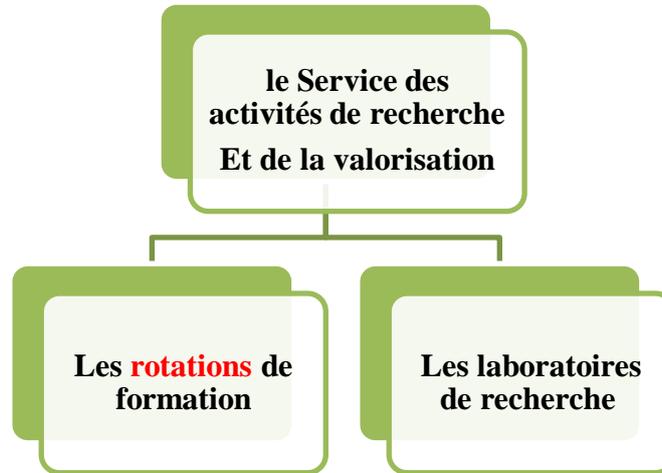


Figure 2.1 : Organigramme du centre universitaire de Mila.

### 2.3 Cadre d'étude

Notre stage contient est effectué au niveau du service des activités de recherche et de la valorisation. La figure suivante montre les activités effectuées au niveau de ce service.



**Figure 2.2** : hiérarchie du SARV

Ce qui nous intéresse est la gestion administrative des projets de recherche au niveau de ce dernier qui consiste à suivre les différents projets depuis leur propositions jusqu'à leur réalisations. Cette activité est effectuée par le responsable du SARV avec sa secrétaire.

Les projets géré sont des projets PNR (projets nationaux de recherche) CNEPRU ( ) et ceux effectué au niveau des laboratoires du CUM.

Les laboratoires de recherche sont des centres de recherche académiques ou particuliers. Ils servent à améliorer le niveau de recherche et de l'enseignement supérieur, ils se composent de plusieurs unités de cherche. Le CUM ne contient que deux laboratoires, un pour le département de mathématique et l'autre pour le département de biologie.

Il existe aussi deux laboratoires nationaux :

- **CNEPRU** : Les organismes de recherche où les équipes de recherche universitaire académique, qui est aussi une exigence nationale du 03 aux 06 membres et le but des chercheurs étudiants d'escorte dans la préparation de leur enseignement et de recherche démontrent les difficultés et les faire contrôler les professeurs.
- **PNR** : Projets de recherche nationaux sont des projets attribués par les autorités nationales pour répondre aux questions sur la table dans la supervision économique, social, culturel et autres, par le professeur de l'enseignement supérieur et de la

recherche scientifique et sa mission n'est pas de l'académie, mais le processus nécessite le partenaire économique.

## **2.4 Description des tâches effectuées au niveau du service:**

Les activités effectuées au niveau du SNVR sont :

- ♦ L'étude des nouveaux projets de recherche.
- ♦ Le suivi des projets de recherche en cours de réalisation.
- ♦ La gestion des demandes de prolongation du délai.

### **2.4.1 L'étude des nouveaux projets**

Les chercheurs remplissent un canevas de proposition d'un nouveau projet déposé par le ministère contenant les renseignements nécessaires concernant les membres de l'équipe ainsi que le thème du projet. Par la suite, il sera adressé vers le conseil scientifique de chaque département du center universitaire qui va l'accepter ou le refuser après étude. Les projets ayant reçu l'avis favorable seront envoyés au SARV. Le chef de SARV vérifie et valide les projets après la validation du conseil scientifique puis les transmis au ministère où ils vont être évalué et agréés.

### **2.4.2 L'évaluation des projets en cours de réalisation**

Les chercheurs doivent rendre compte des avancements des travaux au conseil scientifique en envoyant un bilan où il sera mentionné les résultats obtenues durant 6 mois. Une fois accepté par le conseil scientifique, il sera adressé au SARV pour vérification puis transmis au ministère pour évaluation.

### **2.4.3 Gestion des demandes de prolongation**

Après expiration du délai fixé pour la réalisation du projet, les chercheurs peuvent solliciter une prolongation du délai pour l'achèvement de leur projet.

## **2.5 Étude de document**

Pour mieux maitriser les données manipulées par le système d'information, il faut bien analyser avec précaution les documents utilisés pour extraire toutes les informations qui peuvent figurer dans le système étudié.

Les documents manipulés sont :

- Canevas de projet ;
- Contrat de travail ;
- Rapport de prime ;
- Fiche de résumé du travail scientifique
- Fiche de prolongement

Pour mettre en évidence le volume d'informations que contiennent les documents, leurs structurations ainsi que leur contenu en information, une fiche d'étude été établie pour chaque document

**Fiche d'étude du document : contrat du travail**

Code et rôle du document				
<b>Code:</b> CT				
<b>Désignation:</b> contrat du travail				
<b>Rôle:</b> permet aux chercheurs de démarrer leur travail sur le projet.				
Caractéristiques du document				
<b>Origine:</b> le SARV				
<b>Destinataire:</b> chercheur.				
<b>Nombre d'exemplaires:</b> 2				
<b>Nature:</b> sortie				
Éléments d'information				
Rubrique	Désignation	Type	Taille	Observation
<b>NOM-DIR</b>	Le nom du directeur de CUM	A	20	
<b>PRE-DIR</b>	Le prénom du directeur de CUM	A	20	
<b>NOM-CHE</b>	Le nom du chercheur	A	20	
<b>PRE-CHE</b>	Le prénom du chercher	A	20	
<b>SPI-CHE</b>	Spécialité du chercheur	A	20	
<b>ADR-CHE</b>	Adresse du chercheur	AN	50	
<b>NUM-PRJ</b>	Le numéro de projet	AN	50	

<b>DAT_DEP</b>	La date de départ de projet	<b>N</b>	<b>50</b>	
<b>TIT_PRJ</b>	Le titre de projet	<b>A</b>	<b>50</b>	
<b>DEL_TC</b>	Le délai total du contrat	<b>AN</b>	<b>50</b>	
<b>DEL_RC</b>	Le délai de renouvellement du contrat	<b>AN</b>	<b>50</b>	
<b>PRIME</b>	La prime de recherche	<b>N</b>	<b>50</b>	

**Tableau 2.1: Fiche d'étude du document : contrat de travail**

**Fiche d'étude du document : rapport de prime**

<b>Code et rôle du document</b>				
<b>Code: RP</b>				
<b>Désignation :</b> rapport de prime.				
<b>Rôle:</b> la prime payé pour chaque chercheur.				
<b>Caractéristiques du document</b>				
<b>Origine: SARV</b>				
<b>Destinataire:</b> chercheur.				
<b>Nombre d'exemplaires: 3</b>				
<b>Nature:</b> entrée/sortie				
<b>Éléments d'information</b>				
<b>Rubrique</b>	<b>Désignation</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
<b>NOM-CHE</b>	Le nome du chercheur	<b>A</b>	<b>20</b>	
<b>PRE-CHE</b>	Prénom du chercheur	<b>A</b>	<b>20</b>	
<b>GRA-CHE</b>	grade du chercheur	<b>A</b>	<b>20</b>	MAA ,MAB,MC ,PES
<b>PRI-CHE</b>	La prime du chercheur	<b>N</b>	<b>50</b>	

**Tableau 2.2:Fiche d'étude du document : rapport de prime**

**Fiche d'étude du document : conva de projet**

<b>Code et rôle du document</b>
---------------------------------

<b>Code: CP</b>				
<b>Désignation:</b> conva de projet				
<b>Rôle:</b> demandez un nouveau projet.				
<b>Caractéristiques du document</b>				
<b>Origine:</b> chercheur				
<b>Destinataire:</b> SARV, conseil scientifique				
<b>Nombre d'exemplaires:</b> 2				
<b>Nature:</b> entré				
<b>Éléments d'information</b>				
Rubrique	Désignation	Type	Taille	Observation
<b>Nom Chef</b>	Nom du chef d'équipe	A	20	
<b>Pre_Chef</b>	Prénom du chef d'équipe	A	20	
<b>NOM-CHE</b>	Le nom de chercheur	A	20	
<b>PRE-CHE</b>	Prénom de chercheur	A	20	
<b>GRA-CHE</b>	grade de chercheur	A	20	MAA ,MAB,MC ,PES
<b>TIT-PRJ</b>	Le titre de projet	AN	50	
<b>SPE_CHE</b>	La spécialité de chercheur	A	50	
<b>COD-PRJ</b>	Le code de projet	AN	50	

**Tableau 2.3:Fiche d'étude du document : canevas de projet**

**Fiche d'étude du document : demande de prolongement**

<b>Code et rôle du document</b>				
<b>Code: DP</b>				
<b>Désignation:</b> demande de prolongement				
<b>Rôle:</b> demander une prolongation de délai pour un projet en cours.				
<b>Caractéristiques du document</b>				

<b>Origine:</b> chercheur				
<b>Destinataire:</b> chef de service.				
<b>Nombre d'exemplaires:</b> 1				
<b>Nature:</b> entré				
Éléments d'information				
Rubrique	Désignation	Type	Taille	Observation
<b>NOM-CHF</b>	Le nom de chef de service	A	20	
<b>PRE-CHF</b>	Le prénom de chef de service	A	20	
<b>NOM-CHE</b>	Le nom de chercheur	A	20	
<b>PRE-CHE</b>	Le prénom de chercher	A	20	
<b>SPI-CHE</b>	Spécialité de chercheur	A	20	
<b>ADR-CHE</b>	Adresse de chercheur	AN	50	
<b>COS-PRL</b>	La cause de prolongation	A	50	
<b>NUM-PRJ</b>	Le numéro de projet	AN	50	
<b>DIL-PRL</b>	Le délai de prolongation	N	50	1 année
<b>TIT_PRJ</b>	Le titre de projet	A	50	
<b>DAT_PRL</b>	La date de la demande de prolongation	N	50	

Tableau 2.4: Fiche d'étude du document : demande de prolongation

**Fiche d'étude du document :** Fiche de résumé de travail scientifique

Code et rôle du document
<b>Code:</b> RT
<b>Désignation :</b> Fiche de résumé de travail scientifique
<b>Rôle:</b> présente les résultats de travail.
Caractéristiques du document
<b>Origine:</b> chercheur
<b>Destinataire:</b> chef de service.

<b>Nombre d'exemplaires: 1</b>				
<b>Nature:</b> entré				
<b>Éléments d'information</b>				
<b>Rubrique</b>	<b>Désignation</b>	<b>Type</b>	<b>Taille</b>	<b>Observation</b>
<b>NOM-CHE</b>	Le nome de chercheur	<b>A</b>	<b>20</b>	
<b>PRE-CHE</b>	Prénom de chercheur	<b>A</b>	<b>20</b>	
<b>GRA-CHE</b>	grade de chercheur	<b>A</b>	<b>20</b>	MAA ,MAB,MC ,PES
<b>RSE-TRA</b>	Le résumé de travail	<b>A</b>	<b>100</b>	
<b>NUM-PRJ</b>	Le numéro de projet	<b>AN</b>	<b>50</b>	

**Tableau 2.5: Fiche d'étude du document :** Fiche de résumé du travail scientifique

## 2.6 Conclusion

Durant l'analyse de l'existant nous avons pu recenser toutes les informations nécessaires et indispensables à notre projet à savoir la conception et la réalisation d'un système de gestion de recherche au niveau de CUM. Elle nous a permis la compréhension des besoins de l'organisation et ce à travers l'analyse détaillée des postes de travail et les documents manipulés. Par conséquent, elle nous a aidé à bien fixer les objectifs afin qu'elle puisse être la base de la conception de ce système.

Dans le prochain chapitre nous aborderons l'étude conceptuelle de notre système.

# Chapitre 03

***Etude de CAS***

### 3.1 Introduction

L'objectif de cette étape est de déterminer de façon détaillée et précise ce que le système devra faire, afin de répondre aux objectifs établis lors de l'étude de l'existant, tout en respectant les contraintes établies préalablement.

### 3.2 Identification des cas d'utilisations

Le tableau suivant englobe les différents CU de ce système :

N°	Cas d'utilisation	Acteur
1	Authentification	Chef de SARV
2	Réception des projets agréés	Chef de SARV
3	Ajouter chercheur	Chef de SARV
4	Mise à jour projets	Chef de SARV
5	Mise à jour chercheurs	Chef de SARV
6	prolongation des projets	Chef de SARV
7	réception des états d'avancements	Chef de SARV
8	Evaluation des états d'avancements	Chef de SARV

Tableau 3.1 : Les cas d'utilisations.

### 3.3 Description des cas d'utilisation

Nous allons maintenant donner une description de chaque cas d'utilisation.

#### 3.3.1 Cas d'utilisation «authentification»

<p><b>Titre :</b> Authentification</p> <p><b>Finalité :</b> Ce cas permet au Chef de SARV d'accéder à l'application.</p> <p><b>Acteur :</b> Le Chef de SARV</p> <p><b>Pré condition :</b> Le Chef de SARV possède un compte.</p> <p><b>Enchaînements nominaux:</b></p> <ol style="list-style-type: none"> <li>1. le Chef de SARV lance l'application</li> <li>2. Le système lui affiche une fenêtre pour qu'il s'identifie.</li> <li>3. Le Chef de SARV saisie son login et son mot de passe.</li> <li>4. Le système vérifie leur validité puis lance le menu principal de l'application.</li> </ol> <p><b>Enchaînements alternatifs :</b></p> <ul style="list-style-type: none"> <li>▪ Login ou mot de passe erroné, l'authentification est demandée à nouveau au maximum trois fois.</li> </ul> <p><b>Enchaînements exceptionnels:</b></p> <ul style="list-style-type: none"> <li>▪ Le Chef de SARV ne saisit pas le bon mot de passe. l'application se ferme.</li> </ul> <p><b>Post conditions :</b></p> <ul style="list-style-type: none"> <li>▪ Le menu de l'application est accessible.</li> </ul>
--

#### 3.3.2 Cas d'utilisation « réception des projets»

<p><b>Titre :</b> réception des projets</p> <p><b>Finalité :</b> Ce cas permet d'ajouté un nouveau projet agréé.</p> <p><b>Acteur principal :</b> chef de SARV</p> <p><b>Pré condition :</b> -Le chef de SARV s'est authentifié.</p>
--

**Enchaînements nominaux:**

1. le chef de SARV demande l'ajouter d'un nouveau projet agréé.
2. Le système lui affiche le formulaire pour saisir les informations nécessaires.
3. Le chef de SARV remplit ce formulaire par les données concernant le projet. s'il s'agit d'un nouveau chercheur, le chef du SARV doit ajouter ce dernier.
4. Le système contrôle les informations saisies.
5. Le chef de SARV valide la saisie.
6. Le système enregistre l'ajout du projet.

**Enchaînements alternatifs :**

- Les informations sont incomplètes ou erronées.
- Retour au troisième enchaînement.

**Post conditions :**

- Le projet est ajouté.

### 3.3.3 Cas d'utilisation «ajouter chercheur»

**Titre :** ajouter un chercheur

**Finalité :** Ce cas permet d'ajouter un chercheur.

**Acteur :** Le chef de SARV

**Pré condition :** Le chef de SARV s'est authentifié.

**Enchaînements nominaux:**

1. Le chef de SARV demande l'ajout du nouveau chercheur.
2. Le système affiche le formulaire pour saisir les informations du chercheur (nom, prénom, email, grade...) ainsi que les mémoires et thèses encadrés par celui ci.
3. Le chef de SARV remplit ce formulaire.
4. Le système contrôle les informations saisies.
5. Le chef de SARV valide la saisie du formulaire.
6. Le système enregistre l'ajout du chercheur.

**Enchaînements alternatifs :**

- Les informations sont incomplètes ou erronées.
- Retour a le troisième enchaînement.

**Post conditions :**

Le chercheur est ajouté.

### 3.3.4 Cas d'utilisation «Mise à jour des projets»

#### 3.3.4.1 Modifier projet

**Titre :** Modifier projet

**Finalité :** Ce cas permet de modifier les informations d'un projet existant

**Acteur :** Le chef de SARV

**Pré condition :**

-Le chef de SARV s'authentifie.

-le projet concerné existe.

**Enchaînements nominaux:**

1. Le chef de SARV demande la visualisation de la liste des projets.
2. Le système affiche la liste de tous les projets existants.
3. Le chef de SARV sélectionne un projet pour modification
4. Le système lui affiche les informations de projet sélectionné qu'il peut modifier.
5. le chef de SARV saisit les modifications.

6. Le système contrôle la saisie des informations modifiées.
7. Le chef de SARV valide la modification.
8. Le système enregistre la modification.

**Enchaînements alternatifs :**

- Les informations sont incomplètes ou erronés.
- Retour a le cinquième enchainement.

**Post conditions :**

- Le projet est modifié.

### 3.3.4.2 Supprimer un projet

**Titre : Supprimer un projet**

**Finalité :** Ce cas permet de supprimer un projet existant

**Acteur :** Le chef de SARV

**Pré condition :** Le chef de SARV s'authentifie.

Le projet concerné existe.

**Enchaînements nominaux:**

1. Le chef de SARV demande la visualisation de la liste des projets.
2. Le système affiche la liste de tous les projets.
3. Le chef de SARV sélectionne le projet qu'il veut supprimer.
4. Le système informe le chef de SARV s'il veut vraiment le supprimer.
5. Le chef de SARV valide l'opération.

**Post conditions :**

- Le projet est supprimé.

### 3.3.5 Cas d'utilisation «Mise à jour des chercheurs»

#### 3.3.5.1 Modifier chercheur

**Titre :** Modifier un chercheur

**Finalité :** Ce cas permet de modifier les informations d'un chercheur existant

**Acteur :** Le chef de SARV

**Pré condition :**

- Le chef de SARV s'est authentifié.
- Le chercheur concerné existe.

**Enchaînements nominaux:**

1. Le chef de SARV demande la visualisation de la liste des chercheurs.
2. Le système affiche la liste de tous les chercheurs existants.
3. Le chef de SARV sélectionne un chercheur pour modification
4. Le système affiche les informations qu'il peut modifier concernant le chercheur sélectionné.
5. le chef de SARV saisie les modifications.
6. Le système contrôle la saisie des informations modifiées.
7. Le chef de SARV valide la modification.
8. Le système enregistre la modification.

**Enchaînements alternatifs :**

Les informations sont incomplètes ou erronés.

Retour a le cinquième enchainement.

**Post conditions :**

- Le chercheur est modifié.

### 3.3.5.2 Supprimer un chercheur

**Titre :** Supprimer un chercheur

**Finalité :** Ce cas permet de supprimer un chercheur existant

**Acteur :** Le chef de SARV

**Pré condition :**

- Le chef de SARV s'est authentifié.
- Le chercheur concerné existe.

**Enchaînements nominaux:**

1. Le chef de SARV demande la visualisation de la liste des chercheurs existants.
2. Le système affiche la liste de tous les chercheurs.
3. Le chef de SARV sélectionne le chercheur qu'il veut supprimer.
4. Le système informe le chef de SARV s'il veut vraiment le supprimer.
5. Le chef de SARV valide l'opération.

**Post conditions :**

- Le chercheur est supprimé.

### 3.3.6 Cas d'utilisation « prolongation des projets »

**Titre :** prolongation des projets

**Finalité :** Ce cas permet de prolonger les projets non achevés.

**Acteur :** Le chef de SARV

**Pré condition :**

- Le chef de SARV s'est authentifié.
- le projet existe.

**Enchaînements nominaux:**

1. Le chef de SARV demande de visualiser la liste des projets encours.
2. Le système affiche la liste.
3. Le chef de SARV sélectionné le projet pour prolongement.
4. Le système afficher les informations du projet sélectionné.
5. Le chef de SARV modifie la duré du projet.
6. Le système enregistre les informations.

**Enchaînements alternatifs :**

- Les informations sont incomplètes ou erronés ou la duré saisie dépasse la duré autorisé.
- Retour a le cinquième enchainement.

**Post conditions :**

- le projet est prolongé.

### 3.3.7 Cas d'utilisation « réception des états d'avancement»

**Titre :** réception des états d'avancement

**Finalité :** Ce cas permet d'introduire les résultats effectué pendant un semestre.

**Acteur :** Le chef de SARV

**Pré condition :**

- Le chef de SARV s'est authentifié.
- Le projet existe

**Enchaînements nominaux:**

- 1 Le chef de SARV demande la liste des projets encours.
- 2 Le système affiche la liste.

- 3 Le chef de SARV sélectionne le projet qu'il veut lui créer un état d'avancement.
- 4 Le système lui affiche les informations du projet sélectionné.
- 5 le chef de SARV demande d'ajouter un état d'avancement pour ce projet.
- 6 le système lui affiche un formulaire.
- 7 le chef de SARV saisit les informations de l'état d'avancement (date, résumé, les communications.....) puis valide l'ajout.
- 8 Le système ajoute l'état d'avancement.

**Enchaînements alternatifs :**

- Les informations sont incomplètes ou erronées.
- Retour au cinquième enchaînement.

**Post conditions :**

- état d'avancement créé pour le projet sélectionné.

**3.3.8 Cas d'utilisation «évaluation des états d'avancements »:**

**Titre :** évaluation des états d'avancement s

**Finalité :** Ce cas permet d'évaluer les bilans des projets.

**Acteur :** Le chef de SARV

**Pré condition :**

- Le chef de SARV s'est authentifié.
- L'état d'avancement d'un projet existe

**Enchaînements nominaux:**

- 1 Le chef de SARV demande la liste des projets en cours.
- 2 Le système affiche la liste des projets.
- 3 Le chef de SARV sélectionné le projet.
- 4 Le système affiche le formulaire.
- 5 Le chef de SARV demande l'affichage du dernier état d'avancement.
- 6 Le système affiche le formulaire.
- 7 Saisir l'avis de l'état d'avancement
- 8 Le chef de SARV valide les informations.
- 9 Le système enregistre les informations.
- 10 s'il s'agit du dernier état d'avancement et il est accepté, le système met le projet en état achevé

**Enchaînements alternatifs :**

- Les informations sont incomplètes ou erronées.
- Retour a le cinquième enchaînement.

**Post conditions :**

- état d'avancement du évalué.

### 3.4 Diagramme de cas d'utilisation

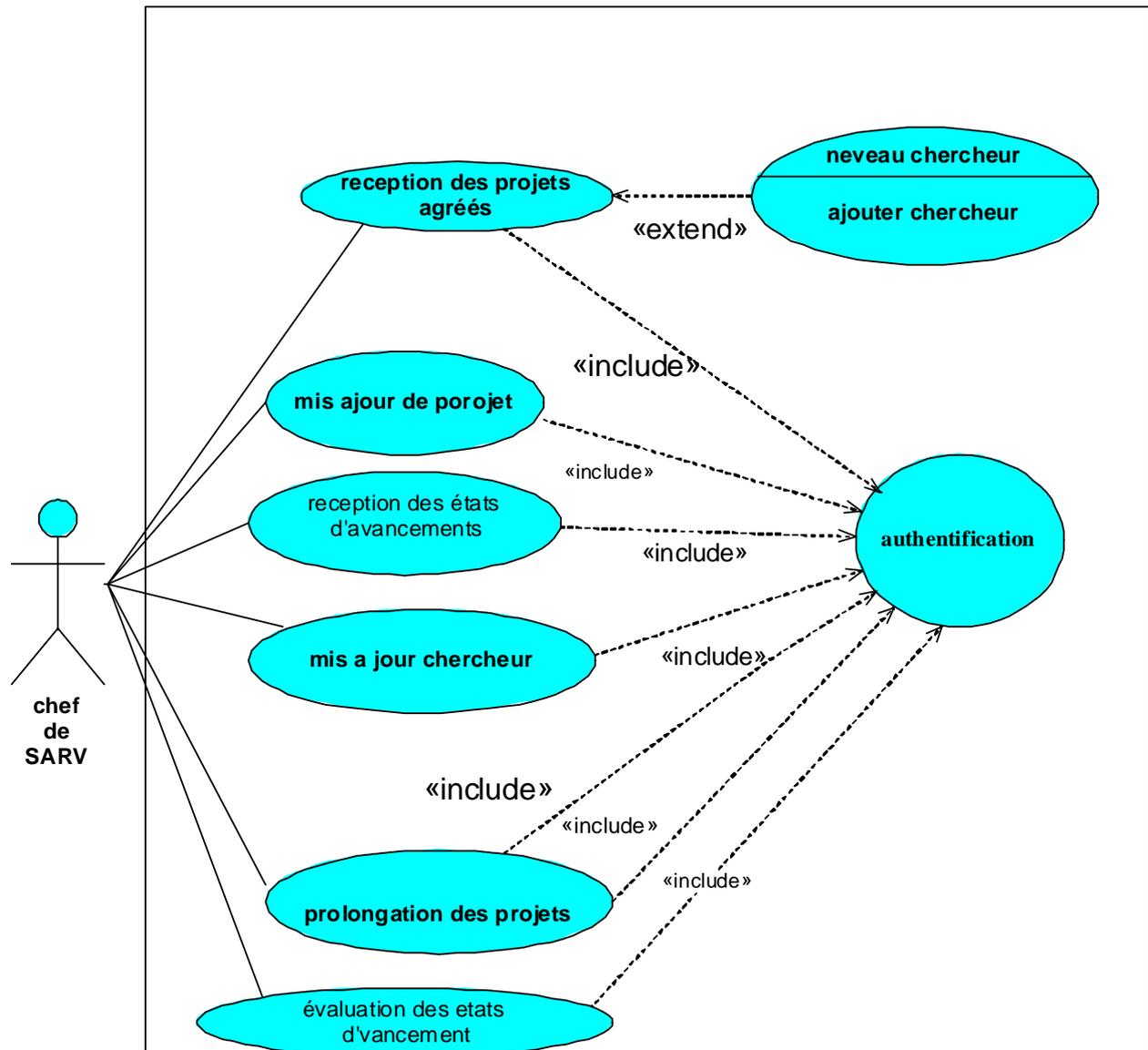


Figure 3.1 : Diagramme de cas d'utilisation

### 3.5 Les diagrammes de séquence

#### 3.5.1 S'authentifier

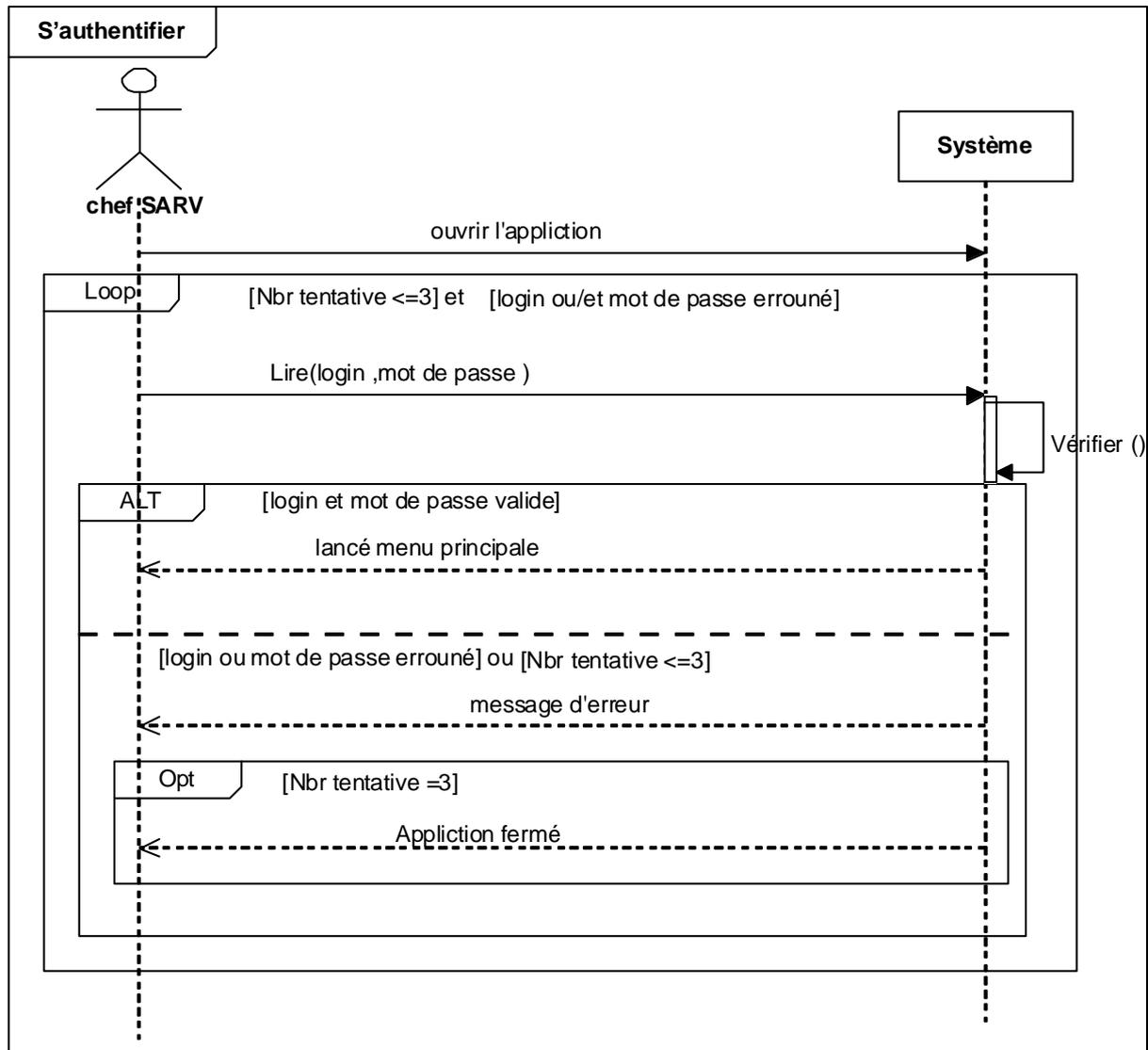


Figure3.2 : Diagramme de séquence du cas d'utilisation Authentification

### 3.5.2 Réception des projets

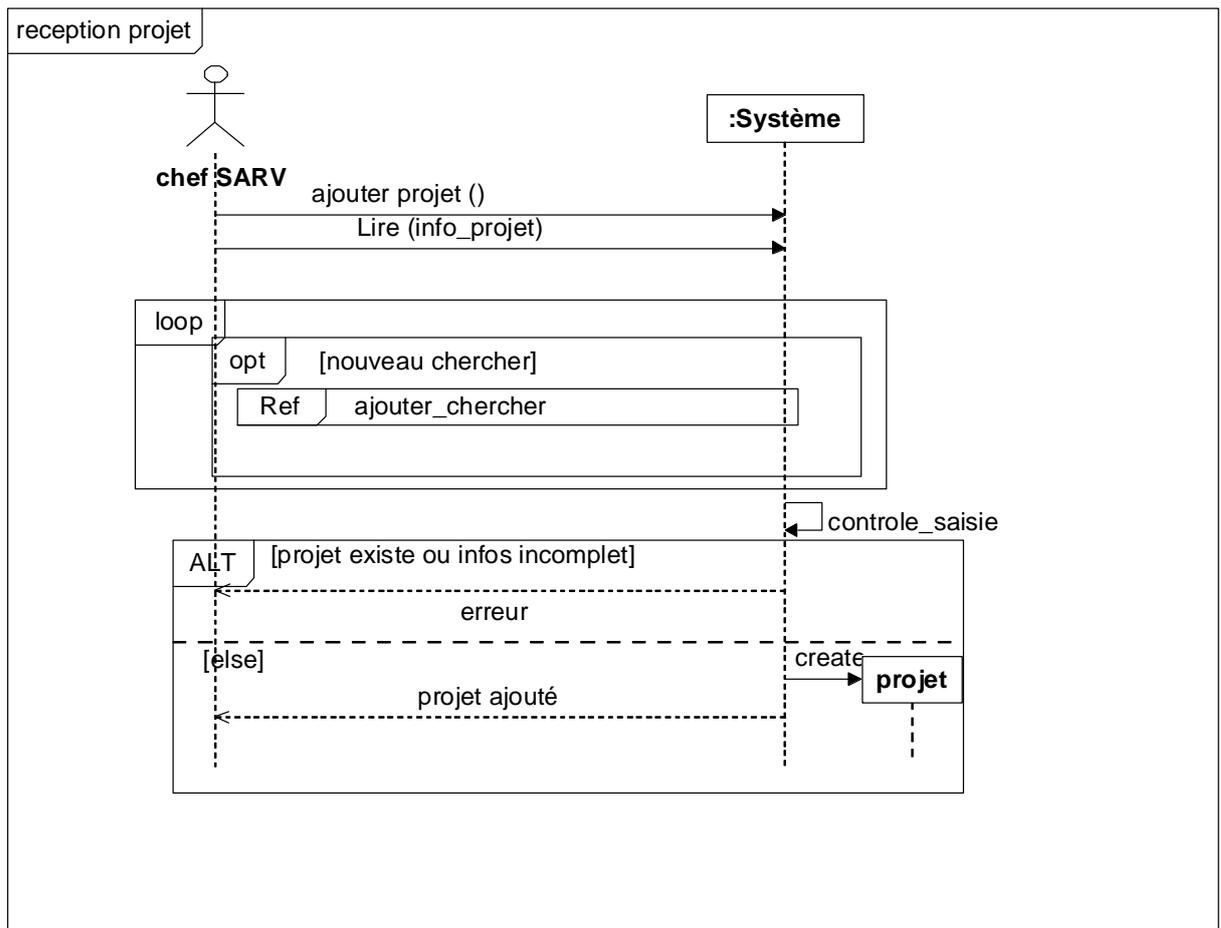


Figure 3.3 : Diagramme de séquence du cas d'utilisation réception des projets agréés.

### 3.5.3 Ajouter le chercheur

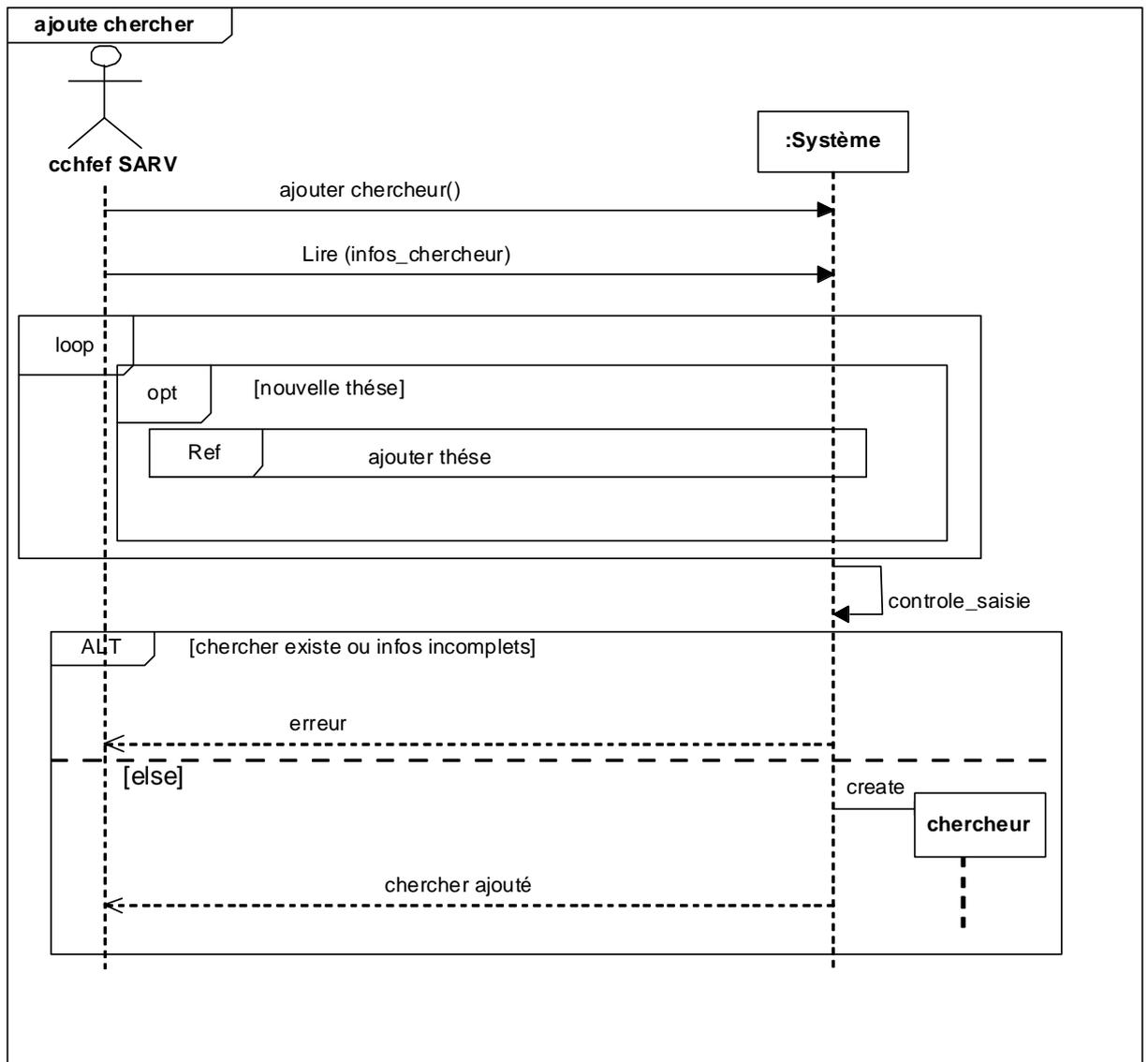


Figure 3.4 : Diagramme de séquence du cas d'utilisation ajouté chercheur.

### 3.5.4 Mise à jour d'un projet

#### 3.5.4.1 Modification d'un projet

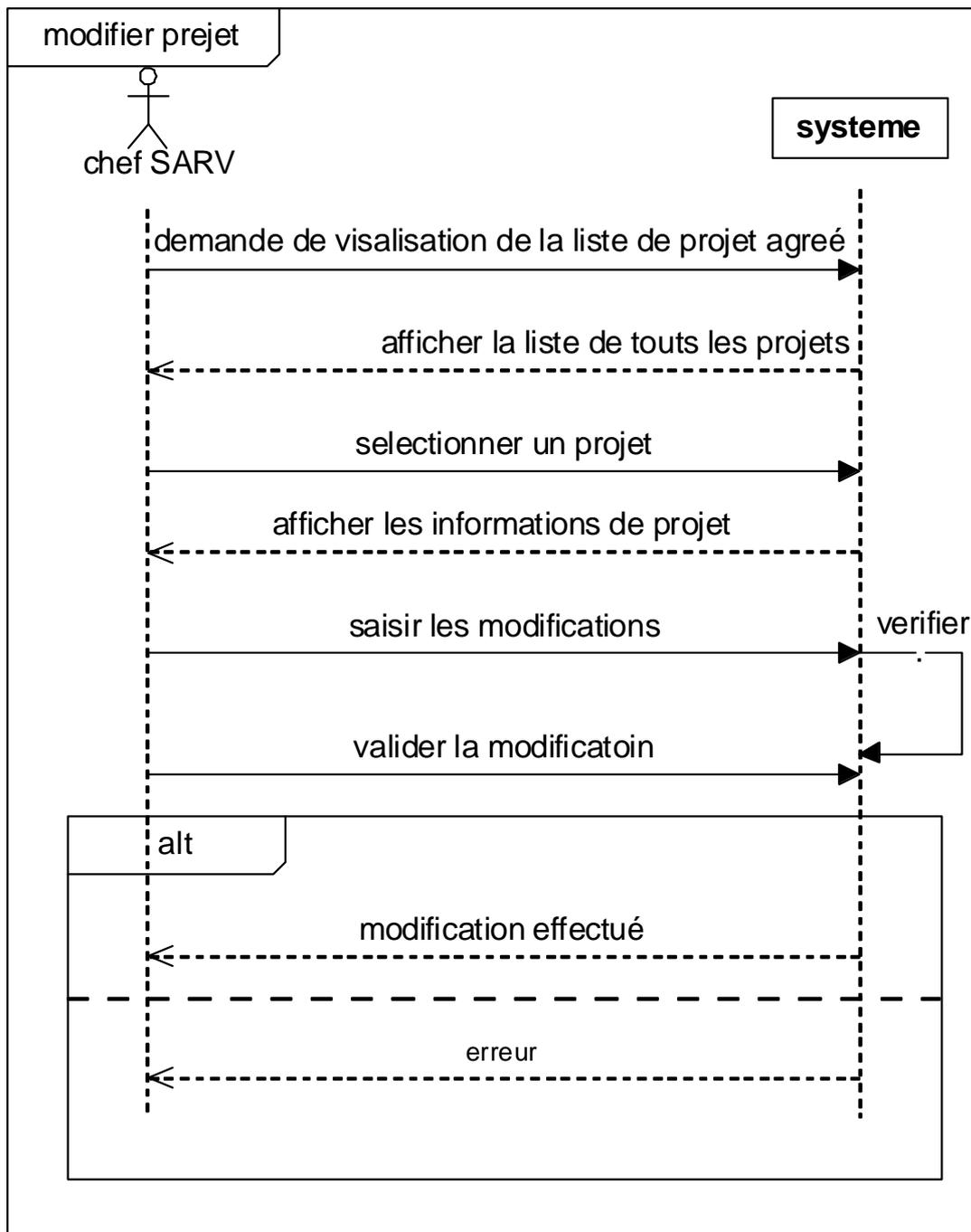


Figure 3.5 : Diagramme de séquence du cas d'utilisation modifie un projet.

### 3.5.4.2 Suppression d'un projet

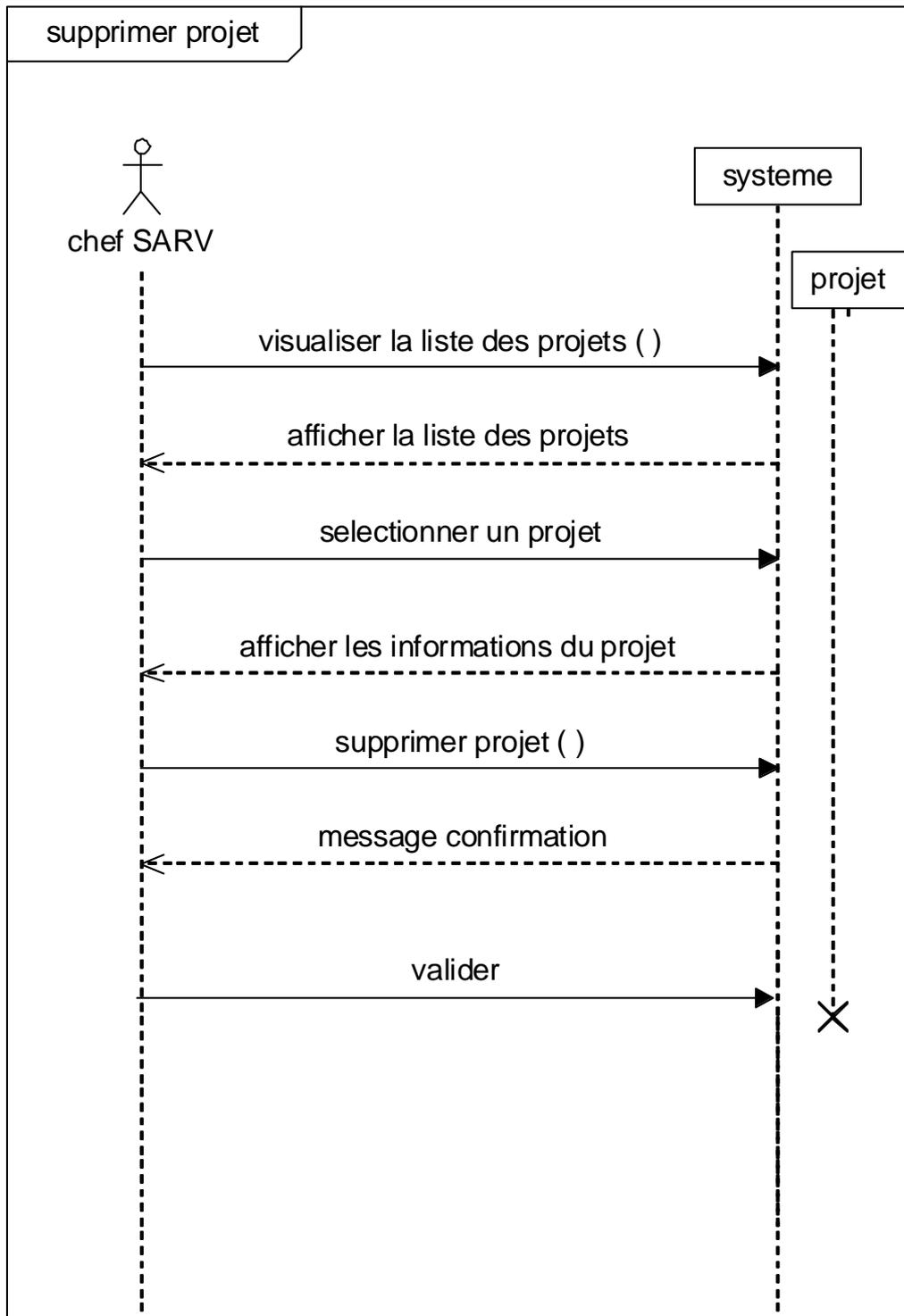


Figure 3.6 : Diagramme de séquence du cas d'utilisation supprimer projet.

### 3.5.5 Mis à jour chercheur

#### 3.5.5.1 Modifie chercheur

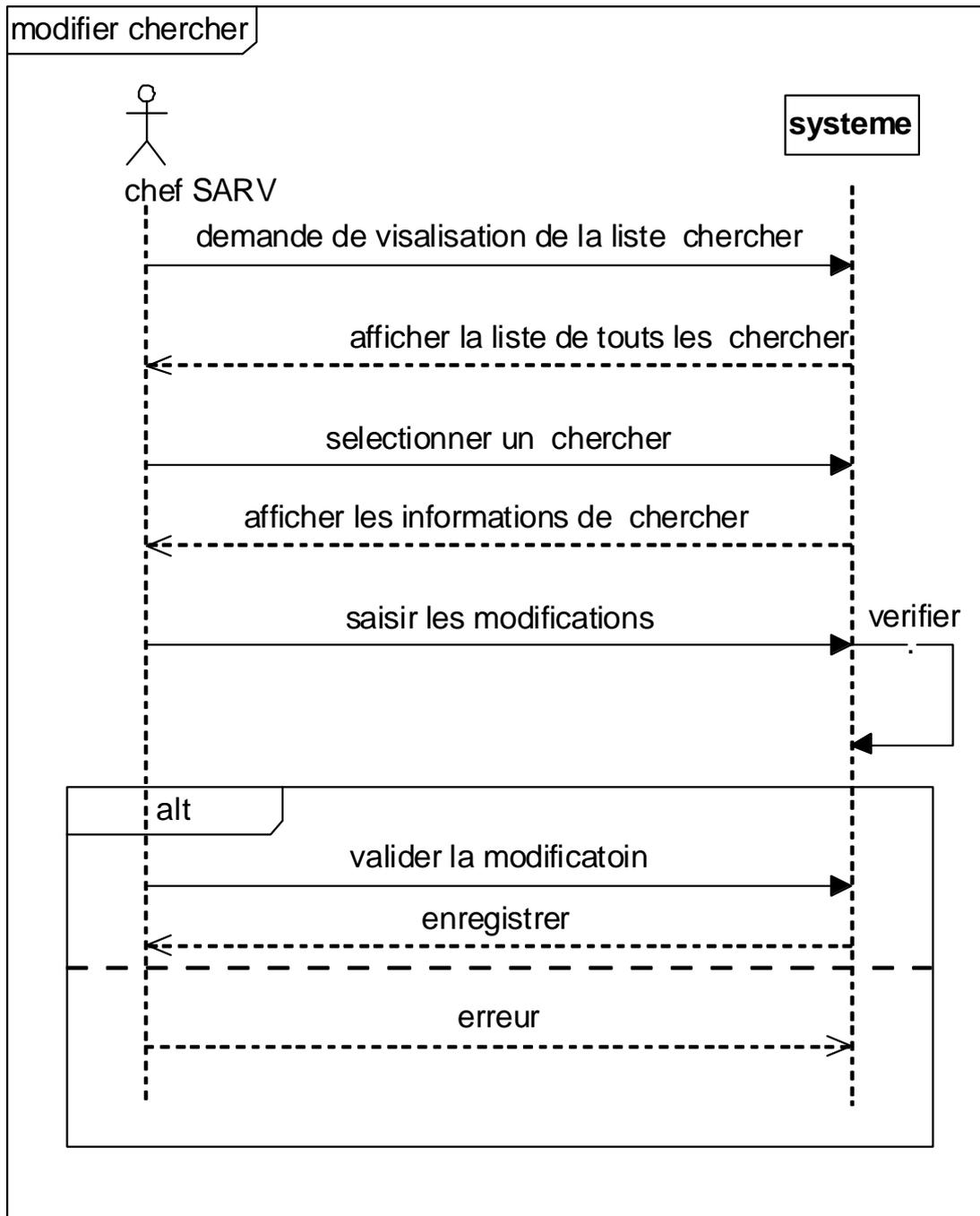


Figure 3.7: Diagramme de séquence du cas d'utilisation modifier chercheur.

### 3.5.5.2 Supprimer chercheur

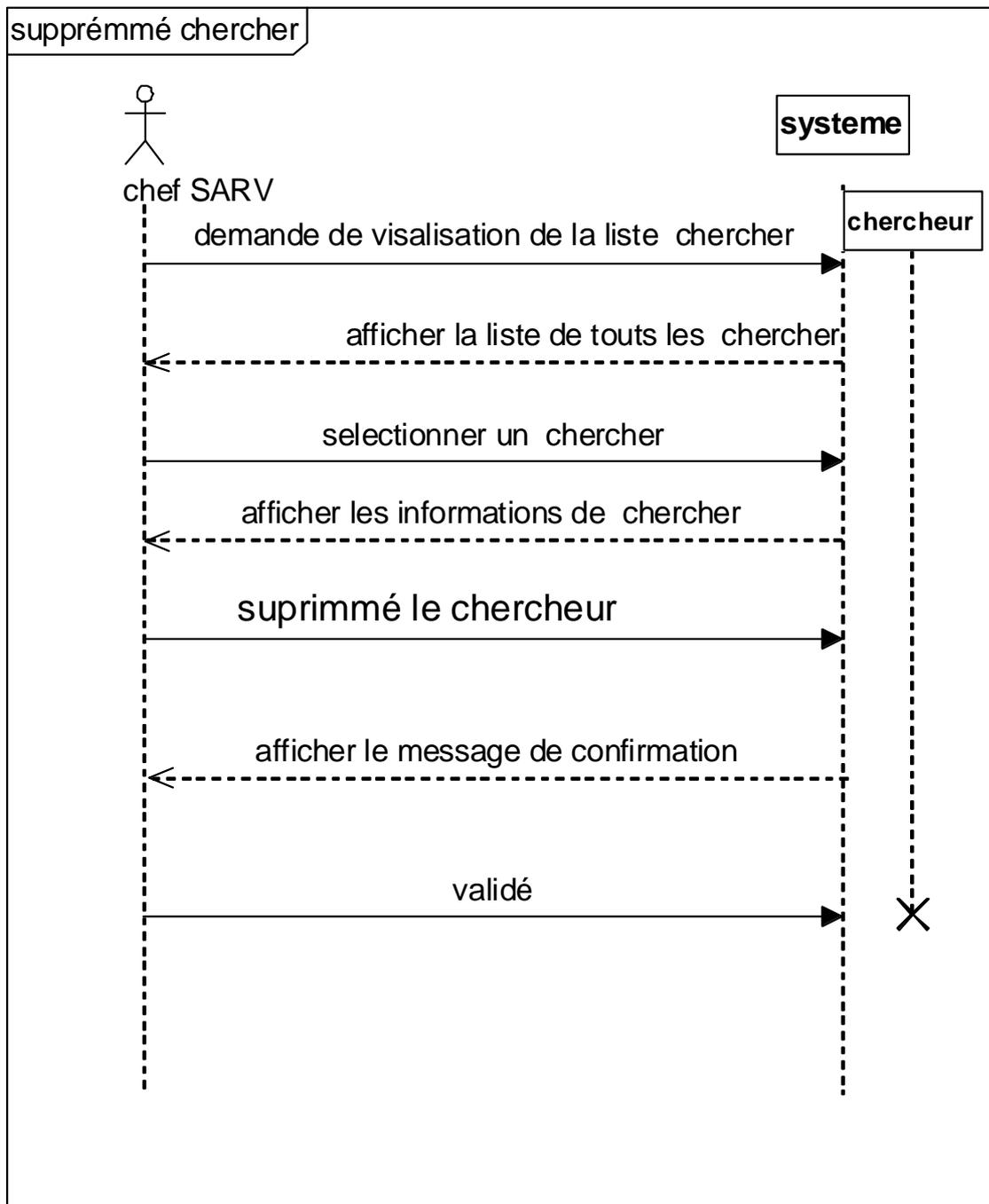


Figure 3.8 : Diagramme de séquence du cas d'utilisation supprimer chercheur.

3.5.6 réception des états d'avancements

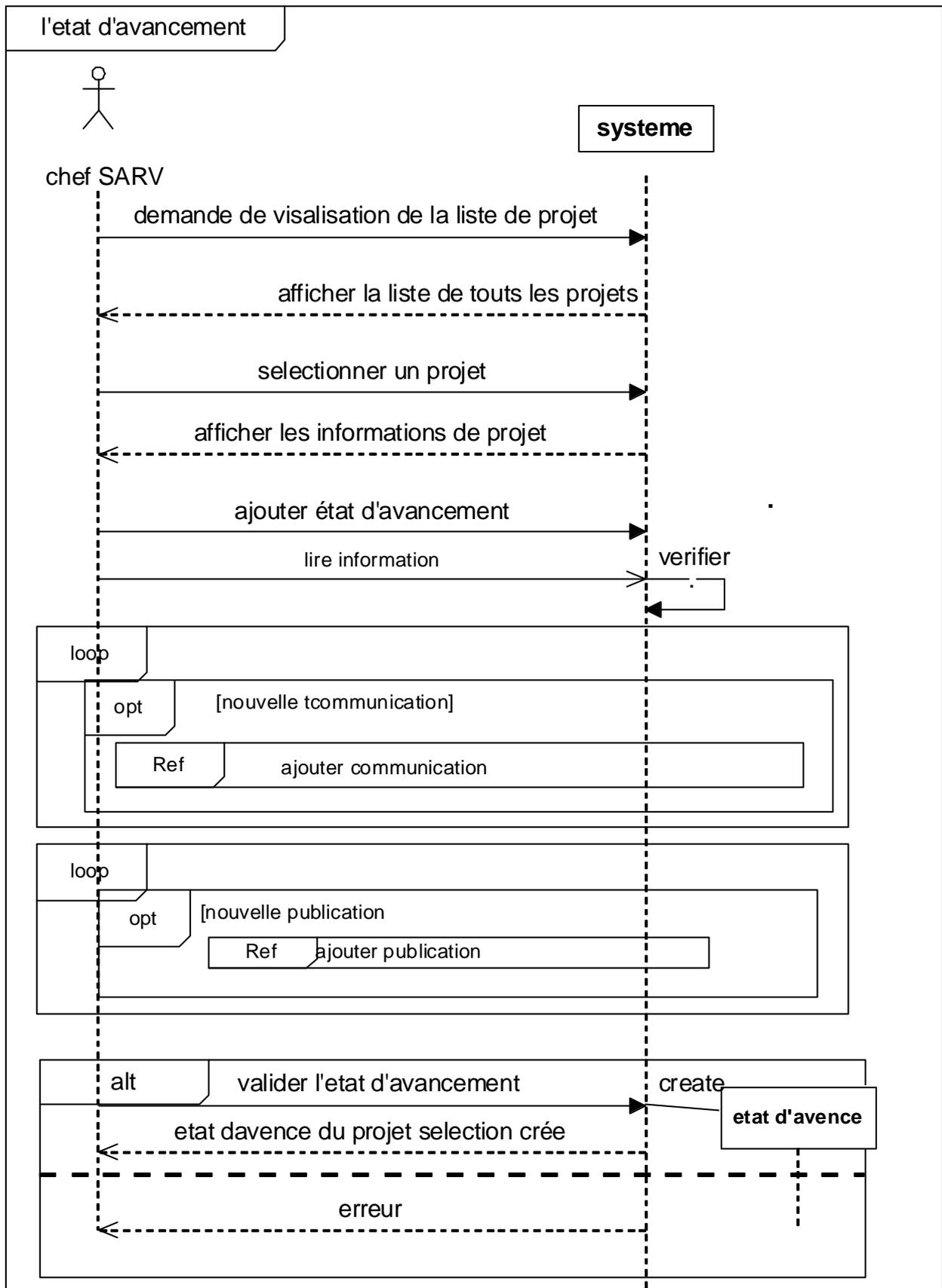


Figure 3.9 : Diagramme de séquence du cas d'utilisation état d'avancement.

### 3.5.7 Prolongation des projets

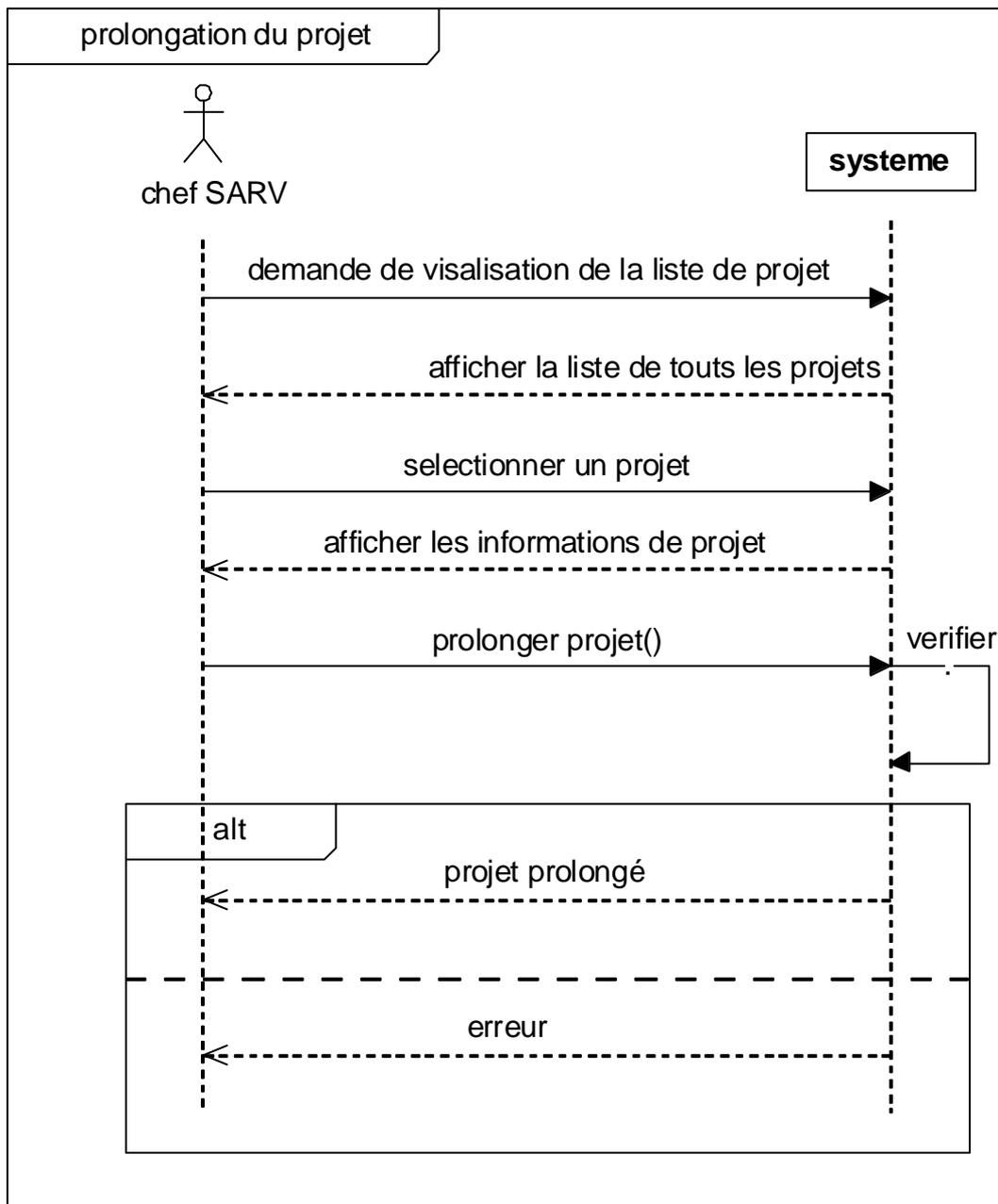


Figure 3.10 : Diagramme de séquence du cas d'utilisation prolongation.

3.5 .8 évaluation des états d'avancement

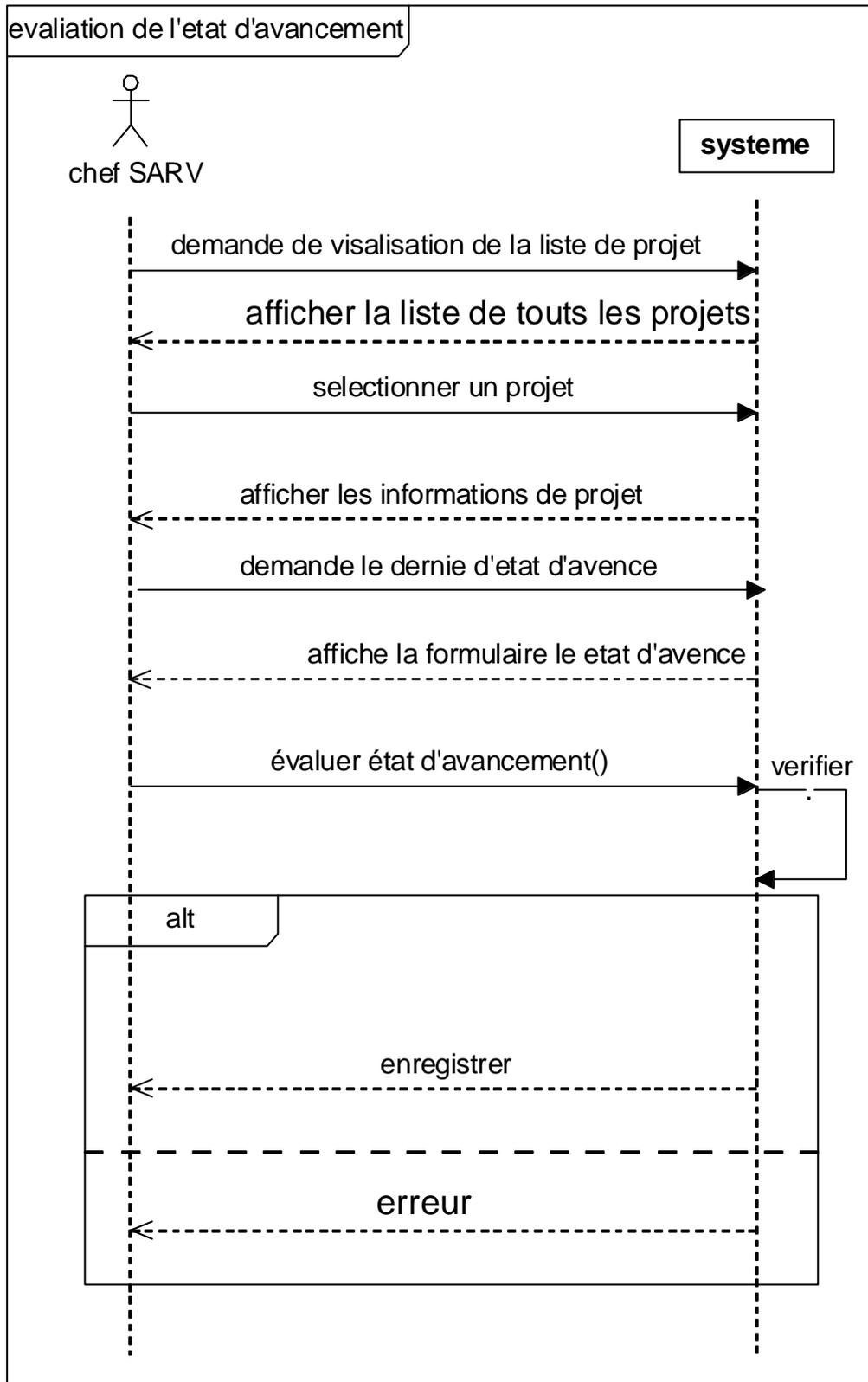


Figure 3.11 : Diagramme de séquence du cas d'utilisation évaluation des états d'avancement.

### 3.6 Le diagramme de classe

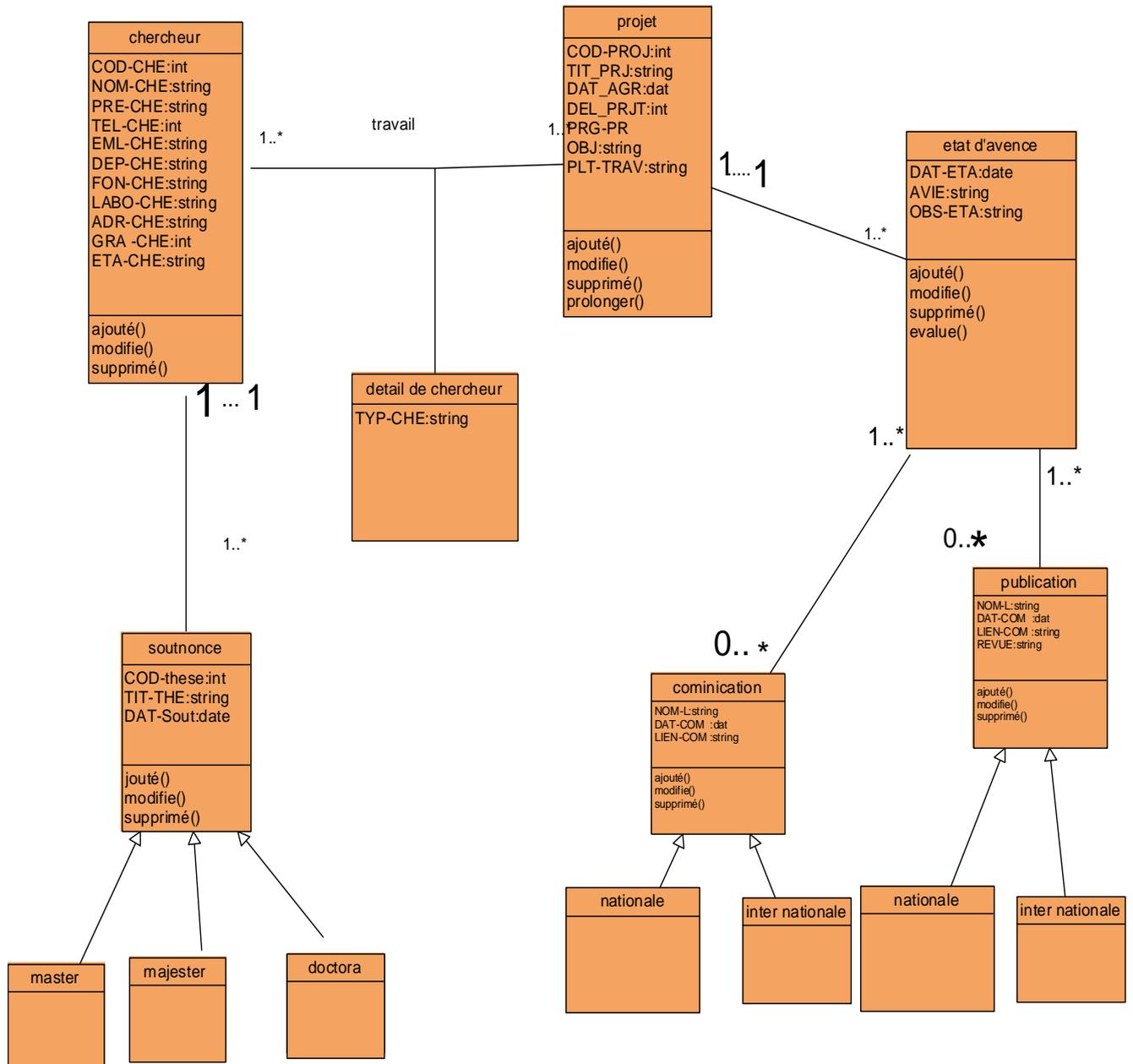


Figure 3.12 : Diagramme de classe

### **3.7 Conclusion**

La phase d'analyse que nous avons réalisée dans ce chapitre nous a permis de bien comprendre fonctionnement du système et ses différentes fonctionnalités.

Il faut noter que cette étape a été mise à jour tout au long de sa réalisation, car nous avons suivi une démarche itérative et incrémentale.

Nous abordons dans le chapitre suivant la phase de réalisation qui concrétisera tout ce qui a été présenté jusque-là.

# Chapitre 04

***Implementation***

## 4.1 Introduction

Après la réalisation de l'étape d'analyse, il faut passer à la dernière phase de la démarche utilisée à savoir la phase d'implémentation. Dans ce chapitre nous présentons d'abord les outils utilisés pour le développement de l'application. Par la suite, nous allons préparer les données nécessaires au logiciel, en convertissant les classes entités en une base de données. Enfin, nous décrivons quelques interfaces du logiciel réalisé.

## 4.2 Les outils de développement

### 4.2.1 L'environnement de développement Delphi

Delphi est un système de développement visuel rapide sous Windows (Rapid Application Développement) qui permet de créer des applications fenêtrées directement exécutables (.EXE) et redistribuables librement sous Windows ou DOS.

Delphi utilise le langage Pascal Orienté Objet (il est toutefois possible d'utiliser d'anciennes sources en Pascal standard grâce au compilateur en ligne de commande). Ce langage est facile à apprendre et beaucoup plus simple que le C++ traditionnel. Les objets utilisés ont des propriétés et des méthodes. Les propriétés sont les caractéristiques de l'objet (couleur, taille, ...) tandis que les méthodes sont les procédures (classiques ou événementielles) et fonctions qui y sont rattachées.

Delphi a beaucoup de caractéristiques qui les justifient comme choix pour la réalisation de notre application :

- C'est est un environnement de développement de type RAD (*Rapide Application Développement*) basé sur le langage Pascal. Il permet de réaliser rapidement et simplement des applications Windows.
- Il propose un ensemble très complet de *composants visuels* prêts à l'emploi incluant la quasi-totalité des composants Windows (boutons, boîtes de dialogue, menus, barres d'outils...) permettant de créer facilement divers types d'applications et de bibliothèques.
- Delphi est un outil moderne, qui fait appel à une conception objet.
- Il prend en charge le maintien automatique d'une partie du code source.
- Il permet de créer facilement de nouveaux composants qui peuvent être intégrés dans la palette des composants déjà existants.
- Son compilateur intégré permet une application rapide et efficace car les erreurs éventuelles du code sont immédiatement détectées. L'utilisateur est alors informé précisément des erreurs de son programme.

- Il permet également d'utiliser des formats images, textes, sons, grâce à certains Composants.
- Delphi n'est pas lié à un format de données spécifiques. Il peut en effet utiliser des tables, DBASE, Access, ou Paradox et accéder à des bases de données SQL serveur à travers Sun ODBC (Open Data Base Connectivity).

#### 4.2.2 Implémentation de la base de données sous Access

Avec Microsoft Office Access 2007, les travailleurs de l'information peuvent, en un clin d'œil et en toute simplicité, contrôler les informations dont ils disposent et créer des états à partir de celles-ci grâce à l'interface utilisateur Microsoft Office Fluent et aux fonctionnalités de conception interactives qui n'exigent aucune connaissance approfondie en matière de base de données.

##### ❖ Les avantages d'Access

- Obtenir de meilleurs résultats plus vite grâce à l'interface utilisateur office fluent,
- Démarrer en un clin d'œil à l'aide de solutions prédéfinies ;
- Créer plusieurs états avec différentes vues des mêmes informations ;
- Créer des tables rapidement sans se soucier de la complexité des bases de données ;
- Bénéficier de nouveaux types de champs dans des scénarios encore plus riches ;
- Accéder et exploiter des informations issues de plusieurs sources.

### 4.3 Le passage du diagramme de classe au modèle relationnel

#### 4.3.1 Les règles de passage

Les classes entités et leurs associations seront convertit à une base de données Relationnelle, qui sera sollicitée par l'application pour consultation et mise à jour. Pour ce faire nous sommes basés sur les règles suivantes :

- **Transformation des entités/ classes** : La règle est simple

##### **R1 :**

- Chaque classe devient une relation, l'identifiant de l'entité devient clé primaire pour la relation.
- chaque classe du diagramme UML devient une relation. Il faut choisir un attribut de la classe pouvant jouer le rôle identifiant.

- **Transformation des associations** : Les règles de transformation que nous allons voir dépendent des cardinalités multiplicités maximale des associations. Nous distinguons trois familles d'association :

- **Association 1..\*** : La règle est la suivante :

**R2** : il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.

- **Association \*.\*** : La règle est la suivante :

**R3** : association/classe – association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des identités connecté à l'association. Chaque attribut devient clé étrangère si entité/classe connectée dont il devient une relation en vertu de la règle R1. Les attributs d'association/classe – association doivent être ajoutés a la nouvelle relation.ces attributs ne sont ni clé primaire, ni clé étrangère

- **Association 1..1** : La règle est la suivante :

**R4** : il faut ajouter un attribut de type clé étrangère dans la relation dérivée de l'entité ayant la cardinalité minimale égale à zéro. Dans le cas de diagramme UML il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un.

L'attribut porte le nom de la clé primaire de la relation dérivée d'entité classe connectée à l'association. Si les deux cardinalités minimales égales à zéro, le choix est donné entre les deux relations dérivées de la **R1**.

Si les deux cardinalités minimales égales à un, il est préférable de fusionner les deux entités/classe en une seule [2]

### 4.3.2 La base de données

Après application des quatre règles de passage précédentes, nous avons obtenus les tables suivantes :

**Projet** (cod\_prjt,tem\_prjt,dat\_agr,,deli\_prjt ,plt\_trav,obj\_prjt)

**Chercheur**(cod\_cher,nom\_cher,pren\_cher,tel\_cher,eml\_cher,grad\_cher,dep\_cher,fon\_cher, adr\_cher,eta\_cher,nom\_labo)

**These** (cod\_these,tit\_the,dat\_set,typ\_these)

**Etat\_avancement** (code\_état,dat\_ave,avie,ops-eta)

**Detail\_communication** (#code\_état,# code\_comm)

**communication** (code\_comm,nom\_com,dat\_com,typ\_com,lien\_com)

**detail\_chercheur**(#cod\_prj,#cod\_cher,typ\_cher)

**Detail\_publication**(#code\_état,# code\_pub)

**publication** (code\_pub,nom\_pub,dat\_pub,typ\_pub,lien,rev\_pub)

## 4.4 Gestion de contrôle et de sécurité

### 4.4.1 Contrôle

L'erreur peut naitre en de nombreux points du circuit de l'information, elle peut être provoquée par un facteur humain ou par une défaillance du matériel.

Il est important de détecter et rectifier l'erreur dès son apparition, car une information non correcte risque de rendre incorrectes de nombreuses autres informations. Par conséquent, différents contrôles s'imposent :

- Vérifier les informations à saisir ;
- Contrôler le type de l'information ;
- Contrôler la présence de l'information ;
- Utiliser les messages d'aides ;
- Des contrôles sont réalisés grâce à des relations de comparaisons.

### 4.4.2 Sécurité

Le concept de sécurité en informatique est très large, il s'agit de la sécurité de tous les dangers qui menacent tous les systèmes d'informations. Pour assurer le bon fonctionnement du système mis en place et pour éviter les destructions dues imprévues, il est indispensable de prévoir des mesures de sécurité pour protéger l'ensemble des informations contre toutes actions frauduleuses.

Pour cela nous avons distingué plusieurs mesures de sécurité :

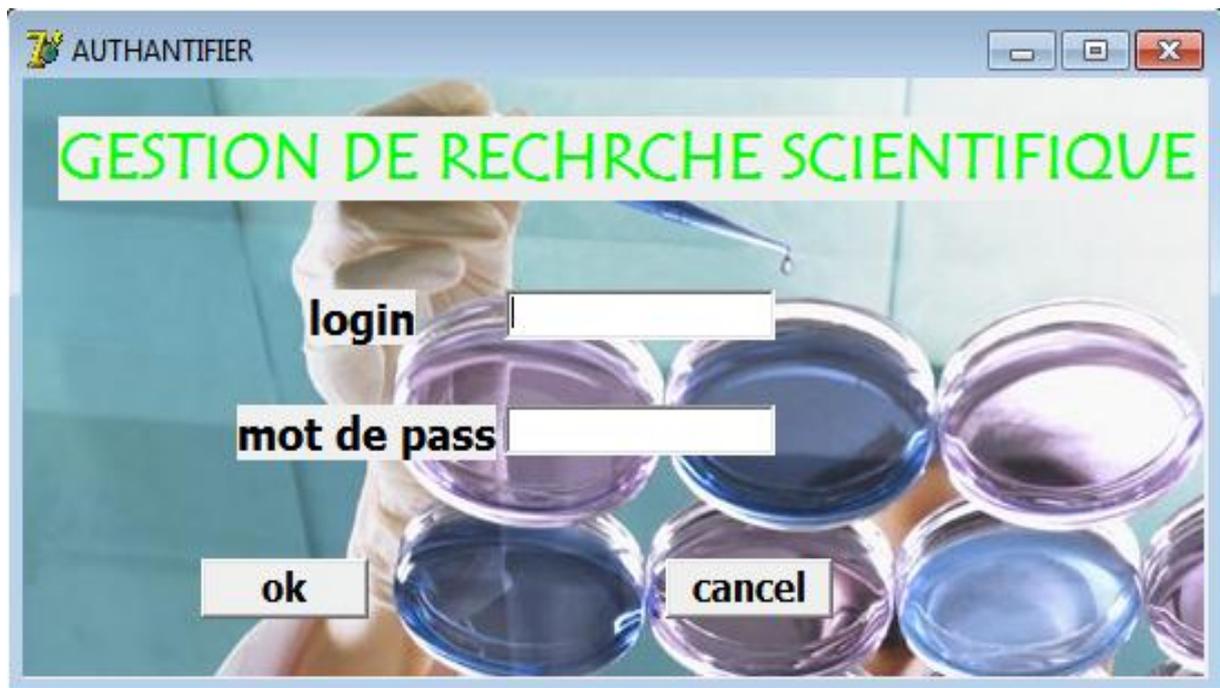
- Il faut sauvegarder l'application sur des supports fiables.
- Utiliser les mots de passe.
- Utiliser les anti-virus.

## 4.5 Les interfaces de l'application

Dans ce qui suit, nous allons présenter quelques interfaces de notre application de gestion des projets de recherche.

### 4.5.1 Interface d'authentification

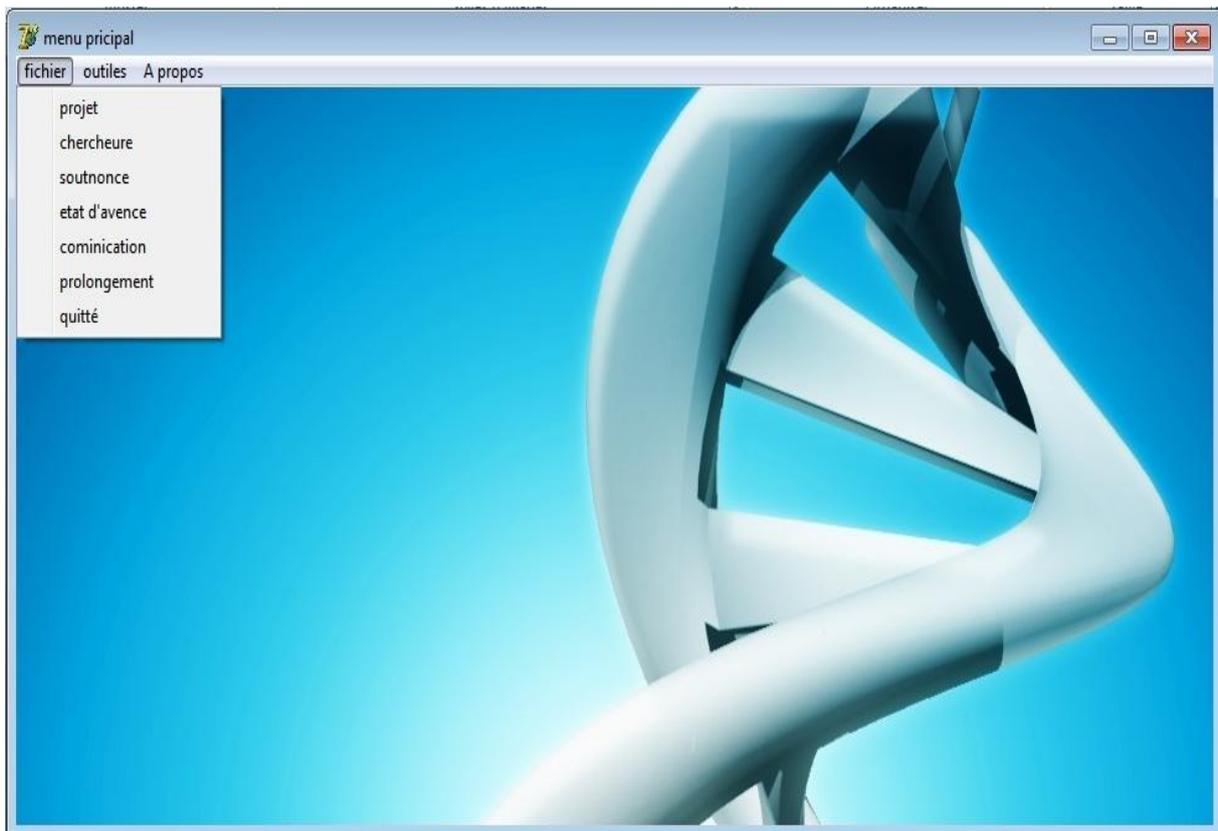
L'interface d'authentification permet utilisateur d'accéder à sa session par le biais de son login et son mot de passe.



**Figure 4.1** Interface authentification.

### 4.5.2 Interface du menu principale (recherche scientifique)

Cette interface permet d'avoir une vue globale sur l'application et l'accès à toutes ses fonctionnalités.



**Figure 4.2** Interface menu principale (Gestion de projet).

### 4.5.3 Interface projet

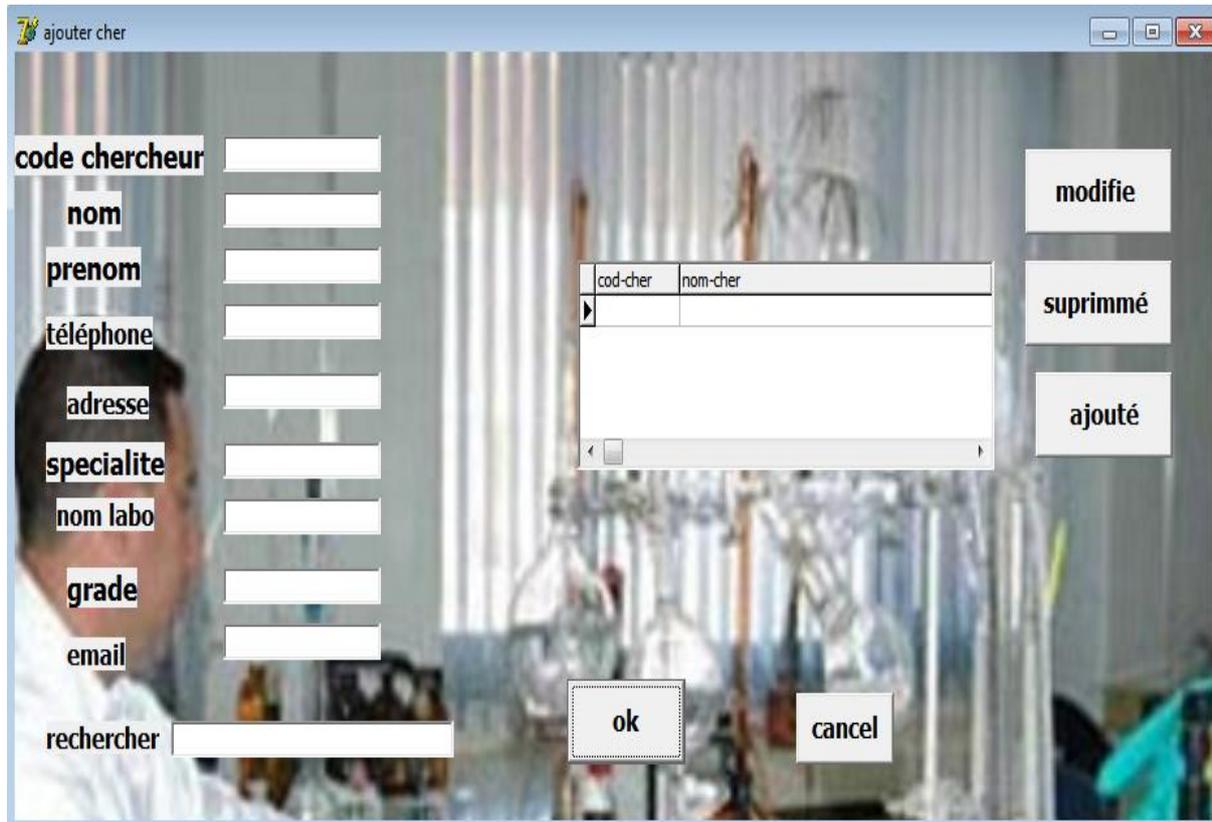
Cette interface permet d'ajouter, supprimer et modifier les informations d'un projet. Et elle permet de visualiser la liste de tous les projets existants.

cod-prjt	tem-prjt	dat-deb	dat-fin	del-prjt	cod-cher

**Figure 4.3** Interface ajouté projet.

#### 4.5.4 Interface chercheur

Cette interface permet d'ajouter, supprimer et modifier un chercheur. Ainsi, elle permet de visualiser la liste de tous les chercheurs existants.



**Figure 4.4** Interface chercheur.

#### 4.5.5 Interface thèse

Cette interface permet d'ajouter, supprimer et modifier les informations de la thèse. Elle permet aussi de visualiser la liste de tous les mémoires et les thèses encadrés et/ou Co-encadrés par les chercheurs.

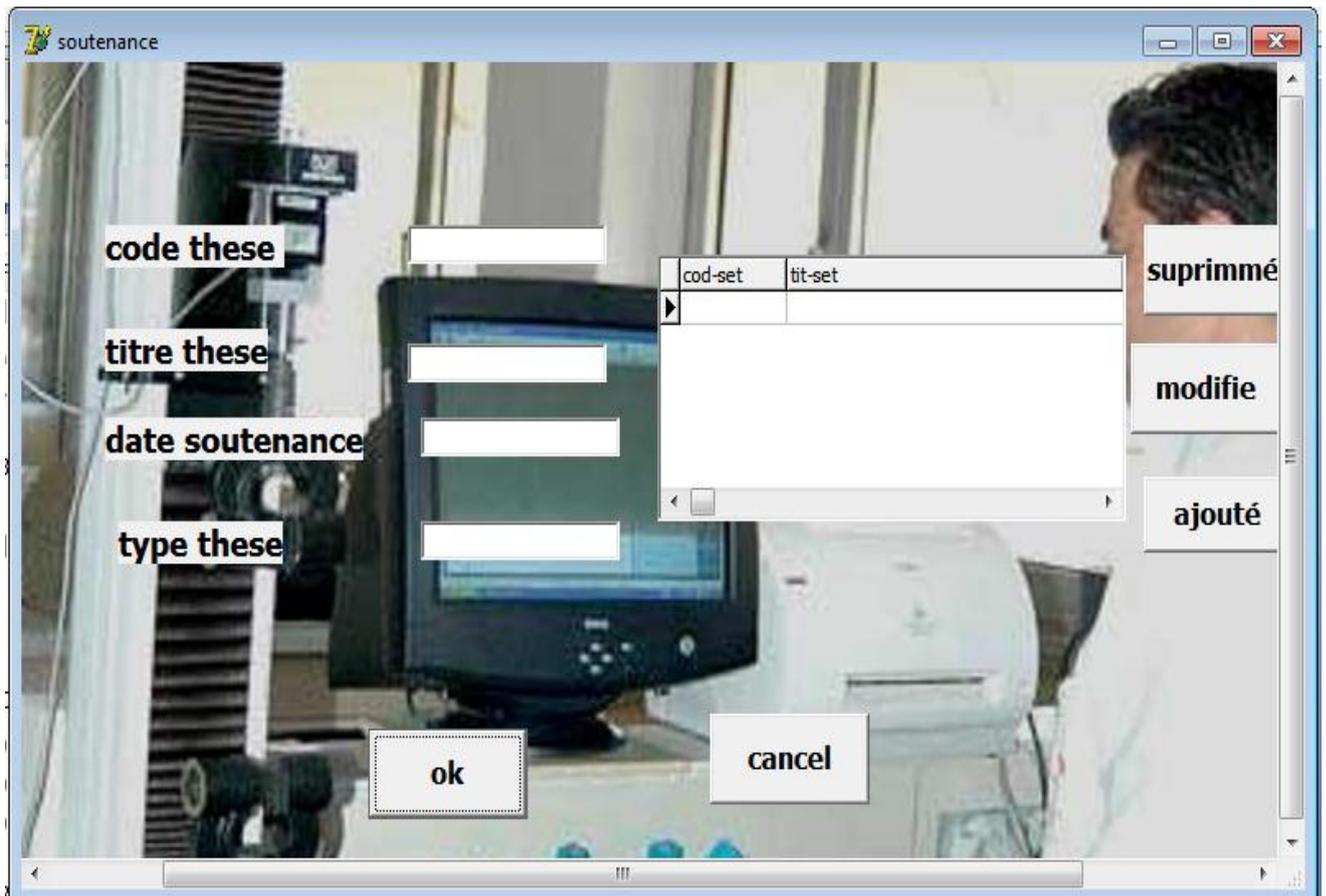
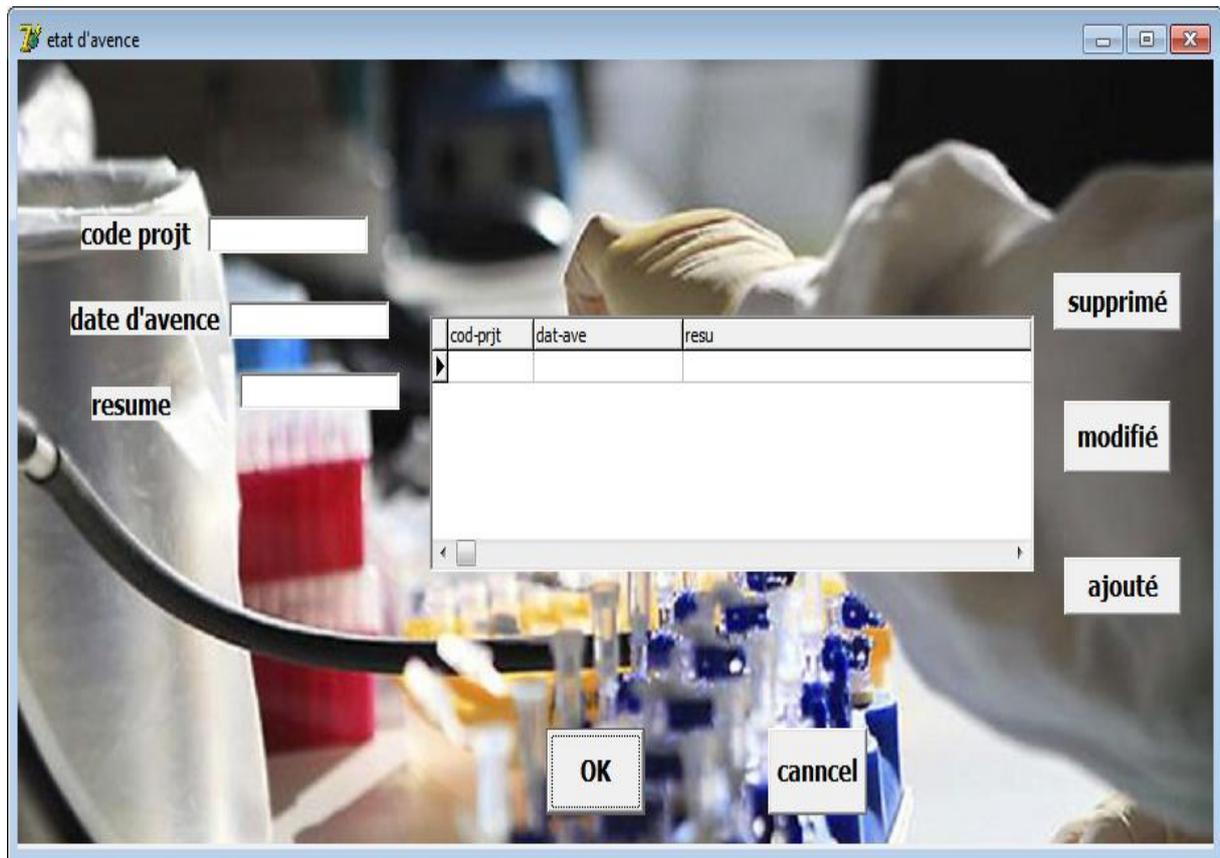


Figure 4.5 Interface de la soutenance.

### 4.5.6 Interface l'état d'avance

Cette interface permet d'ajouter, supprimer et modifier un état d'avance d'un projet.



**Figure 4.6** Interface l'état d'avance.

### 4.5.7 Interface communication

Cette interface permet d'ajouter, supprimer les informations de communication d'un projet.



## 4.6 Conclusion

Dans ce chapitre, nous avons décrit les outils utilisés dans la phase d'implémentation ainsi qu'une brève description des interfaces de l'application que nous avons réalisée afin de faciliter son utilisation.

# *Conclusion générale*

Notre projet sur lequel nous avons travaillé est la conception et le développement d'une application de gestion des projets de recherche au niveau du centre universitaire de MILA. Nous avons effectué un stage pour obtenir beaucoup d'information sur les projets de recherche. Par la suite, nous avons utilisé l'UML pour modéliser l'application et l'environnement Delphi dans la programmation.

Ce travail nous a donné un avant-goût du métier de développeur et il nous a permis de concevoir pour la première fois une vraie application pour un vrai client et cela constitue une grande satisfaction personnelle et professionnelle qui marque le début d'aboutissement de notre formation.

# *Références bibliographiques*

- [1] L. AUDIBERT/ UML 2/2007-2008.
- [2] <http://www.acces.fr.com>.
- [3] formation multimédia à la programmation Delphi: Cours, applications et Travaux Pratiques ; Mc BELAID