

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



N° Réf :.....

**Centre Universitaire de Mila**

**Institut des sciences et de la technologie**

**Département de Mathématiques et Informatique**

**Conception et réalisation d'une application de gestion  
de la paie-service de budget  
Centre universitaire de Mila**

**Mémoire préparé En vue de l'obtention du diplôme de License  
En informatique**

**Préparé par : Bilfan chouki  
Guerbaa Moussaab**

**Encadré par : Mr Bourideh Adel**

**Filière : Mathématiques et Informatique**

**Spécialité : Informatique**

**Année universitaire : 2012/2013**

## *Remerciements*

*Au terme de cette étude nous remercions avant tout DIEU de nous avoir accordé la santé et les moyens de réaliser ce travail.*

*Nous remercions vivement Mr BOURIDEH ADEL, le directeur de ce travail pour nous avoir guidés et conseiller pour la réalisation de ce mémoire.*

*Nous tenons à remercier Mr BOUDEMIJA SAMIR le chef service de la comptabilité.*

*Nous tenons à remercier Mr BOUMALI FOUJIL, TAHER, MANEL pour accepter de juger ce travail.*

*A tous ceux qui nous ont aidés de près ou de loin à la réalisation de ce modeste travail.*

*Merci a tout*

## *Dédicaces*

*Je dédie ce modeste travail,  
À Ma chère Mère DRIFA, pour ses sacrifices depuis  
Qu'elle m'a mis au monde, Mon Père YOUCEF, qui m'a  
Toujours soutenu et aidé à affronter les  
Difficultés,*

*À Tous mes Frères et sœurs,  
À Tous ma famille  
A mon binôme Guerbaa Mousaab,  
À Tous mes amis Badis, Tourki, Abderrahim,  
Mehdi, Ammar et mes collègues de la  
Promotion 2013,*

*À Tous ceux qui me sont chers,*

*Chouki*

*Dédicaces*

*Je dédie ce modeste  
travail,*

*À Ma chère Mère*

*NOURA*

*Pour son amour et  
son encouragement.*

*MOUSSAAB*

# Table de matière

## Chapitre1 : Langage de modélisation

Introduction .....	2
1. concepts UML .....	2
1.1. Historique des méthodes des modélisations.....	2
1.2. Quelques dates .....	3
1.3. Pourquoi UML .....	4
1.4. Avantages et inconvénients d'UML.....	4
2. modélisation avec UML.....	6
2.1. C'est pourquoi un modèle.....	6
2.2. Pourquoi modéliser .....	7
3. mise en œuvre d'UML.....	8
3.1. UML n'est pas une méthode.....	8
3.1.1. Une méthode simple et générique .....	9
3.2 Identification des besoins et spécification des fonctionnalités.....	10
3.2.1 Identification et représentation des besoins : diagramme de cas d'utilisation.....	11
3.2.2 Maquette de l'IHM de l'application (non couvert par UML).....	12
3.3. Phases d'analyse).....	13

3.3.1. Analyse du domaine : modèle du domaine.....	14
3.3.2. Diagramme de classe participante.....	16
3.4. Phase de conception.....	20
3.4.1. Diagramme de séquence.....	20
3.4.2. Diagramme de classes de conception.....	22

## Chapitre2 : Identification des besoins

Introduction .....	26
1. Cahier de charge .....	26
1.1. Présentation de projet.....	26
2. Dictionnaire de donnés .....	27
3. diagramme de cas d'utilisation .....	30
3.1. Notre diagramme d'utilisation.....	31
4. description de cas d'utilisation .....	32
4.1. S'authentifier .....	32
4.2. Gestion comptabilité.....	33
4.3. Gestion budget.....	33
4.4. Initialisation .....	34
4.5. Gestion salarié.....	35
4.6. Gestion de la paie.....	36
4.7. Gestion impression budget.....	37
4.8. Gestion impression comptabilité.....	38
4.9. Ajouter salarié.....	39

4.10. Modifier salarié.....	40
4.11. Supprimé salarié.....	41
4.12. Consulter salarié.....	42
4.13. Rechercher salarié.....	43
4.14. Calcule de la paie.....	44
4.15. Imprimer état matrice.....	45
4.16. Imprimer la fiche d'engagement .....	46
4.17. Imprimer la fiche de paie .....	47
4.18. Imprimer l'ATS .....	48
4.19. Imprimer avis de virement.....	49
4.20. Imprimer ordre de virement .....	50
5. Diagramme de séquence système .....	51
4.1. Authentifier .....	52
5.2. Gestion budget.....	53
5.3. Gestion comptabilité.....	54
5.4. Salarié.....	55
5.5. Paie.....	56
5.6. Impression budget.....	57
5.7. Impression comptabilité.....	58
5.8. Initialisation .....	59
5.9. Ajouter salarié.....	60
5.10. Supprimé salarié.....	61
5.11. Modifier salarié.....	62
5.12. Rechercher salarié.....	63
5.13. Consulter salarié.....	64

5.14. Calcule de la paie.....	65
5.15. Imprimer état matrice.....	66
5.16. Imprimer la fiche d'engagement .....	67
5.17. Imprimer la fiche de paie .....	68
5.18. Imprimer l'ATS .....	69
5.19. Imprimer avis de virement.....	70
5.20. Imprimer ordre de virement .....	71

### Chapitre3 : Phase d'analyse

1. analyse du domaine .....	72
1.1. Affectation des attributs aux classes.....	73
2. Modèle de domaine.....	74
3. Diagramme de classe participante.....	75
3.1. S'authentifier .....	76
3.2. Gestion salarié.....	77
3.3. Ajouter salarié.....	78
3.4. Modifier salarié.....	79
3.5. Supprimer salarié.....	79
3.6. Consulter salarié.....	80
3.7. Gestion grille indiciaire.....	81
3.8. Initialiser grille indiciaire.....	81
3.9. Modifier grille indiciaire.....	82
3.10. Gestion comptabilité.....	83
3.11. Imprimer fiche de paie.....	84

3.12. Imprimer ATS.....	84
3.13. Imprimer avis de virement.....	85
3.14. Imprimer ordre de virement.....	85
3.15. Gestion budget.....	86
3.16. Imprimer état matrice.....	86
3.17. Imprimer fiche d'engagement.....	87
3.18. Gestion de la paie.....	87

#### Chapitre4 : Phase de conception

1. Diagramme de séquence.....	89
1.2. Authentifier .....	90
1.3. Gestion comptabilité.....	91
1.4. Gestion budget.....	92
1.5. Gestion grille indiciaire.....	93
1.6. Gestion de la paie.....	94
1.7. Gestion salarié.....	95
1.8. Initialiser grille indiciaire.....	96
1.9. Modifier grille indiciaire.....	97
1.10. Ajouter salarié.....	98
1.11. Modifier salarié.....	99
1.12. Supprimer salarié.....	100
1.13. Consulter salarié.....	101
1.14. Calcule de la paie.....	102

1.15. Imprimer fiche de paie.....	103
1.16. Imprimer ATS.....	104
1.17. Imprimer avis de virement.....	105
1.18.. Imprimer ordre de virement.....	106
1.19. Imprimer état matrice.....	107
1.20. Imprimer fiche d'engagement.....	108
2. Diagramme de classes de conception .....	109
2.1. Authentifier .....	110
2.2. Comptabilité.....	111
2.3. Budget .....	112
2.4. Paie.....	113
2.5. Grille.....	114
2.6. Salarié.....	115
2.7. Imprimer fiche de paie.....	116
2.8. Imprimer ATS.....	117
2.9. Imprimer ordre de virement.....	118
2.10. Imprimer avis de virement.....	119
2.11. Imprimer état matrice.....	120
2.12. Fiche d'engagement.....	121
2.13. Ajouter salarié.....	122
2.14. Modifier salarié.....	123

2.15. Supprimer salarié.....	124
2.16. Consulter salarié.....	125
2.17. Initialiser grille indiciaire.....	126
2.18. Modifier grille indiciaire.....	127
2.19. Calcule de la paie.....	128
3. Diagramme de class.....	129

### Chapitre5 : Implémentation

Introduction .....	131
1. représentation de notre base de donnés .....	131
2. le langage de programmation DELPHI 7.....	134
2.1. Définition .....	134
2.2. Les avantage de Delphi 7.....	134
3. la base de données ACCES 2010.....	135
3.1. Avantage accès 2010.....	135
4. l'interface graphique .....	136
4.1. Fenêtre authentification .....	136
4.2. Fenêtre gestion de la paie.....	137
4.3. Fenêtre ajouté salarié.....	138
4.4. Fenêtre modifié salarié.....	139
4.5. Fenêtre supprimé salarié.....	140
4.6. Fenêtre consulté salarié.....	141

4.7. Fenêtre payé salarié.....	142
4.8. Fenêtre imprimer fiche de paie.....	143
4.9. Fenêtre imprimer avis de virement.....	144
4.10. Imprimer ordre de virement.....	145
Conclusion .....	146
Référence bibliographie.....	147

*Chapitre 01 : langage de  
modélisation*

*Introduction*

*1. concepts UML*

*2. modélisation avec UML*

*3. mise en œuvre d'UML*

# Introduction

## UML (Unified Modeling Language)

Se définit comme un langage de modélisation graphique qui permet la spécification, la représentation et la construction des composantes d'un système informatique.

UML, que l'on peut traduire par "langage de modélisation unifié" est une notation permettant de modéliser un problème de façon standard. UML est né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 90 : OMT, Booch et OOSE. Issu du "terrain" et fruit d'un travail d'experts reconnus, UML est le résultat d'un large consensus. De très nombreux acteurs industriels de renom ont adopté UML et participent à son développement. En Fin 1997, UML est devenu une norme OMG (Object Management Group).

## 1. Concepts UML

### 1.1 Historique

- Les années 1980: Utilisation de méthodes adaptées à la programmation impérative (notamment Merise)

- Fin 80 / début 90 la programmation par objets prend de l'importance.
- Conséquence naturelle: mise en place de méthodes orientées objet. Plus de cinquante méthodes apparaissent entre 1990 et 1995:
  - OOD : Object Oriented Design (Booch, 1991)
  - HOOD: Hierarchical Object Oriented Design (Delatteet 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)
- Les méthodes proposées définissent des concepts assez proches avec des notations différentes.
- 1994:le consensus se fait autour de trois méthodes:
  - OMT (Object Modeling Technique) de James Rumbaugh fournit une représentation graphique des aspects statique, dynamique et fonctionnel d'un système;
  - OOD (Object Oriented Design) de Grady Booch, définie pour le Département of Défense, introduit le concept de paquetage (package);
  - OOSE (Object Oriented Software Engineering) d'Ivar Jacobson fonde l'analyse sur la description des besoins des utilisateurs (cas d'utilisation, ou use cases).
- 1995: Fusion des 3 principales méthodes pour définir un langage de modélisation commun : UML (Unified Modeling Language).

## **1.2. Quelques dates**

- 1995 : Méthode unifiée 0.8 (intégrant les méthodes OOD et OMT)
- 1995 : UML 0.9 (intégrant la méthode OOSE)
- 1996 : UML 1.0
- 1997 : UML 1.1
- 1998 : UML 1.2
- 1999 : UML 1.3
- 2000 : UML 1.4
- 2001 : UML 1.4.2
- 2003 : UML 1.5
- 2004 : UML 2.0 (Evolution importante)
- 2007 : UML 2.1

### **1.3. Pourquoi UML ?**

#### **UML pour :**

- Obtenir une modélisation de très haut niveau indépendante des langages et des environnements.
- Faire collaborer des participants de tous horizons autour d'un même document de synthèse.
- Faire des simulations avant de construire un système.
- Exprimer dans un seul modèle tous les aspects statiques, dynamiques, juridiques, spécifications, etc...
- Documenter un projet.
- Générer automatiquement la partie logiciel d'un système.

### **1.4 Avantages et inconvénients d'UML**

### **1.4.1 Les points forts d'UML**

- UML est un langage formel et normalisé, il permet un gain de précision et un
- gain de stabilité. Ce qui encourage l'utilisation d'outils.
- UML est un support de communication performant, il cadre l'analyse et il
- facilite la compréhension des représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel.

### **1.4.2 Les points faibles d'UML**

- La mise en pratique d'UML nécessite un apprentissage et passe par une
- période d'adaptation.
- UML n'est pas à l'origine des concepts objets, mais en constitue une étape
- majeure, car il unifie les différentes approches et en donne une définition plus
- formelle.
- Le processus (non couvert par UML) est une autre clé de la réussite d'un
- projet. Or, l'intégration d'UML dans un processus n'est pas triviale et
- améliorer un processus et une tâche complexe et longue.

### **1.5 Remarque**

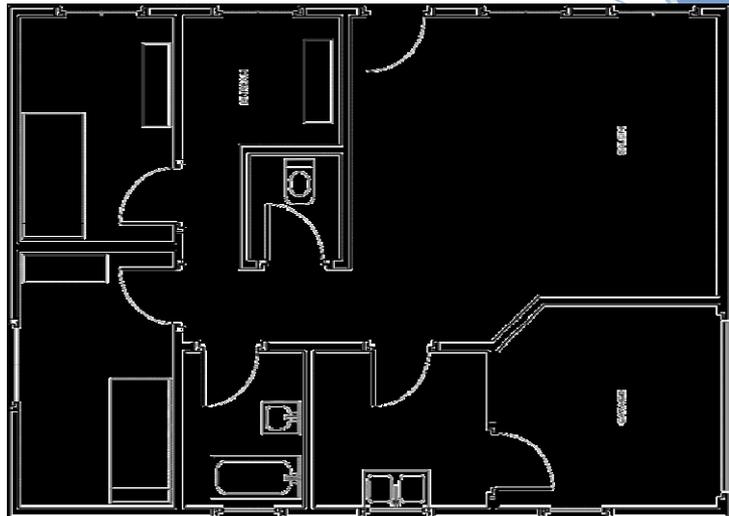
- UML est un *langage* et non une *méthode* dans la mesure où elle ne présente aucune *démarche* (*Méthode=langage+démarche*).
- La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.

## 2. Modélisation avec UML

### 2.1 C'est quoi un modèle ?

#### ❖ Exemple

- Pour la construction d'un immeuble, il faut préalablement élaborer de nombreux *plans*:
  - plans généraux du bâtiment et de sa structure ;
  - plans détaillées des différents locaux, bureaux, appartements, .
  - ..
  - plans des câblages électriques ;
  - plans d'écoulements des eaux, etc.
- Les plans sont des *modèles*



- Un modèle est une représentation abstraite et simplifiée (qui exclut certains détails), d'une entité (phénomène, processus, système, etc.) du monde réel en vue de le décrire, de l'expliquer ou de le prévoir.
- Un modèle permet de réduire la complexité d'un phénomène en éliminant les détails qui n'influencent pas son comportement de manière significative.
- Il reflète ce que le concepteur croit important pour la compréhension du phénomène modélisé.

## 2.2 Pourquoi modéliser ?

Modéliser un système avant sa réalisation permet de:

- **Faciliter la compréhension du système étudié:** Un modèle réduit la complexité du système étudié.
- **Faciliter la communication entre les membres de l'équipe:** Un modèle est un langage commun, précis, qui est connu par tous les membres de l'équipe et il est donc, à ce titre, un vecteur privilégié pour communiquer. Cette communication est essentielle pour aboutir à une compréhension commune entre différentes parties prenantes (notamment entre la maîtrise d'ouvrage et la maîtrise d'œuvre informatique).

- **Mieux répartir les tâches entre les membres de l'équipe.**
- **Réduire les coûts et les délais:** Par exemple, les plateformes de modélisation savent maintenant exploiter les modèles pour faire de la génération de code (au moins au niveau du squelette).
- **Faciliter la maintenance:** une fois mise en production, l'application va devoir être maintenue, probablement par une autre équipe qui n'est pas nécessairement celle ayant créée l'application.

### 3 Mise en œuvre d'UML

#### Introduction

#### 3.1 UML n'est pas une méthode

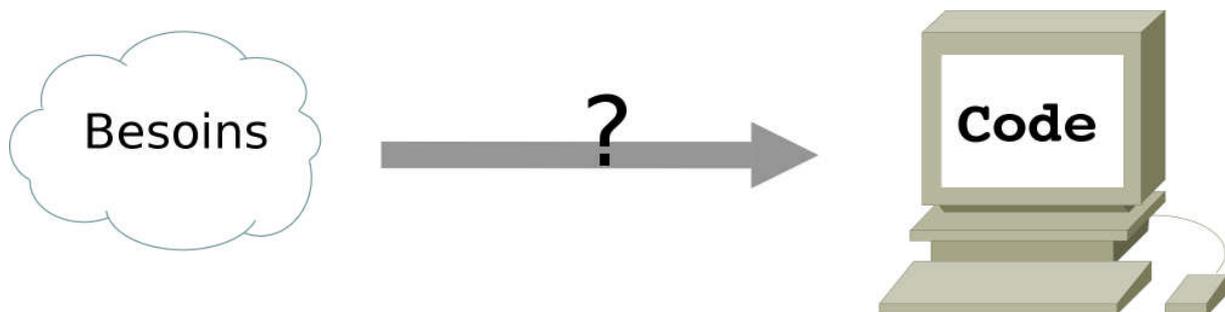


Figure1.1

- **Quelle méthode pour passer de l'expression des besoins au code de l'application ?**

La problématique que pose la mise en œuvre d'UML est simple : comment passer de l'expression des besoins au code de l'application ? Cette problématique est parfaitement illustrée par la figure 1.1.

Comme nous l'avons déjà dit, à maintes reprises, UML n'est qu'un langage de modélisation, ce n'est pas une méthode. En effet, UML ne propose pas une démarche de modélisation explicitant et encadrant toutes les étapes d'un projet, de la compréhension des besoins à la production du code de l'application. Une méthode se doit de définir une séquence d'étapes, partiellement ordonnées, dont l'objectif est de

produire un logiciel de qualité qui répond aux besoins des utilisateurs dans des temps et des coûts prévisibles.

Bien qu'UML ne soit pas une méthode, ses auteurs précisent néanmoins qu'une méthode basée sur l'utilisation UML doit être :

**Pilotée par les cas d'utilisation:** La principale qualité d'un logiciel étant son utilité, c'est-à-dire son adéquation avec les besoins des utilisateurs, toutes les étapes, de la spécification des besoins à la maintenance, doivent être guidées par les cas d'utilisation qui modélisent justement les besoins des utilisateurs.

**Centrée sur l'architecture:** L'architecture est conçue pour satisfaire les besoins exprimés dans les cas d'utilisation, mais aussi pour prendre en compte les évolutions futures et les contraintes de réalisation. La mise en place d'une architecture adaptée conditionne le succès d'un développement. Il est important de la stabiliser le plus tôt possible.

**Itérative et incrémentale:** L'ensemble du problème est décomposé en petites itérations, définies à partir des cas d'utilisation et de l'étude des risques. Les risques majeurs et les cas d'utilisation les plus importants sont traités en priorité. Le développement procède par des itérations qui conduisent à des livraisons incrémentales du système. Nous avons déjà présenté le modèle de cycle de vie par incrément dans la section 1.2.3.

### **3.1.1 Une méthode simple et générique**

Dans les sections qui suivent (sections 9.2, 9.3 et 9.4) nous allons présenter une méthode simple et générique qui se situe à mi-chemin entre UP (Unified Process), qui constitue un cadre général très complet de processus de développement, et XP (eXtreme Programming) qui est une approche minimaliste à la mode centrée

sur le code. Cette méthode est issue de celle présentée par Roques (2002) dans son livre « UML - Modéliser un site e-commerce » qui résulte de plusieurs années d'expérience sur de nombreux projets dans des domaines variés. Elle a donc montré son efficacité dans la pratique et est :

- conduite par les cas d'utilisation, comme UP, mais bien plus simple ;
- relativement légère et restreinte, comme XP, mais sans négliger les activités de modélisation en analyse et conception ;
- fondée sur l'utilisation d'un sous-ensemble nécessaire et suffisant du langage UML (modéliser 80% des problèmes en utilisant 20% d'UML).

Dans tous les cas, il faut garder à l'esprit qu'une méthode n'est pas une formule magique.

Le fait de produire des diagrammes UML selon un ordre établi n'est en aucun cas une garantie de réussite. Une méthode ne sert qu'à canaliser et ordonner les étapes de la modélisation. La valeur n'est pas dans la méthode mais dans les personnes qui la mettent en œuvre.

### **3.2 Identification des besoins et spécification des fonctionnalités**

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

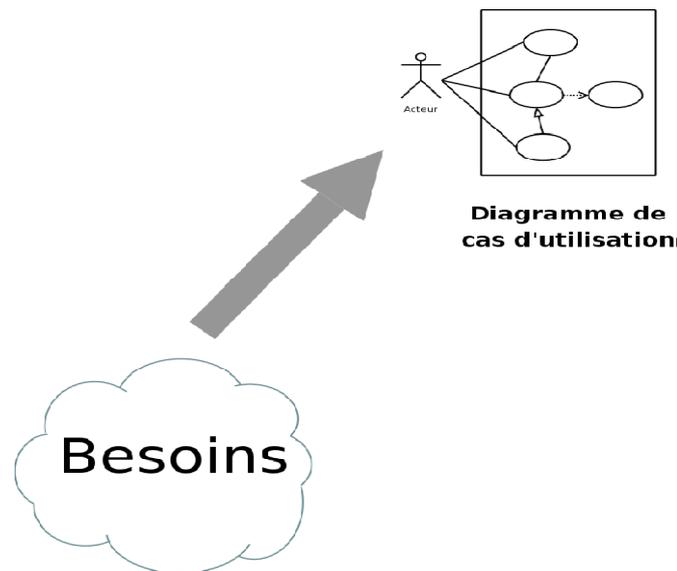


Figure1.2

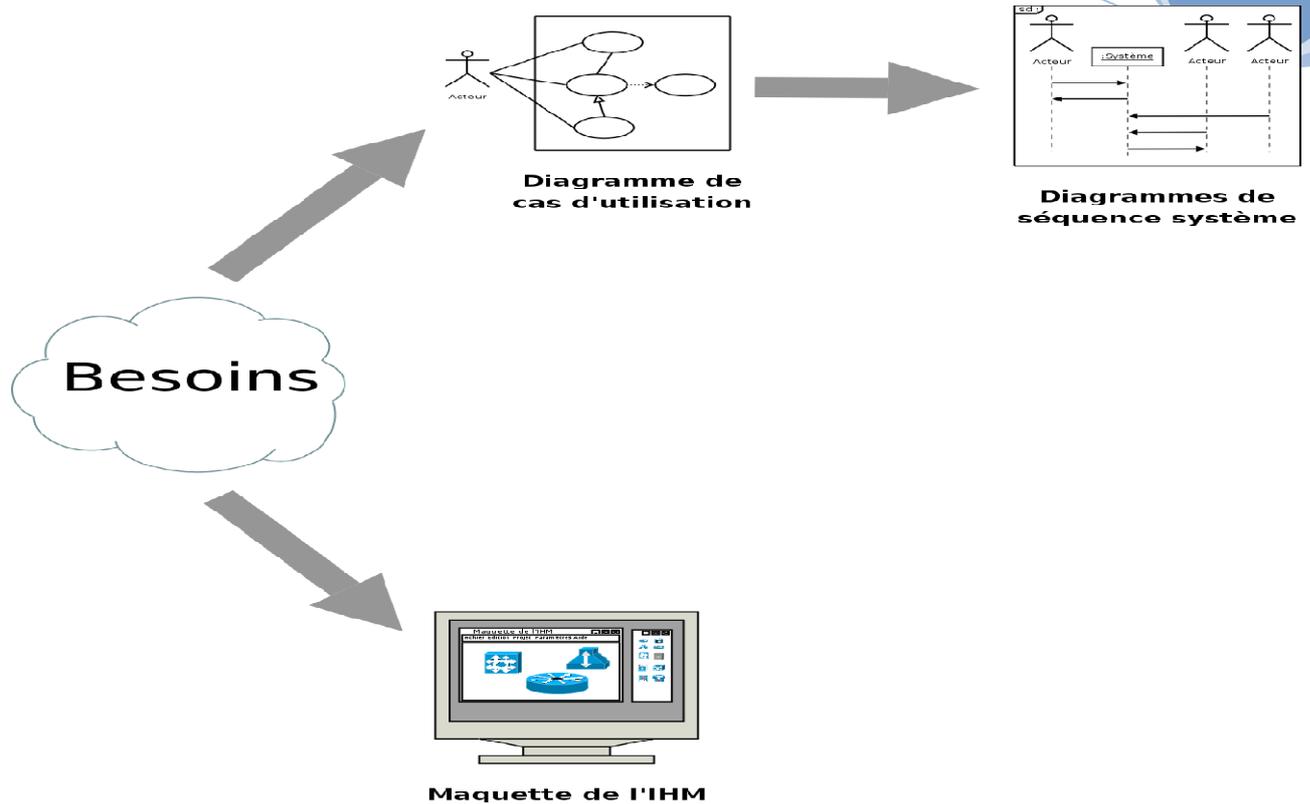
Les cas d'utilisation sont utilisés tout au long du projet. Dans un premier temps, on les crée pour identifier et modéliser les besoins des utilisateurs. Ces besoins sont déterminés à partir des informations recueillies lors des rencontres entre informaticiens et utilisateurs. Il faut impérativement proscrire toute considération de réalisation lors de cette étape.

Durant cette étape, vous devrez déterminer les limites du système, identifier les acteurs et recenser les cas d'utilisation. Si l'application est complexe, vous pourrez organiser les cas d'utilisation en paquetages.

Dans le cadre d'une approche itérative et incrémentale, il faut affecter un degré d'importance et un coefficient de risque à chacun des cas d'utilisation pour définir l'ordre des incréments à réaliser.

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. Les utilisateurs ont souvent beaucoup de difficultés à exprimer clairement et précisément ce qu'ils attendent du système. L'objectif de cette étape et des deux suivantes (section 1.1 et 1.2) est justement de les aider à formuler et formaliser ces besoins.

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



**Figure 1.3**

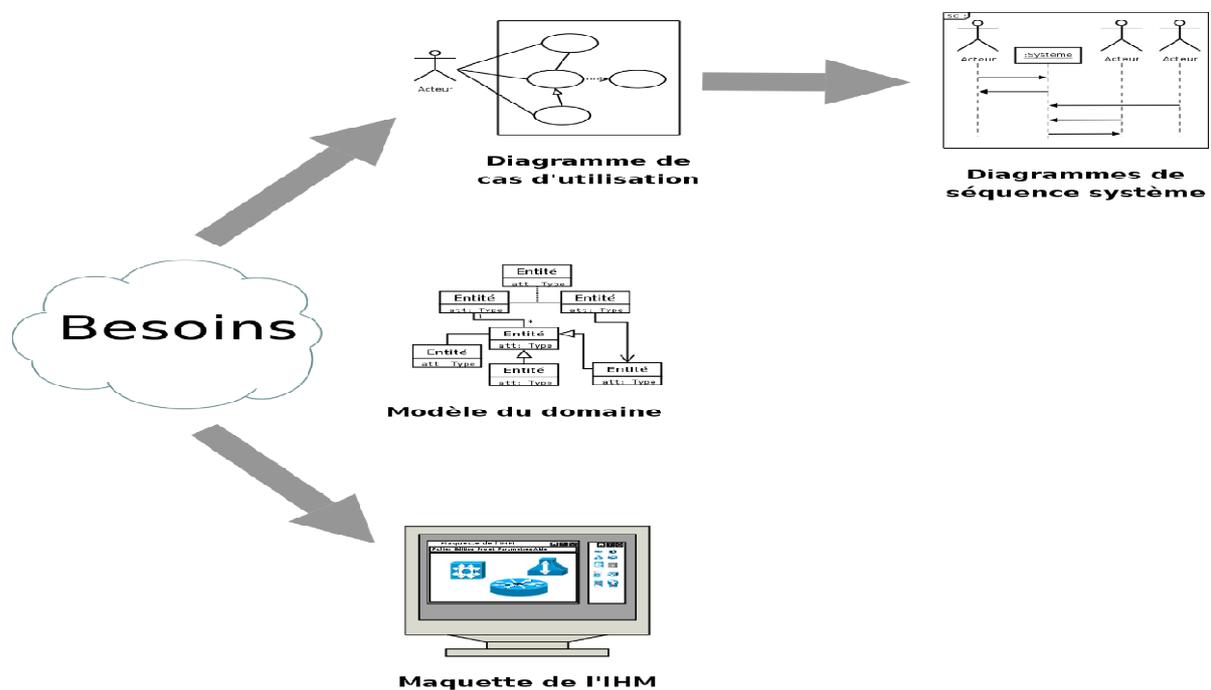
Une maquette d'IHM facilite les discussions avec les futurs utilisateurs.

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 (figure 1.3). 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.

### 3.3 Phases d'analyse

#### 3.3.1 Analyse du domaine : modèle du domaine



**Figure1.4**

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

La modélisation des besoins par des cas d'utilisation s'apparente à une analyse fonctionnelle classique. 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 (sections 3.1, 3.2 et 3.3). 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 (figure 1.4) appelée modèle du domaine. Ce modèle doit définir les classes qui modélisent les entités ou concepts présents dans le domaine (on utilise aussi le terme de métier) 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. Il faut absolument utiliser le vocabulaire du métier pour nommer les classes et leurs attributs. Les classes du modèle du domaine ne doivent pas contenir d'opération, mais seulement des attributs. Les étapes à suivre pour établir ce diagramme sont:

- identifier les entités ou concepts du domaine ;
- identifier et ajouter les associations et les attributs ;
- organiser et simplifier le modèle en éliminant les classes redondantes et en utilisant l'héritage ;
- le cas échéant, structurer les classes en paquetage selon les principes de cohérence et d'indépendance.

L'erreur la plus courante lors de la création d'un modèle du domaine consiste à modéliser un concept par un attribut alors que ce dernier devait être modélisé par une classe. Si la seule chose que recouvre un concept est sa valeur, il s'agit simplement d'un attribut. Par contre, si un concept recouvre un ensemble d'informations, alors il s'agit plutôt d'une classe qui possède elle-même plusieurs attributs.

### 3.3.2 Diagramme de classe participante

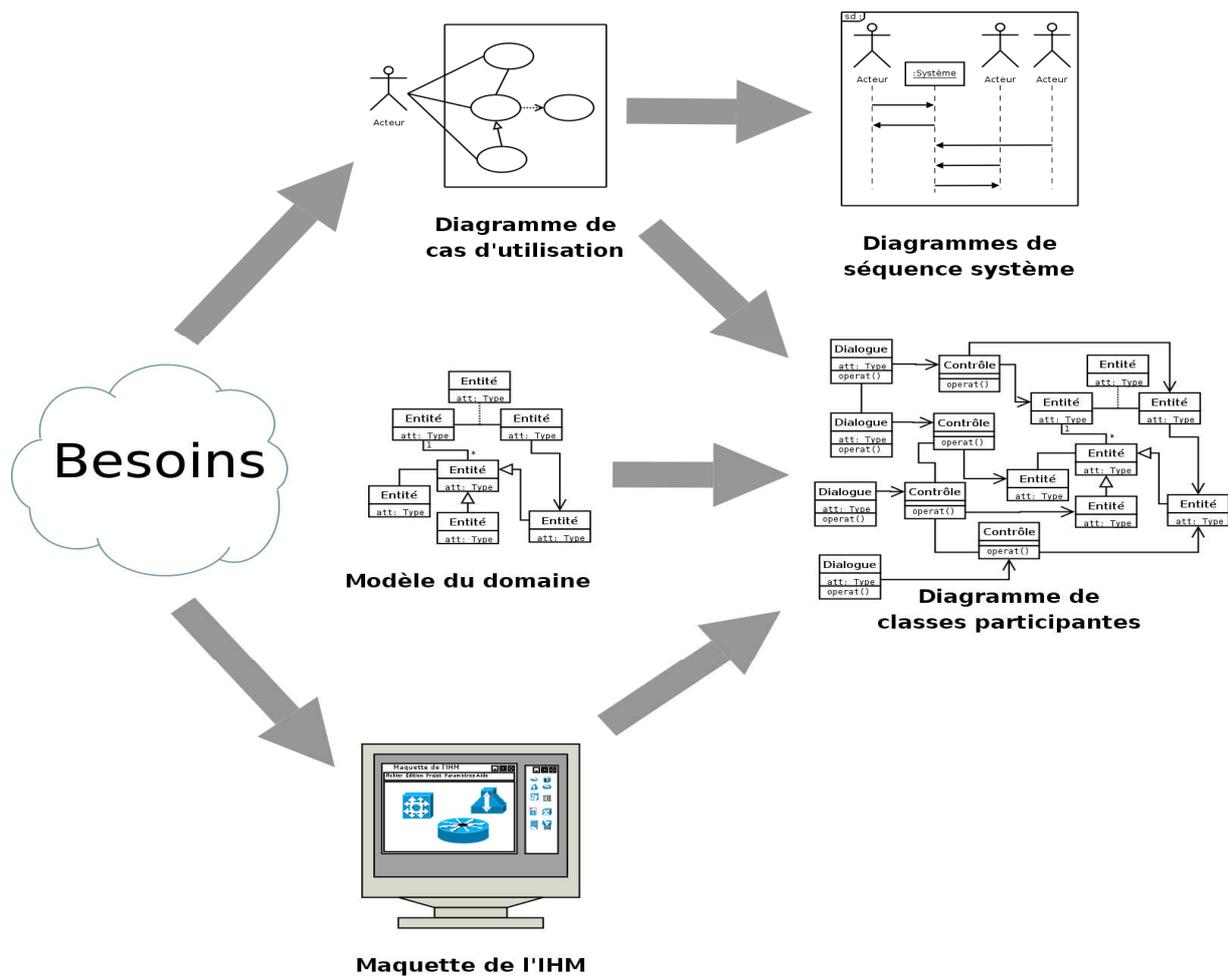


Figure1.5

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

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. Les diagrammes de conception logicielle n'apparaissent pas encore sur la figure 1.5.

Il n'est pas souhaitable que les utilisateurs interagissent directement avec les instances des classes du domaine par le biais de l'interface graphique. En effet, le modèle du domaine doit être indépendant des utilisateurs et de l'interface graphique. De même, l'interface graphique du logiciel doit pouvoir évoluer sans répercussion sur le cœur de l'application. C'est le principe fondamental du découpage en couches d'une application. Ainsi, le diagramme de classes participantes modélise trois types de classes d'analyse, les dialogues, les contrôles et les entités ainsi que leurs relations.

**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 présentée section 3.2.2. Il y a au moins un dialogue pour chaque association entre un acteur et un cas d'utilisation du diagramme de cas d'utilisation de la section 3.2.1.

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 au 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.

Lors de l'élaboration du diagramme de classes participantes, il faut veiller au respect des règles suivantes :

- Les entités, qui sont issues du modèle du domaine, ne comportent que des attributs.
- Les entités ne peuvent être en association qu'avec d'autres entités ou avec des contrôles, mais, dans ce dernier cas, avec une contrainte de navigabilité interdisant de traverser une association d'une entité vers un contrôle.
- Les contrôles ne comportent que des opérations. Ils implémentent la logique applicative (i.e. les fonctionnalités de l'application), et peuvent correspondre à des règles transverses à plusieurs entités. Chaque contrôle est généralement associé à un cas d'utilisation, et vice versa. Mais rien n'empêche de décomposer un cas d'utilisation complexe en plusieurs contrôles.

- Les contrôles peuvent être associés à tous les types de classes, y compris d'autres contrôles.

Dans le cas d'une association entre un dialogue et un contrôle, une contrainte de navigabilité doit interdire de traverser l'association du contrôle vers le dialogue.

- Les dialogues comportent des attributs et des opérations. Les attributs représentent des informations ou des paramètres saisis par l'utilisateur ou des résultats d'actions. Les opérations réalisent (généralement par délégation aux contrôles) les actions que l'utilisateur demande par le biais de l'IHM.
- Les dialogues peuvent être en association avec des contrôles ou d'autres dialogues, mais pas directement avec des entités.
- Il est également possible d'ajouter les acteurs sur le diagramme de classes participantes en respectant la règle suivante : un acteur ne peut être lié qu'à un dialogue.

Certaines classes possèdent un comportement dynamique complexe. Ces classes auront intérêt à être détaillées par des diagrammes d'états-transitions.

L'attribution des bonnes responsabilités, dégagée dans la section 3.2, aux bonnes classes est l'un des problèmes les plus délicats de la conception orientée objet. Ce problème sera affronté en phase de conception lors de l'élaboration des diagrammes d'interaction et du diagramme de classes de conception.

Lors de la phase d'élaboration du diagramme de classes participantes, le chef de projet a la possibilité de découper le travail de son équipe d'analystes par cas d'utilisation. L'analyse et

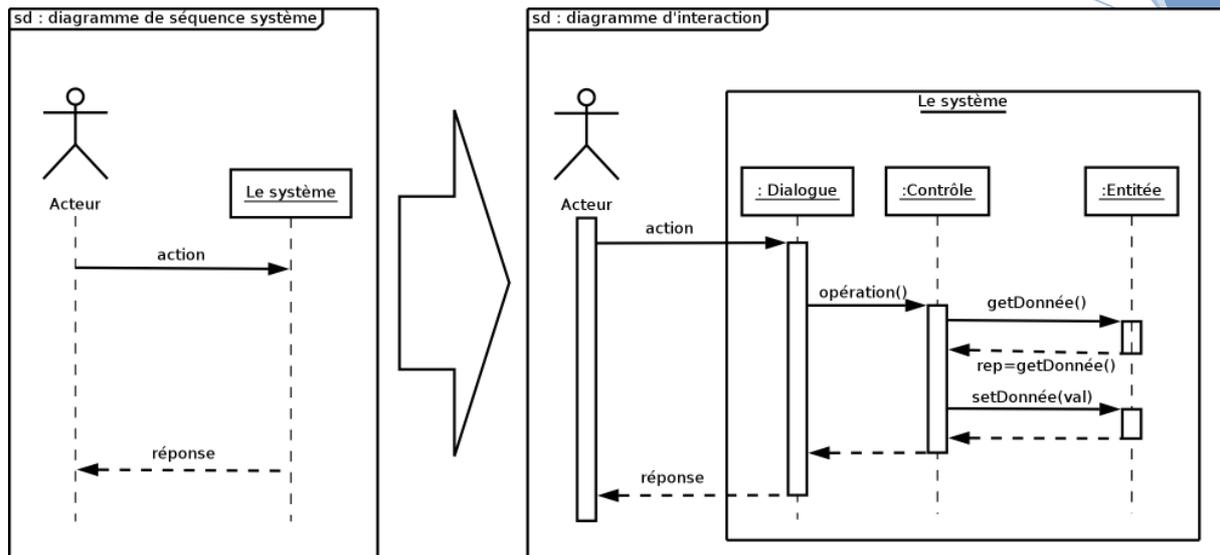
l'implémentation des fonctionnalités dégagées par les cas d'utilisation définissent alors les itérations à réaliser. L'ordonnancement des itérations étant défini par le degré d'importance et le coefficient de risque affecté à chacun des cas d'utilisation dans la section 3.1.

## **3.4 Phase de conception**

### **3.4.1 Diagramme de séquence**

Maintenant, il faut attribuer précisément les responsabilités de comportement, dégagée par les diagrammes de séquence système dans la section 3.2, aux classes d'analyse du diagramme de classes participantes élaboré dans la section 3.3.2. Les résultats de cette réflexion sont présentés sous la forme de diagrammes d'interaction UML. Inversement, l'élaboration de ces diagrammes facilite grandement la réflexion.

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



**Figure 1.6**

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

Pour chaque service ou fonction, il faut décider quelle est la classe qui va le contenir. Les diagrammes d'interactions (i.e. de séquence ou de communication) sont particulièrement utiles au concepteur pour représenter graphiquement ces décisions d'allocations des responsabilités.

Chaque diagramme va représenter un ensemble d'objets de classes différentes collaborant dans le cadre d'un scénario d'exécution du système.

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 suivre visuellement les interactions dynamiques entre objets, et les traitements réalisés par chacun d'eux. Avec un outil de modélisation UML (comme Rational Rose ou PowerAMC), la spécification de l'envoi d'un message entre deux objets crée effectivement une opération publique sur la classe de l'objet cible. Ce type d'outil permet réellement de mettre en œuvre l'allocation des responsabilités à partir des diagrammes d'interaction.

Par rapport aux diagrammes de séquences système de la section 3.2, 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. Les diagrammes de séquences élaborés dans cette section doivent donc toujours respecter les règles édictées dans la section 3.2. Ces règles doivent cependant être transposées car, pour que deux objets puissent interagir directement, il faut que :

- les classes dont ils sont issus soient en association dans le diagramme de classes participantes<sup>1</sup>;
- l'interaction respecte la navigabilité de l'association en question.

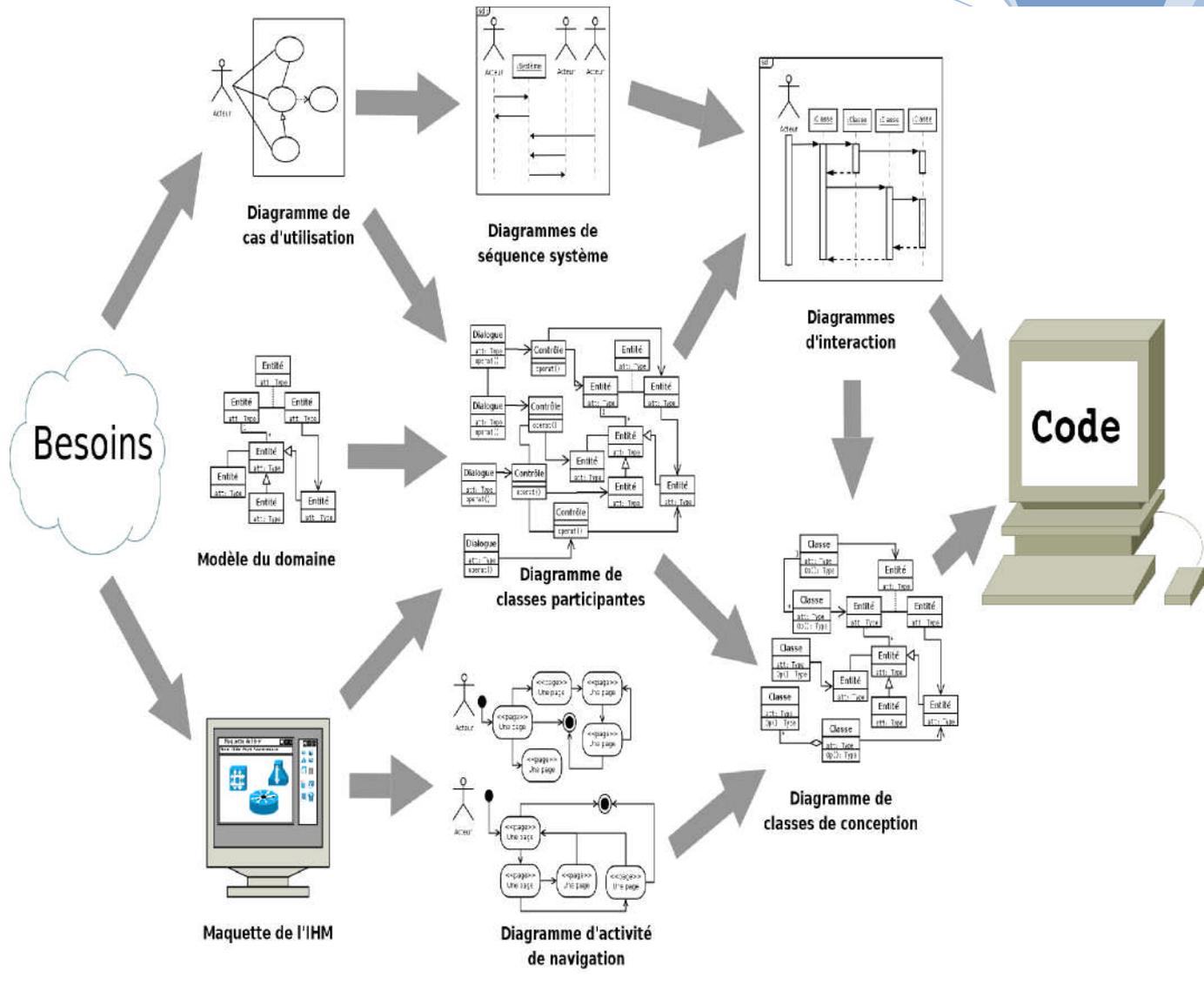
### **3.4.2 Diagramme de classes de conception**

L'objectif de cette étape est de produire le diagramme de classes qui servira pour l'implémentation (figure 1.7). Une première ébauche du diagramme de classes de conception a déjà été élaborée en parallèle du diagramme d'interaction (section 3.4.1). Il faut maintenant le compléter en précisant les opérations privées des différentes classes. Il faut prendre en compte les choix techniques, comme le choix du langage de programmation, le choix des différentes bibliothèques utilisées (notamment pour l'implémentation de l'interface graphique), etc.

Pour une classe, le couplage est la mesure de la quantité d'autres classes auxquelles elle est connectée par des associations, des relations de dépendances, etc. Durant toute l'élaboration du diagramme de classes de conception, il faut veiller à conserver un couplage faible pour obtenir une application plus évolutive et plus

facile à maintenir. L'utilisation des design patterns (cf. section ??) est fortement conseillée lors de l'élaboration du diagramme de classes de conception.

Pour le passage à l'implémentation, référez-vous à la section 3.6. Parfois, les classes du type entités ont intérêt à être implémentées dans une base de données relationnelle.



**Figure 1.7**

Chaîne complète de la démarche de modélisation du besoin jusqu'au code.

## *Chapitre 02 : identification des besoins*

### *Introduction*

- 1. cahier de charge*
- 2. dictionnaire de données*
- 3. diagramme de cas d'utilisation*
- 4. la description des cas d'utilisation*
- 5. diagramme de séquence*

- Introduction

Dans ce chapitre nous allons identifier les besoins du système à réaliser. Au début nous allons définir le cahier de charges qui contient toutes les informations de ce projet, puis nous commençons à identifier les acteurs qui interagiront avec le système, ensuite nous identifions et nous découvrons les cas d'utilisation du système.

1. Cahier de charge

- 1.1. Présentation du projet

La **gestion de la paie** est l'organisation du mode de rémunération des employés et du calcul des charges salariales et patronales relatives à cette rémunération.

Dans l'entreprise, la gestion de la paie est généralement de la responsabilité du service comptable et de la direction des ressources humaines.

La gestion de la paie constitue un aspect important de l'administration des salariés. Matérialisant la relation entre l'employeur et le salarié, le bulletin de paie est un document périodique obligatoire devant répondre à certaines règles. Le salaire, contrepartie de la prestation du travail effectuée par le salarié, constitue un coût pour l'entreprise qui doit être comptabilisé. La gestion de la paie permet aussi d'assurer un suivi des taxes salariales et patronales sur les salaires.

Les règles de traitement administratif de la **gestion de la paie** sont en constante évolution : modification périodique des bases et des taux de cotisations, modifications réglementaires ou conventionnelles (convention collective)... Nouvelles réformes de l'État, nouvelles aides sur une durée limitée. De fait la gestion de la paie nécessite de se maintenir informé en permanence de ces évolutions ou modifications.

Pour contourner cette contrainte un certain nombre d'entreprises choisissent d'externaliser la gestion de la paie

L'externalisation de la gestion de la paie consiste à transférer cette gestion à un prestataire externe et ainsi à s'affranchir de ses contraintes et responsabilités<sup>1</sup>.

L'externalisation de la gestion de la paie consiste à transférer cette gestion à un prestataire externe et ainsi à s'affranchir de ses contraintes et responsabilités<sup>1</sup>.

## 2. dictionnaire de données

Le dictionnaire de données permet de définir tous les termes et les concepts du projet.

<b>Notion</b>	<b>définition</b>
Paie	Définit la paie d'un salarié
fonction	Définit la fonction d'un salarié
Grade	Définit le grade d'un salarié
Comptable	Définit le chef service de comptabilité
Service comptabilité	Définit le service qui fait le mandatement et l'ordonnancement
Service budget	Définit le service qui fait l'engagement de la dépense
Compte bancaire	Définit le numéro du compte bancaire de salarier ou il reçoit la paie
Compte CCP	Définit le numéro du compte postale d'un salarier ou il reçoit la paie
Ancienneté	Définit l'ancienneté d'un salarié
IEP	Définit l'expérience professionnelle d'un salarié
Prime complète	Définit la prime complète d'un salarié
Prime supplémentaire	Définit la prime supplémentaire d'un salarié
Date de recrutement	Définit la date de recrutement d'un salarié
Rendement	Définit le rendement d'un salarié, il dépense chaque 3 mois
PFC	Définit la prime forfaitaire compensatoire d'un salarié
Salaire de base	Définit le salaire de base d'un salarié
Salaire brute	Définit le salaire de brute d'un salarié
Impôt	Définit l'impôt qui touche la paie d'un salarié
Fiche de paie	Définit la fiche de paie d'un salarié
ATS	Définit l'attestation de travail et de salaire d'un salarié

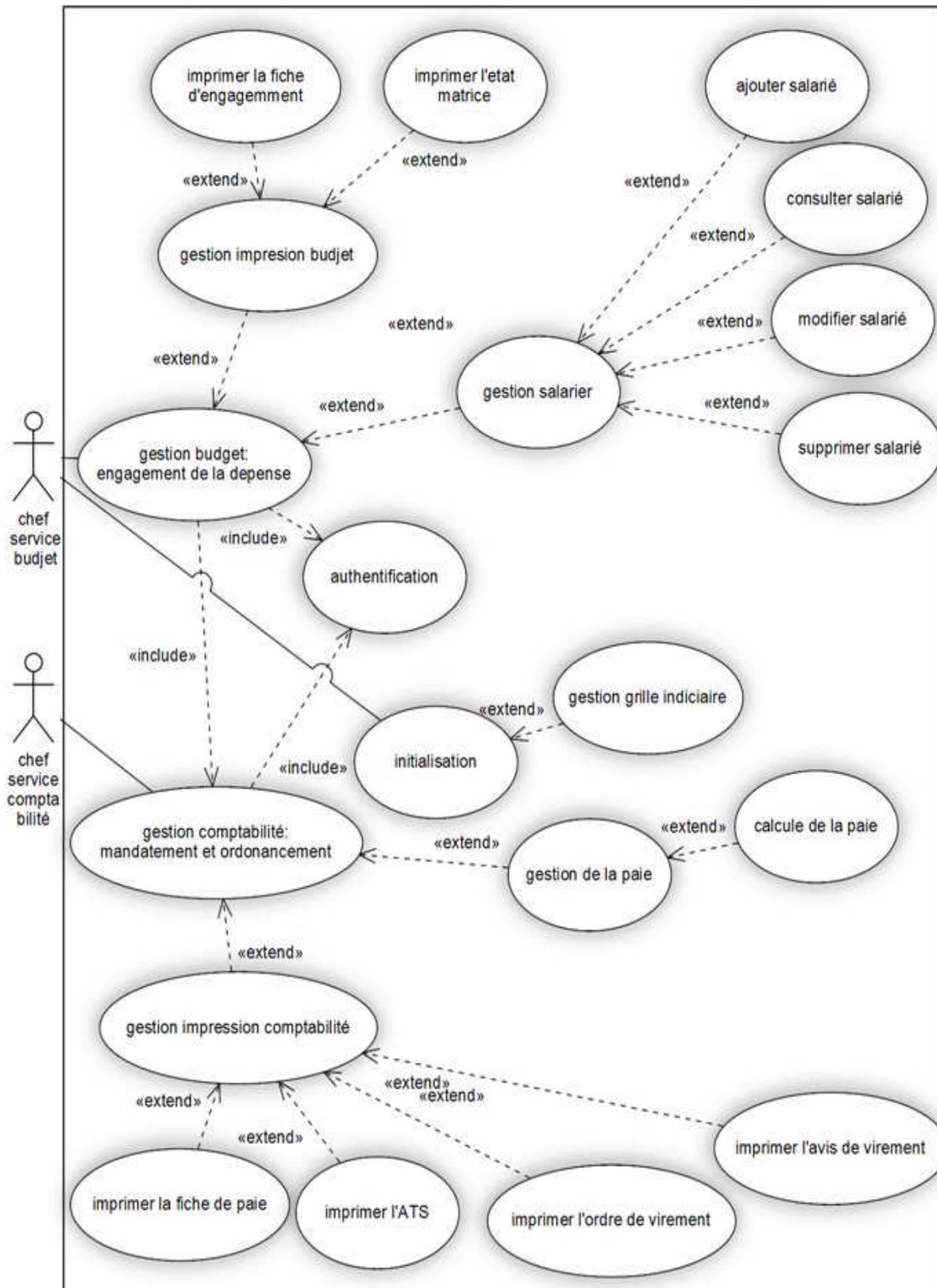
Bonus	Définit les bonus ajouté à la paie d'un salarié
CNAS	Définit la caisse nationale de sécurité sociale des salariés
Fiche de salaire annuel	Définit la fiche de salaire annuel d'un salarié
Mondât de paiement	Définit le mondât de paiement d'un salarié
Trésor	Définit le bureau ou envoyer tout les fiche des salariés
Mode de paiement	Définit le mode de paiement, il existe deux mode bancaire et postale
Réduction	Définit les réductions qui touchent la paie d'un salarié
Délai de paiement	Définit le mois ou le salarier reçoit la paie
SNMG	Définit le salaire national minimal garanti
Fiche d'engagement	Définit la fiche d'engagement envoyé au service comptabilité par le service budget
Grille indiciaire des traitements	Définit un tableau comporte les groupe, catégorie, indice minimaux, échelon .....
Echelon	Définit l'échelon du salarié dans la grille indiciaire du traitement
Indice d'échelon	Définit l'indice d'échelon du salarié dans la grille indiciaire du traitement
Indice minimale	Définit l'indice minimal du salarié dans la grille indiciaire du traitement
taux IPP	Définit le taux perf pédagogique d'un salarié
Taux d'encadrement	Définit le taux d'encadrement d'un salarié
Taux qualifs enseignant	Définit le Taux qualifs enseignant d'un salarié
Taux IFS	Définit le taux IFS d'un salarié
Ind service	Définit l'ind service d'un salarié

<b>Action</b>	<b>définition</b>
Connexion	Permet à un utilisateur de se connecter au système
Initialisation	Permet d'initialiser les informations
Ajouter salarié	Permet d'ajouter un salarié
Supprimer un salarié	Permet de supprimer un salarié
Modifier salarié	Permet de modifier un salarié
Consulter salarié	Permet de consulter un salarié
Rechercher salarié	Permet de rechercher un salarié
Quitter	Permet à l'utilisateur de quitter le système
Calculer la paie	Permet de calculer la paie d'un salarié pour un mois
Imprimer la fiche de paie	Permet d'imprimer la fiche de paie d'un salarié
Imprimer avis de virement	Permet d'imprimer l'avis de virement d'un salarié
Imprimer ordre de virement	Permet d'imprimer l'ordre de virement d'un salarié
Imprimer le mondât de paiement	Permet d'imprimer le mondât de paiement d'un salarié
Imprimer l'ATS	Permet d'imprimer l'attestation de travail et de salaire d'un salarié
Imprimer la fiche de salaire annuel	Permet d'imprimer la fiche de salaire annule d'un salarié
Imprimer la fiche d'engagement	Permet d'imprimer la fiche d'engagement d'un salarié

### **3. 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, vous devrez déterminer les limites du système, identifier les acteurs et recenser les cas d'utilisation.

### 3.1. Notre diagramme d'utilisation



#### 4. description de cas d'utilisation

Un cas d'utilisation est une manière spécifique d'utiliser un système. Les acteurs sont à l'extérieur du système ; ils modélisent tout ce qui interagit avec lui. Un cas d'utilisation réalise un service de bout en bout.

Avec un déclenchement, un déroulement et une fin pour l'acteur qui l'initie.

Les cas d'utilisation permettant de trouver la classe de la vue structurelle du modèle d'analyse.

Les scénarios élaborés par cas d'utilisation permettent de trouver les opérations des interfaces de la vue logique du modèle de conception système.

Le cas d'utilisation identifie des fonctions qu'il faut repartir sur le dépoilement du modèle de la conception système.

##### 4.1. S'authentifier

Cas d'utilisation	S'authentifier	
acteur	Chef service budget et chef service comptabilité	
But	Vérifier l'autorisation d'accéder au système	
Pré condition	Les acteurs possèdent les informations d'accès	
Post condition	les acteurs sont authentifiés par le système	
Scénario nominale	etape	action
	1	L'acteur choisit d'accéder au système
	2	Le système demande à l'acteur d'entrer un login et un mot de passe
	3	Les acteurs saisi le login et le mot de passe et valide
Scénario alternatif	étape	action
	4	<p><b>Le mot de passe est erroné</b></p> <p>1. le système affiche un message d'erreur</p> <p>2. le système propose à l'utilisateur de renseigner une nouvelle fois le login et le mot de passe</p> <p>3. reprise de l'enchaînement du scénario nominale au point 2</p>

Scénario erreurs	<b>Login et mot de passe erroné pour la 3<sup>eme</sup> fois</b> Il sera déconnecté du system
------------------	--

#### 4.2. Gestion comptabilité

Cas d'utilisation	Gestion comptabilité	
acteur	Chef service comptabilité	
But	il permettre au chef service comptabilité de gère la comptabilité	
Pré condition	Le chef service comptabilité authentifié	
Post condition	Les informations ont été enregistrées	
Scénario nominale	étape	action
	1	Le chef service comptabilité accéder au system
	2	Le system présente l'état actuel
	3	1. imprimé {fiche de paie, ATS, avis et ordre de virement, .....} sous cas : gestion impression comptabilité 2. calcule de la paie : sous cas gestion de la paie
Scénario alternatif	étape	Action
	3	le chef service comptabilité annule l'accès au sous cas et retour en 2
Scénario erreurs	/	

#### 4.3. Gestion budget

Cas d'utilisation	Gestion budget
acteur	Chef service budget
But	Il permettre au chef service budget de gère le budget
Pré condition	Le chef service budget authentifié

Post condition	Les informations ont été enregistrées	
Scénario nominale	étape	action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	1. {ajouter, supprimer, modifier, consulter, rechercher} Salarié sous cas : gestion salarier 2. imprimer {fiche d'engagement, état matrice} sous cas : gestion impression budget
Scénario alternatif	étape	action
	3	le chef service budget annule l'accès au sous cas et retour en 2
Scénario erreurs	/	

#### 4.4 Initialisation

Cas d'utilisation	Initialisation	
acteur	Chef service budget	
But	Il permettre au chef service budget d'initialiser la grille indiciaire	
Pré condition	Le chef service budget possède les informations	
Post condition	Les informations ont été enregistrées	
Scénario nominale	étape	action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	-Le system demande d'initialiser la grille indiciaire 1. initialiser la grille indiciaire sous cas : gestion grille indiciaire

Scénario alternatif	etape	action
	3	le chef service budget annule l'accès au sous cas et retour en 2
Scénario erreurs	/	

#### 4.5 Gestion salarié

Cas d'utilisation	Gestion salarié	
acteur	Chef service budget	
But	Ajouter modifier supprimer consulter rechercher : salarié	
Pré condition	Le chef service budget possède les informations concernant un salarié	
Post condition	Les informations ont été enregistrées	
Scénario nominale	étape	action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	-Le chef service choisit l'opération ajouté salarié sous cas : ajouter salarié - Le chef service choisit l'opération modifié salarié sous cas : modifier salarié - Le chef service choisit l'opération supprimé salarié sous cas : supprimer salarié - Le chef service choisit l'opération recherché salarié sous cas : rechercher salarié - Le chef service choisit l'opération consulté salarié sous cas : consulter salarié
	4	Le system enregistre les modifications
Scénario alternatif	etape	Action
	3	le chef service budget annule l'accès au sous cas et retour en 2

Scénario erreurs	/
------------------	---

#### 4.6 Gestion de la paie

Cas d'utilisation	Gestion de la paie	
acteur	Chef service comptabilité	
But	Calcule de la paie	
Pré condition	Le chef service comptabilité possède les informations concernant un salarié	
Post condition	Les informations ont été enregistrées	
Scénario nominale	étape	action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	calcule de la paie sous cas : calcule de la paie
	4	Le system enregistre les modifications
Scénario alternatif	etape	Action
	3	le chef service comptabilité annule l'accès au sous cas et retour en 2
Scénario erreurs	/	

4.7 Gestion impression budget

Cas d'utilisation	Gestion impression budget	
acteur	Chef service budget	
But	Imprimer fiche d'engagement et l'état matrice	
Pré condition	Le chef service budget possède les informations concernant un salarié	
Post condition	Les informations ont été enregistrées	
Scénario nominale	étape	action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	-imprimer fiche d'engagement sous cas : imprimer fiche d'engagement -imprimer état matrice sous cas : imprimer état matrice
	4	Le system enregistre les modifications
Scénario alternatif	etape	Action
	3	le chef service budget annule l'accès au sous cas et retour en 2
Scénario erreurs	/	

#### 4.8 Gestion impression comptabilité

Cas d'utilisation	Gestion impression comptabilité	
acteur	Chef service comptabilité	
But	Imprimer (fiche de paie, ATS, avis de virement, ordre de virement)	
Pré condition	Le chef service comptabilité possède les informations concernant un salarié	
Post condition	Les informations ont été enregistrées	
Scénario nominale	étape	action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	-imprimer fiche de paie sous cas : imprimer fiche de paie -imprimer ATS sous cas : imprimer ATS - imprimer avis de virement sous cas : imprimer avis de virement - imprimer ordre de virement sous cas : imprimer ordre de virement
	4	Le system enregistre les modifications
Scénario alternatif	étape	Action
	3	le chef service comptabilité annule l'accès au sous cas et retour en 2
Scénario erreurs	/	

4.9 ajouter salarié

Cas d'utilisation	Ajouter salarié	
acteur	Chef service budget	
But	Ajouter un salarié	
Pré condition	Le chef service budget possède les informations concernant un salarié	
Post condition	Les informations ont été enregistrées	
Scénario nominale	étape	action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	-le système demande de saisi les informations du salarié et le donne un numéro -le chef service saisi les informations du salarié et donne un numéro -le système vérifier le numéro : ajout avec succès
	4	Le system enregistre les modifications
Scénario alternatif	étape	Action
	3	- Le numéro du salarié existe déjà : affiche un message d'existence et retour en 2
Scénario erreurs	/	

4.10 modifier salarié

Cas d'utilisation	Modifier salarié	
acteur	Chef service budget	
But	Modifier un salarié	
Pré condition	Le chef service budget possède les informations concernant un salarié	
Post condition	Les informations ont été enregistrées	
Scénario nominale	étape	Action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	-le système demande de saisi le numéro du salarié -le chef service saisi le numéro du salarié -le système affiche les informations du salarié -le système demande de saie les nouvelles informations de salarié -le chef service saisi les nouvelles informations de salarié et valide la modification
	4	Le system enregistre les modifications
Scénario alternatif	étape	Action
	3	- Le numéro du salarié n'existe pas : affiche un message d'absence et retour en 2
Scénario erreurs	/	

4.11 supprimé salarié

Cas d'utilisation	supprimer salarié	
acteur	Chef service budget	
But	supprimer un salarié	
Pré condition	Le chef service budget possède les informations concernant un salarié	
Post condition	Les informations ont été enregistrées	
Scénario nominale	étape	Action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	-le système demande de saisi le numéro du salarié -le chef service saisi le numéro du salarié -le système affiche les informations du salarié -le système demande de confirmer la suppression -le chef service confirme la suppression
	4	Le system enregistre les modifications
Scénario alternatif	étape	Action
	3	- Le numéro du salarié n'existe pas : affiche un message d'absence et retour en 2
Scénario erreurs	/	

4.12 consulter salarié

Cas d'utilisation	consulter salarié	
acteur	Chef service budget	
But	consulter un salarié	
Pré condition	Le chef service budget possède les informations concernant un salarié	
Post condition	Le chef service fait la consultation	
Scénario nominale	étape	Action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	-le système demande de saisi le numéro du salarié -le chef service saisi le numéro du salarié -le système affiche les informations du salarié
Scénario alternatif	étape	Action
	3	- Le numéro du salarié n'existe pas : affiche un message d'absence et retour en 2
Scénario erreurs	/	

4.13 rechercher salarié

Cas d'utilisation	rechercher salarié	
acteur	Chef service budget	
But	Rechercher sur un salarié	
Pré condition	Le chef service budget possède les informations concernant un salarié	
Post condition	Le système affiche les informations du salarié	
Scénario nominale	étape	Action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	-le système demande de saisi le numéro du salarié -le chef service saisi le numéro du salarié -le système affiche les informations du salarié
Scénario alternatif	étape	Action
	3	- Le numéro du salarié n'existe pas : affiche un message d'absence et retour en 2
Scénario erreurs	/	

4.14 calcul de la paie

Cas d'utilisation	Calcul de la paie	
acteur	Chef service comptabilité	
But	Calcul de la paie	
Pré condition	Le chef service comptabilité possède les informations concernant un salarié	
Post condition	Les informations ont été enregistrées	
Scénario nominale	étape	action
	1	Le chef service comptabilité accédé au system
	2	Le system présente l'état actuel
	3	<ul style="list-style-type: none"> <li>- le chef service choisir l'opération paie salarié</li> <li>-le système demande de choisi le mois</li> <li>-le chef service choisi le mois</li> <li>-le système donne trois cas (par personnel, par banque, tout les personne) <ul style="list-style-type: none"> <li>a- par personnel <ul style="list-style-type: none"> <li>-le système demande de saisi le numéro du salarier</li> <li>-le chef service saisi le numéro salarié et valide la saisi</li> <li>-la paie est calculée</li> </ul> </li> <li>b-par banque <ul style="list-style-type: none"> <li>-le système demande de choisir la banque</li> <li>-le chef service choisir la banque et valide le choix</li> <li>-la paie est calculée</li> </ul> </li> <li>c-tout les personne <ul style="list-style-type: none"> <li>-le chef service valide le choix</li> <li>-la paie est calculée</li> </ul> </li> </ul> </li> </ul>
	4	Le system enregistre les modifications
Scénario		

alternatif	étape	Action
	3	- 3-a-2 Le numéro du salarié n'existe pas : affiche un message d'absence et retour en 2
Scénario erreurs	/	

#### 4.15 imprimer état matrice

Cas d'utilisation	Imprimer état matrice	
acteur	Chef service budget	
But	Imprimer l'état matrice	
Pré condition	Le chef service budget possède les informations concernant un salarié	
Post condition	L'état matrice est imprimée	
Scénario nominale	étape	Action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	-le chef service choisir l'opération 'imprimer' -le chef service choisir l'état matrice -le système demande de saisi le début et la fin de la durée d'engagement -le chef service saisi le début et la fin de la durée d'engagement -le système demande de saisi les informations du salarié -le chef service saisi le numéro du salarié et valide l'impression -l'état matrice est imprimé
Scénario alternatif	étape	Action
	3	Le numéro du salarié n'existe pas : affiche un message d'absence et retour en 2
Scénario erreurs	/	

4.16 imprimer la fiche d'engagement

Cas d'utilisation	Imprimer fiche d'engagement	
acteur	Chef service budget	
But	Imprimer fiche d'engagement	
Pré condition	Le chef service budget possède les informations concernant un salarié	
Post condition	La fiche d'engagement est imprimée	
Scénario nominale	étape	Action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	-le chef service choisir l'opération 'imprimer' -le chef service choisir la fiche d'engagement -le système demande de saisi le début et la fin de la durée d'engagement -le chef service saisi le début et la fin de la durée d'engagement -le système demande de saisi les informations du salarié -le chef service saisi le numéro du salarié et valide l'impression -l'état matrice est imprimé
Scénario alternatif	étape	Action
	3	Le numéro du salarié n'existe pas : affiche un message d'absence et retour en 2
Scénario erreurs	/	

4.17 imprimer la fiche de paie

Cas d'utilisation	Imprimer la fiche de paie	
acteur	Chef service comptabilité	
But	Imprimer la fiche de paie	
Pré condition	Le chef service comptabilité possède les informations concernant un salarié	
Post condition	La fiche de paie est imprimée	
Scénario nominale	étape	Action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	<ul style="list-style-type: none"> <li>- le chef service choisir l'opération 'imprimer'</li> <li>-le chef service choisir la fiche de paie</li> <li>-le système donne deux cas (mensuel-annuel) <ul style="list-style-type: none"> <li>a- cas mensuel</li> </ul> </li> <li>-le système demande de choisi le mois</li> <li>- le chef service choisi le mois et la valide le choix</li> <li>- le système demande de saisi le numéro du salarié</li> <li>- le chef service saisi le numéro du salarié</li> <li>-le système vérifier le numéro</li> <li>-fiche de paie imprimée</li> <li>b- cas annuel</li> <li>-le système demande de choisi l'année</li> <li>- le chef service choisi l'année et la valide le choix</li> <li>- le système demande de saisi le numéro du salarié</li> <li>-le système vérifier le numéro</li> <li>-fiche de paie imprimée</li> </ul>
Scénario alternatif	étape	Action
	3	Le numéro du salarié n'existe pas : affiche un message d'absence et retour en 2
Scénario erreurs	/	

4.18 imprimer l'ATS

Cas d'utilisation	Imprimer ATS	
Acteur	Chef service comptabilité	
But	Imprimer l'ATS	
Pré condition	Le chef service comptabilité possède les informations concernant un salarié	
Post condition	L'ATS est imprimé	
Scénario nominale	Etape	Action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	- le chef service choisir l'opération 'imprimer' -le chef service choisir ATS - le système demande de saisi le numéro du salarié -le chef service saisi le numéro du salarié -le système vérifier le numéro -ATS imprimée
Scénario alternatif	étape	Action
	3	Le numéro du salarié n'existe pas : affiche un message d'absence et retour en 2
Scénario erreurs	/	

4.19 imprimer avis de virement

Cas d'utilisation	Imprimer avis de virement	
Acteur	Chef service comptabilité	
But	Imprimer l'avis de virement	
Pré condition	Le chef service comptabilité possède les informations concernant un salarié	
Post condition	avis de virement est imprimé	
Scénario nominale	Etape	Action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	- le chef service choisir l'opération 'imprimer' -le chef service choisir avis de virement - le système demande de saisi le numéro du salarié -le chef service saisi le numéro du salarié -le système vérifier le numéro -avis de virement imprimée
Scénario alternatif	étape	Action
	3	Le numéro du salarié n'existe pas : affiche un message d'absence et retour en 2
Scénario erreurs	/	

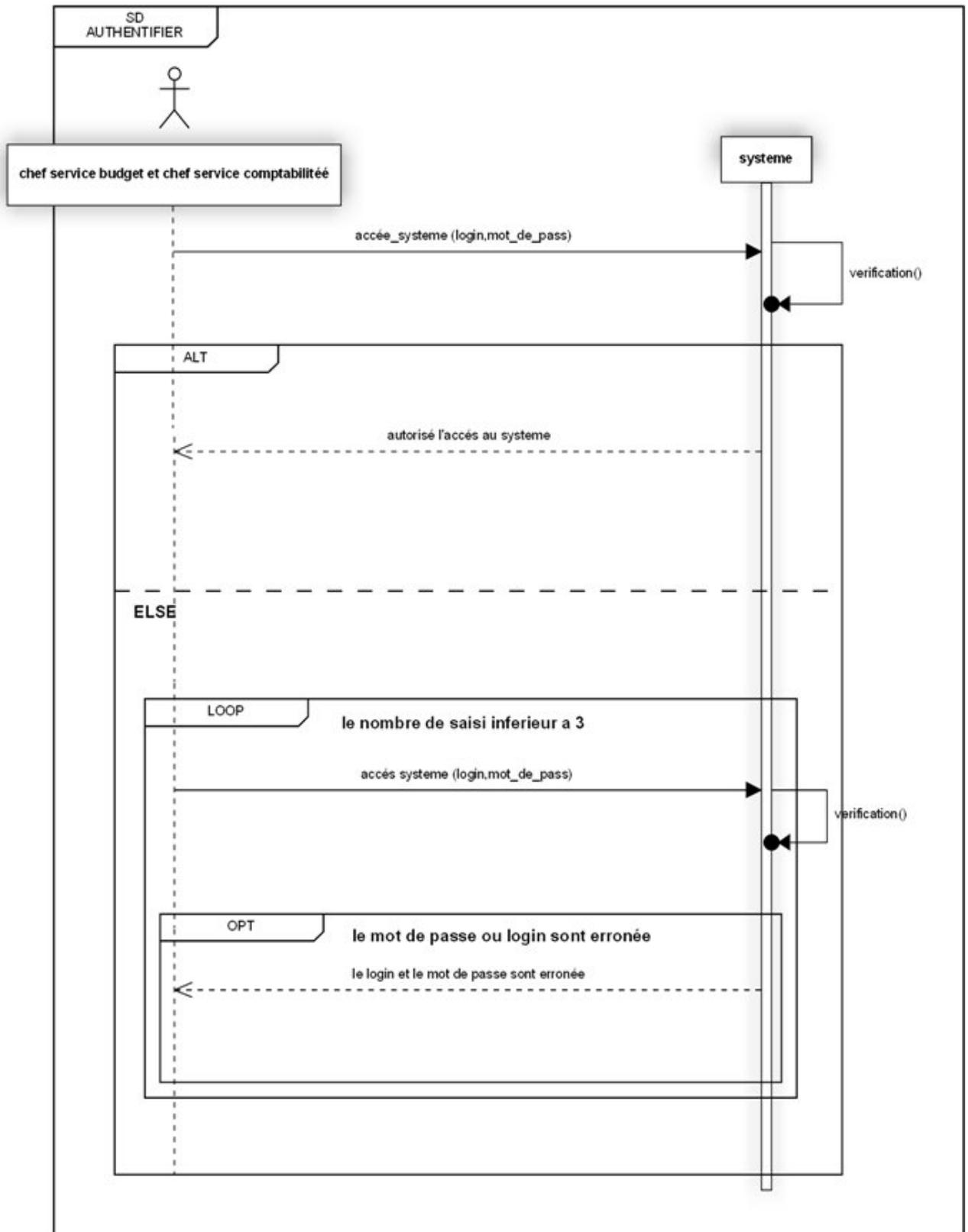
4.20 imprimer ordre de virement

Cas d'utilisation	Imprimer ordre de virement	
acteur	Chef service comptabilité	
But	Imprimer l'ordre de virement	
Pré condition	Le chef service comptabilité possède les informations concernant un salarié	
Post condition	L'ordre de virement est imprimé	
Scénario nominale	Etape	Action
	1	Le chef service budget accédé au system
	2	Le system présente l'état actuel
	3	- le chef service choisir l'opération 'imprimer' -le chef service choisir ordre de virement - le système demande de saisi le numéro du salarié -le chef service saisi le numéro du salarié -le système vérifier le numéro -ordre de virement imprimée
Scénario alternatif	étape	Action
	3	Le numéro du salarié n'existe pas : affiche un message d'absence et retour en 2
Scénario erreurs	/	

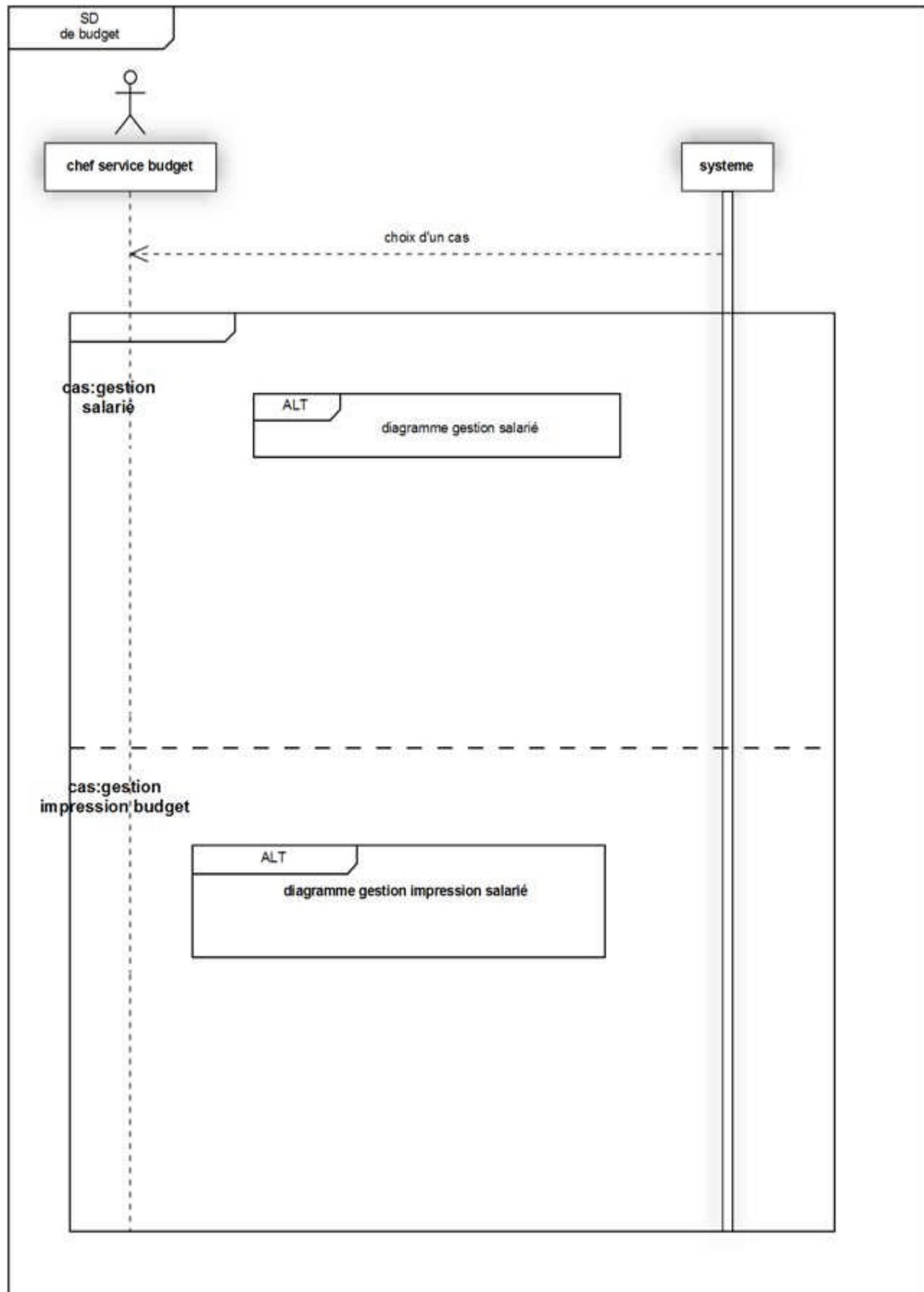
## 5. Diagramme de séquence système

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

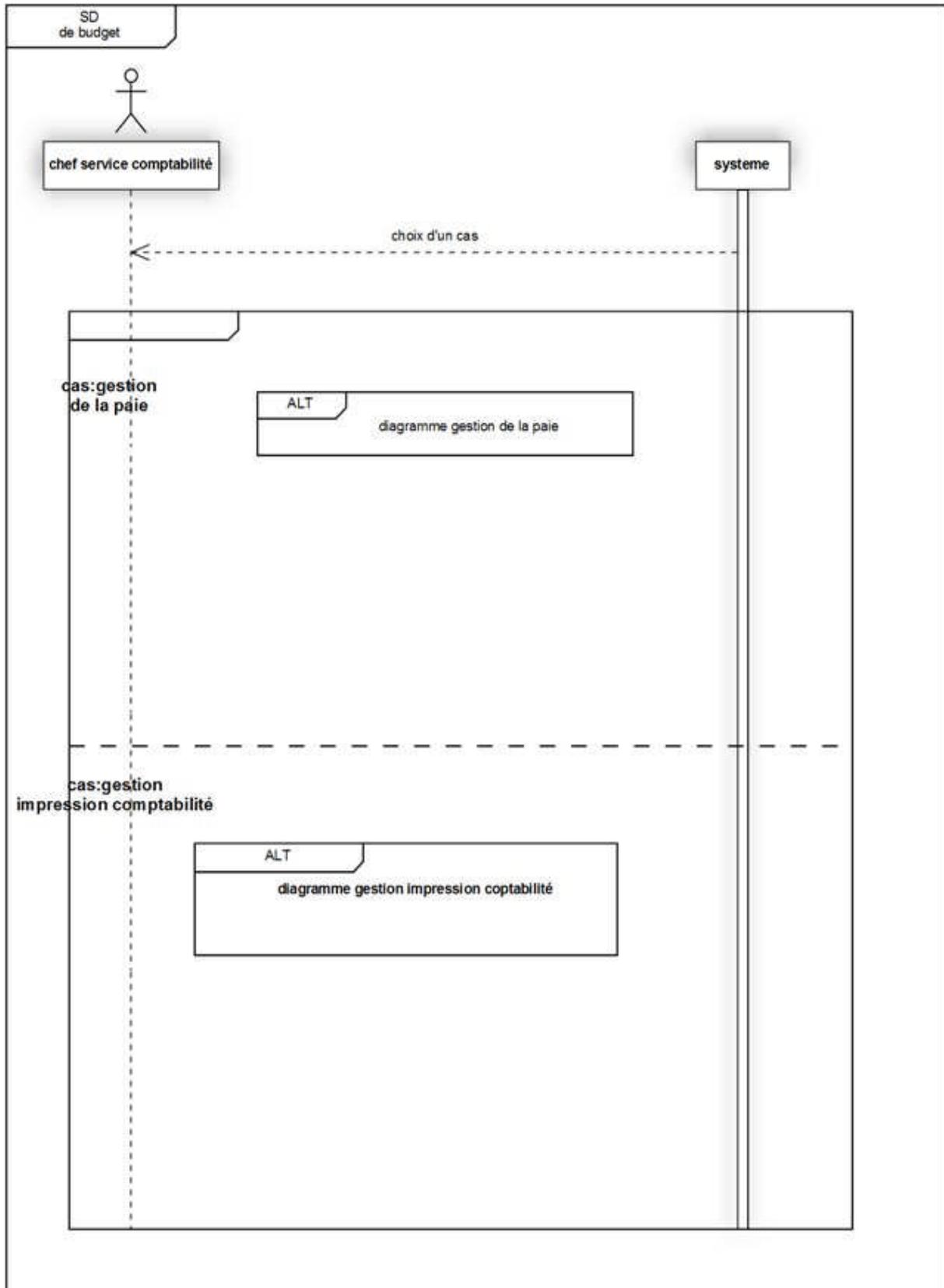
### 5.1 Authentifier



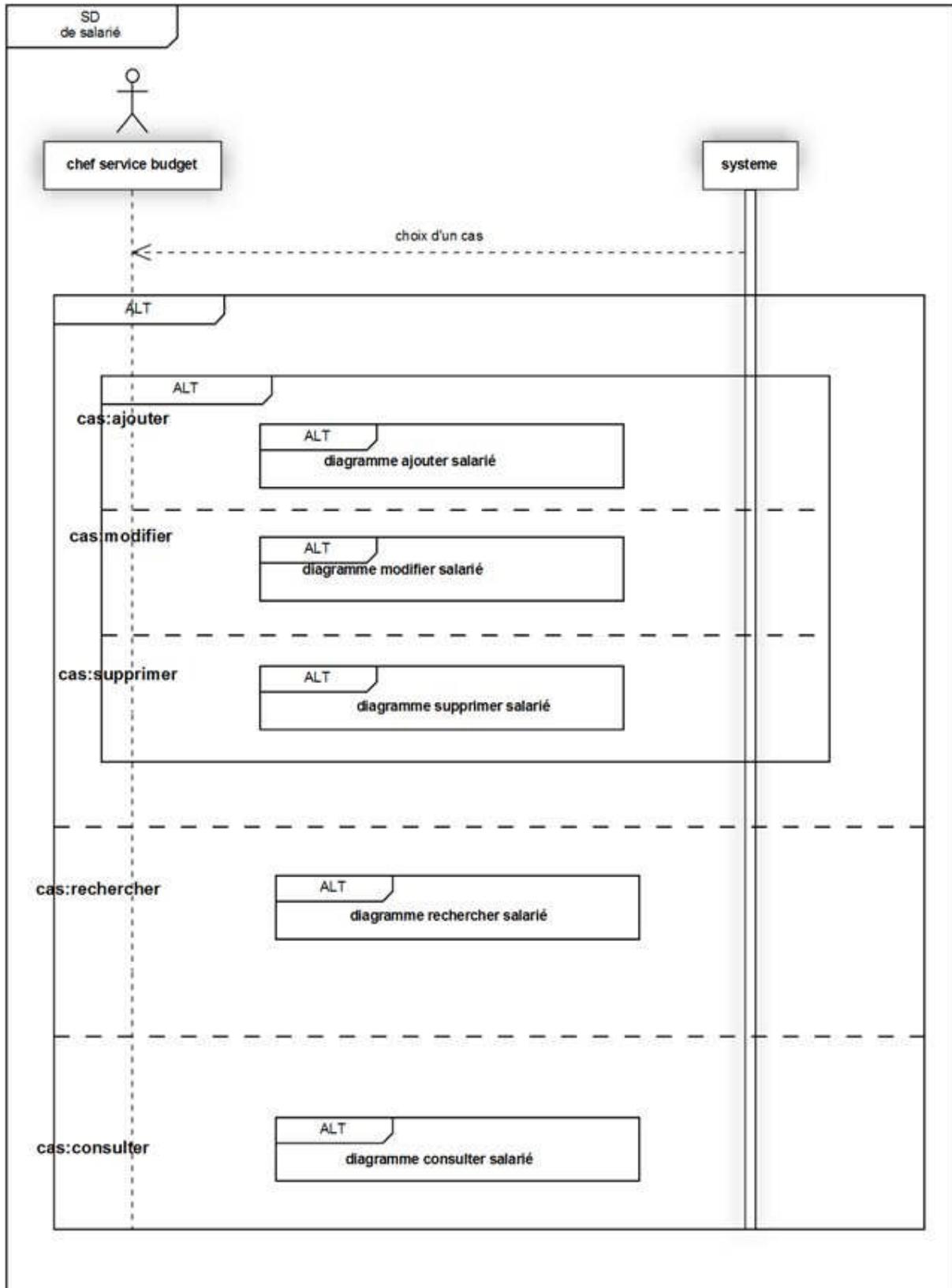
## 5.2. Gestion budget



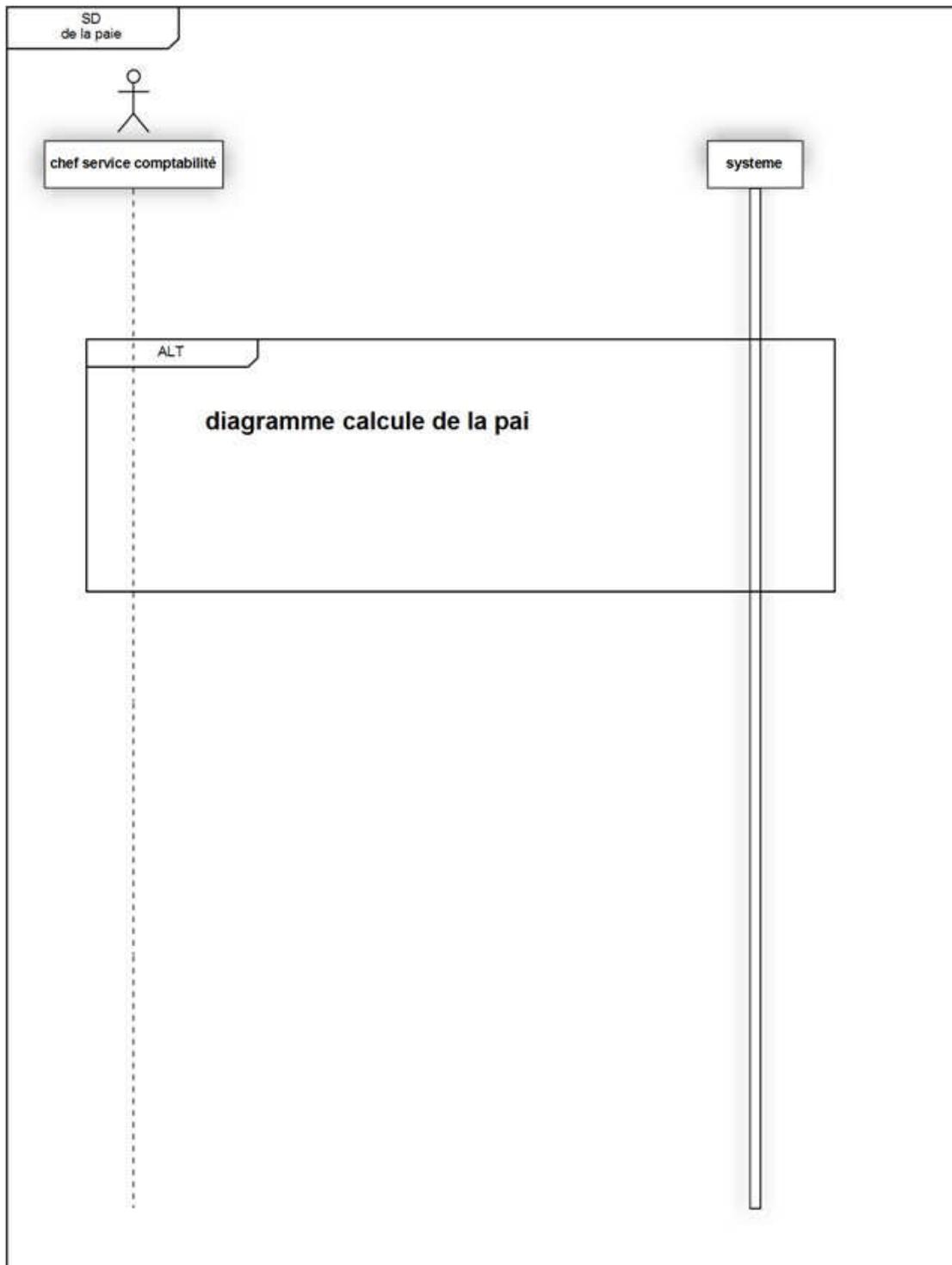
### 5.3 Gestion comptabilité



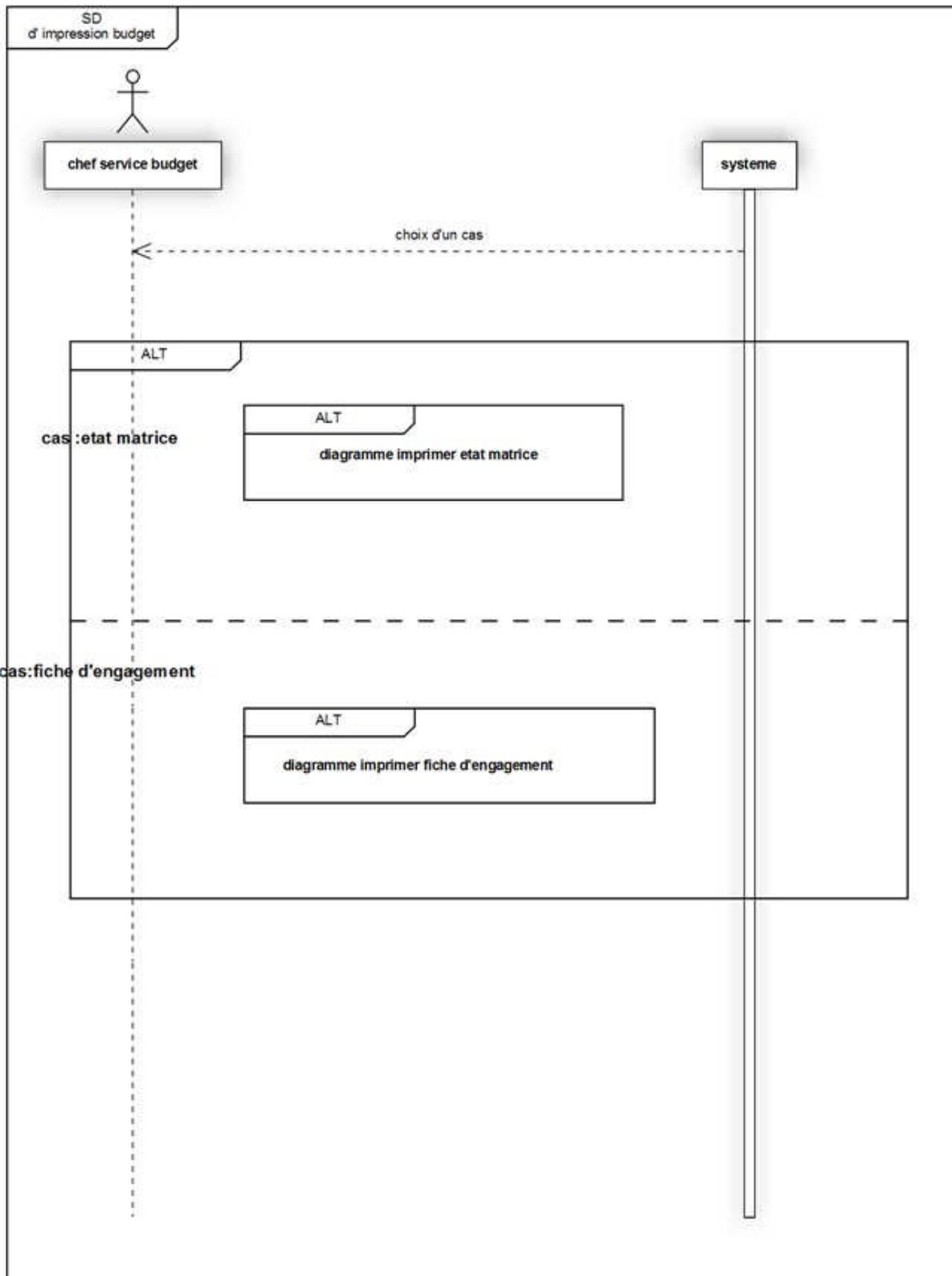
### 5.4 Salarié



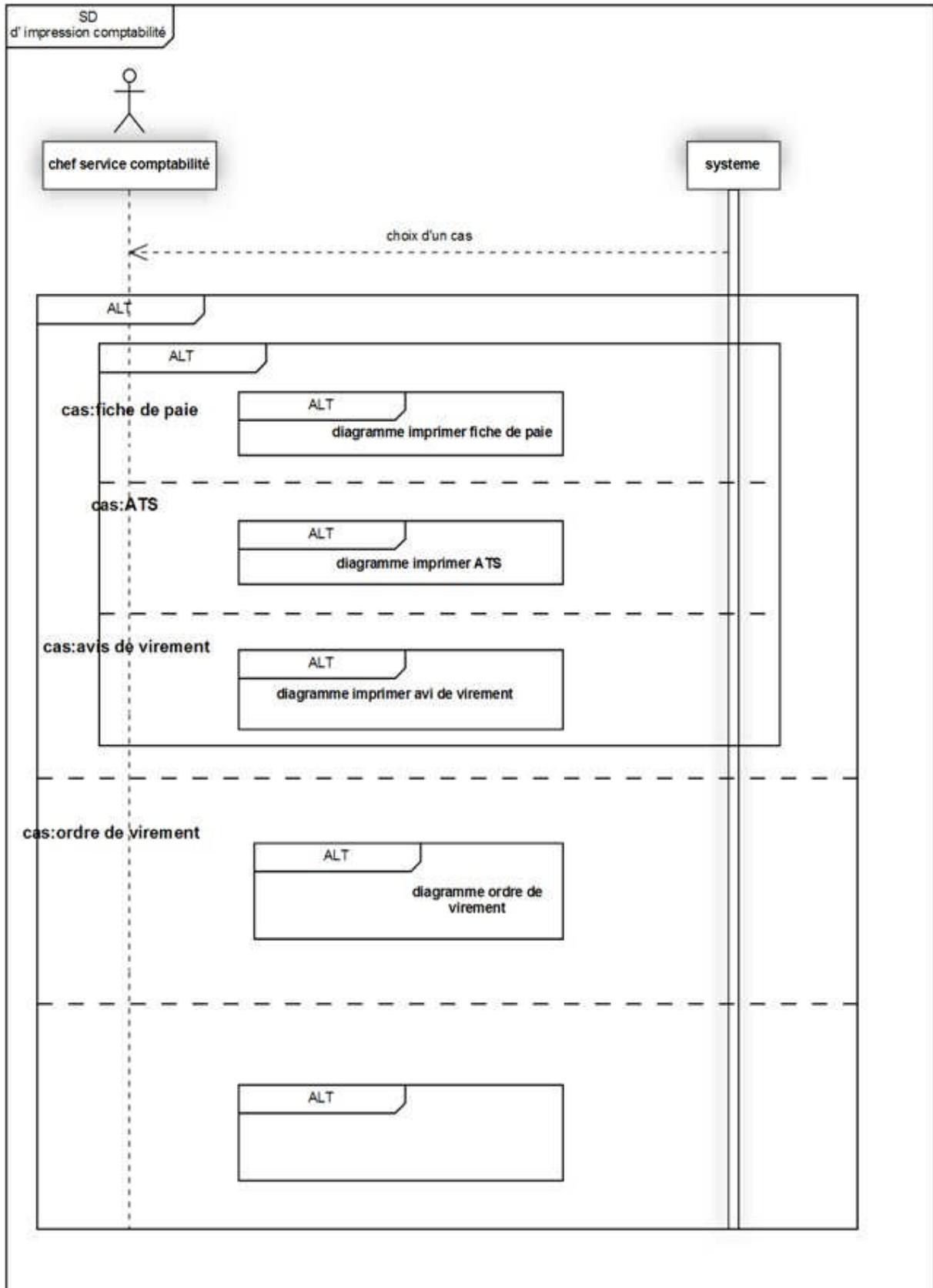
## 5.5 Paie



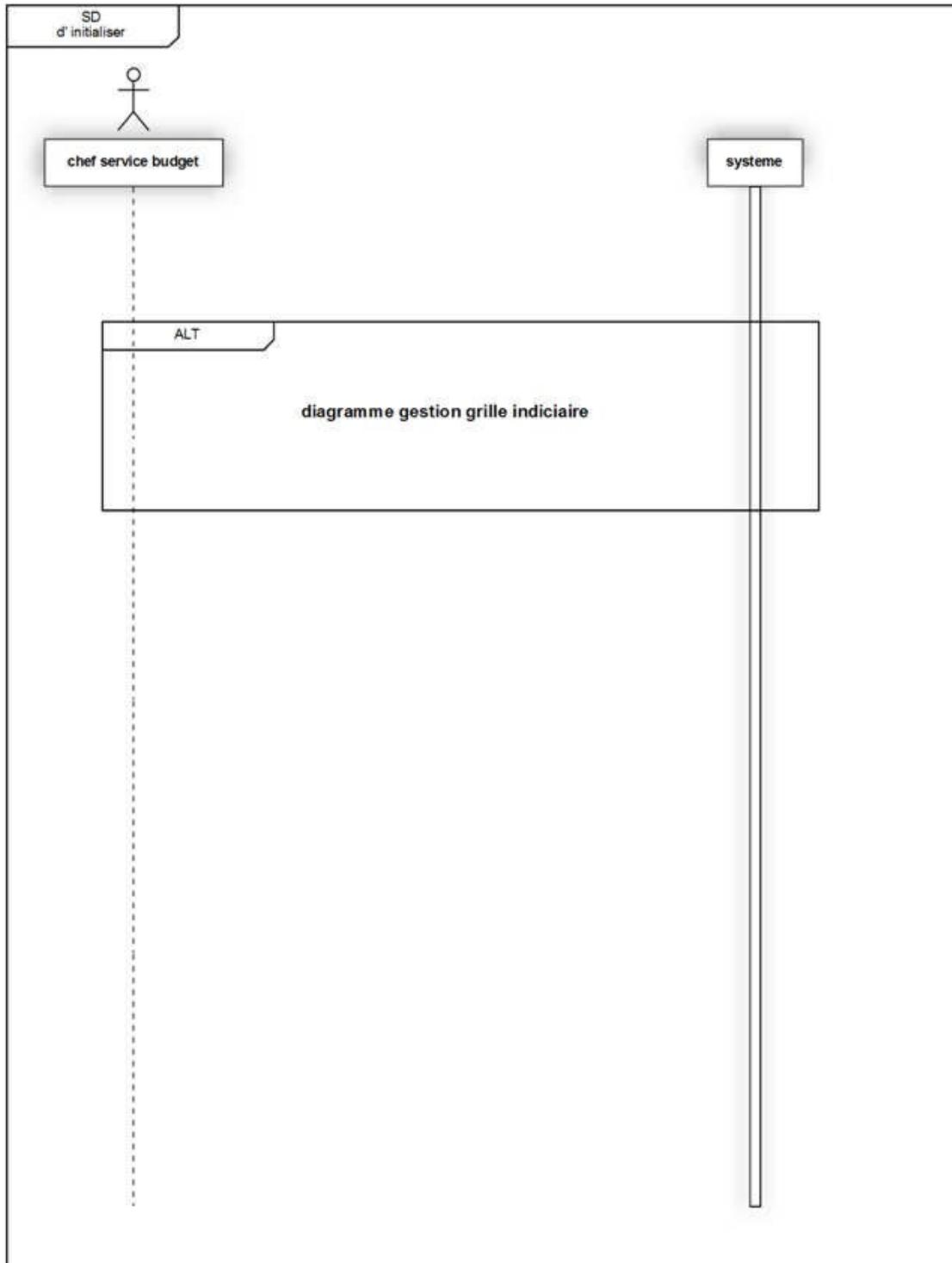
### 5.6 Impression budget



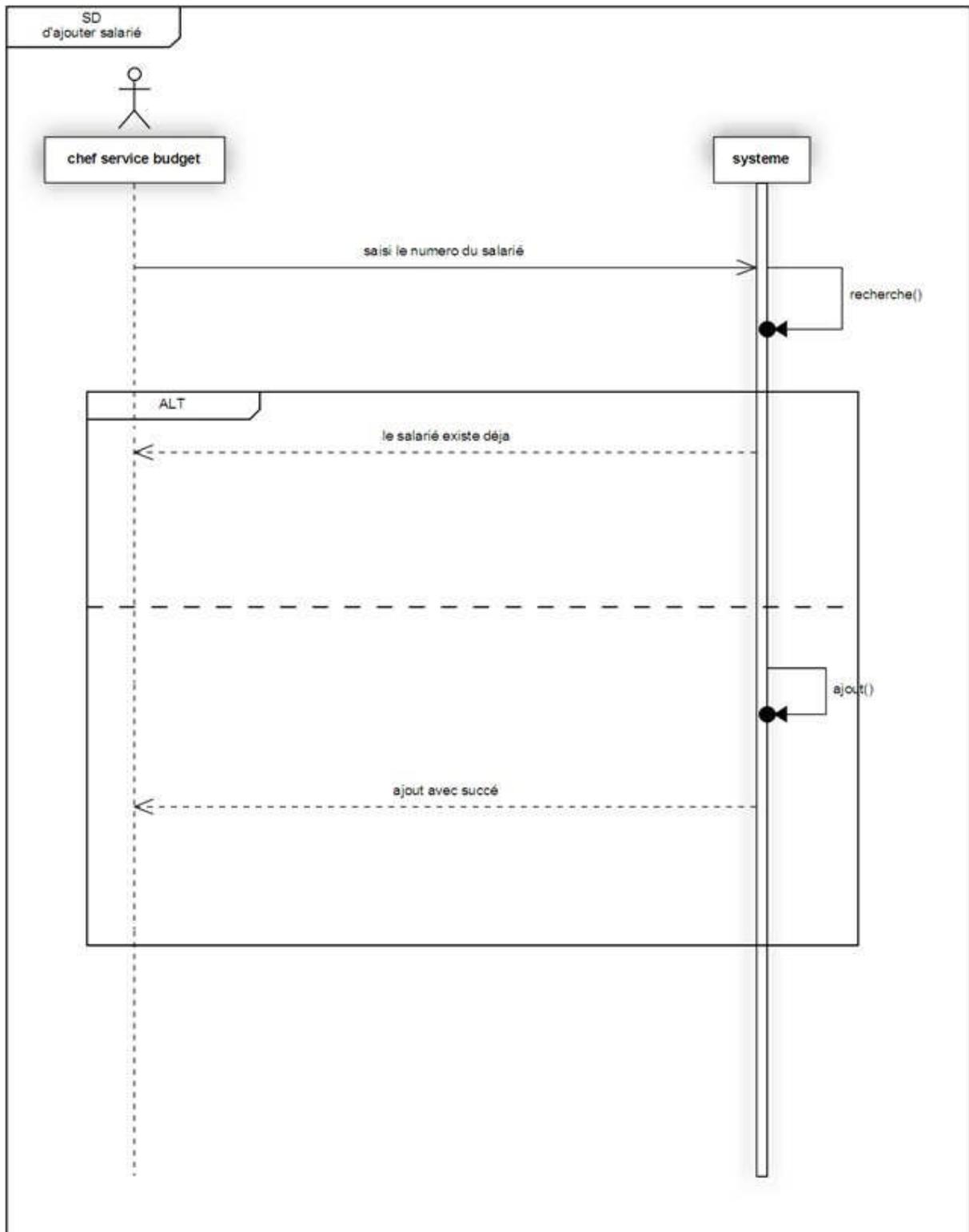
### 5.7 Impression comptabilité



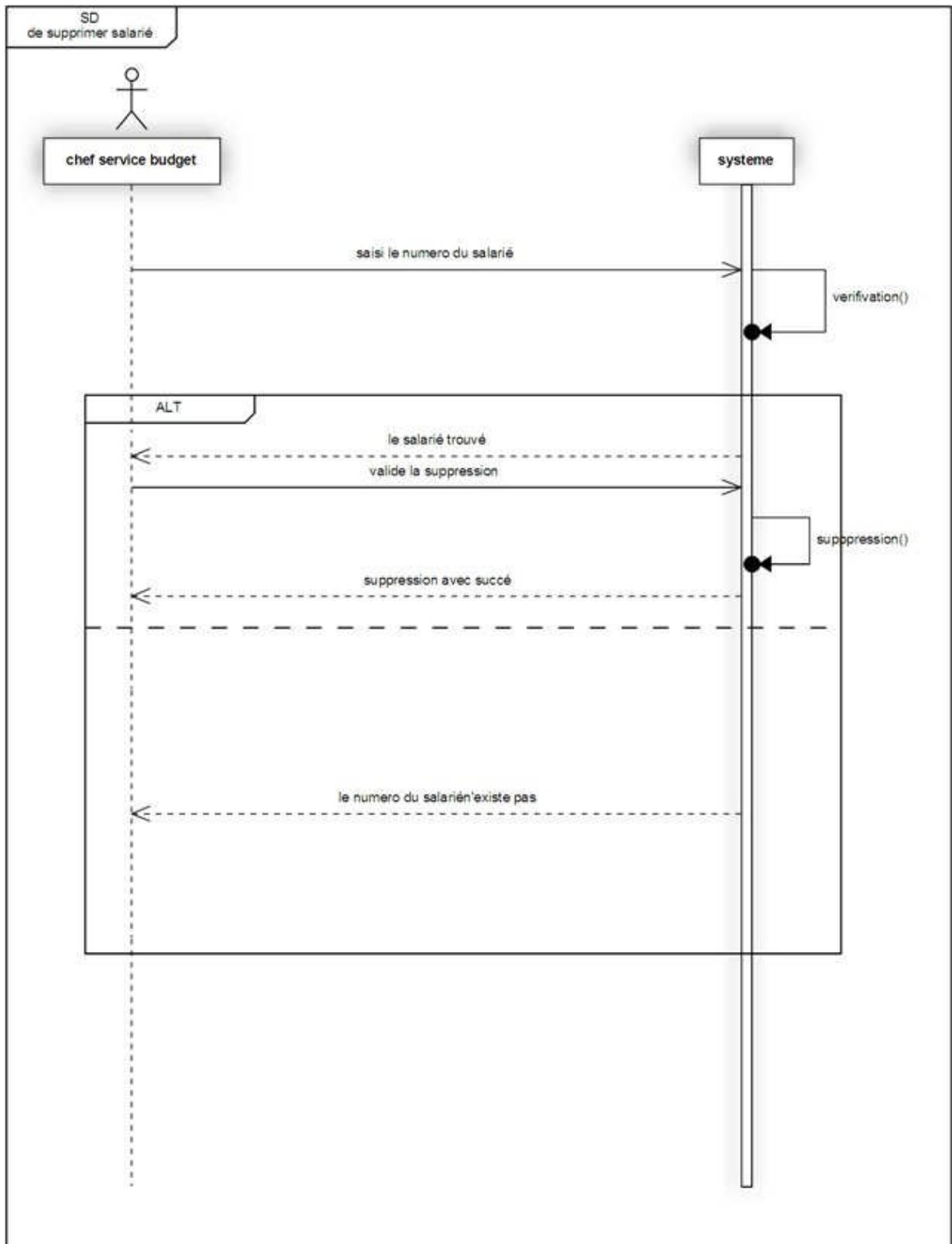
## 5.8 Initialisation



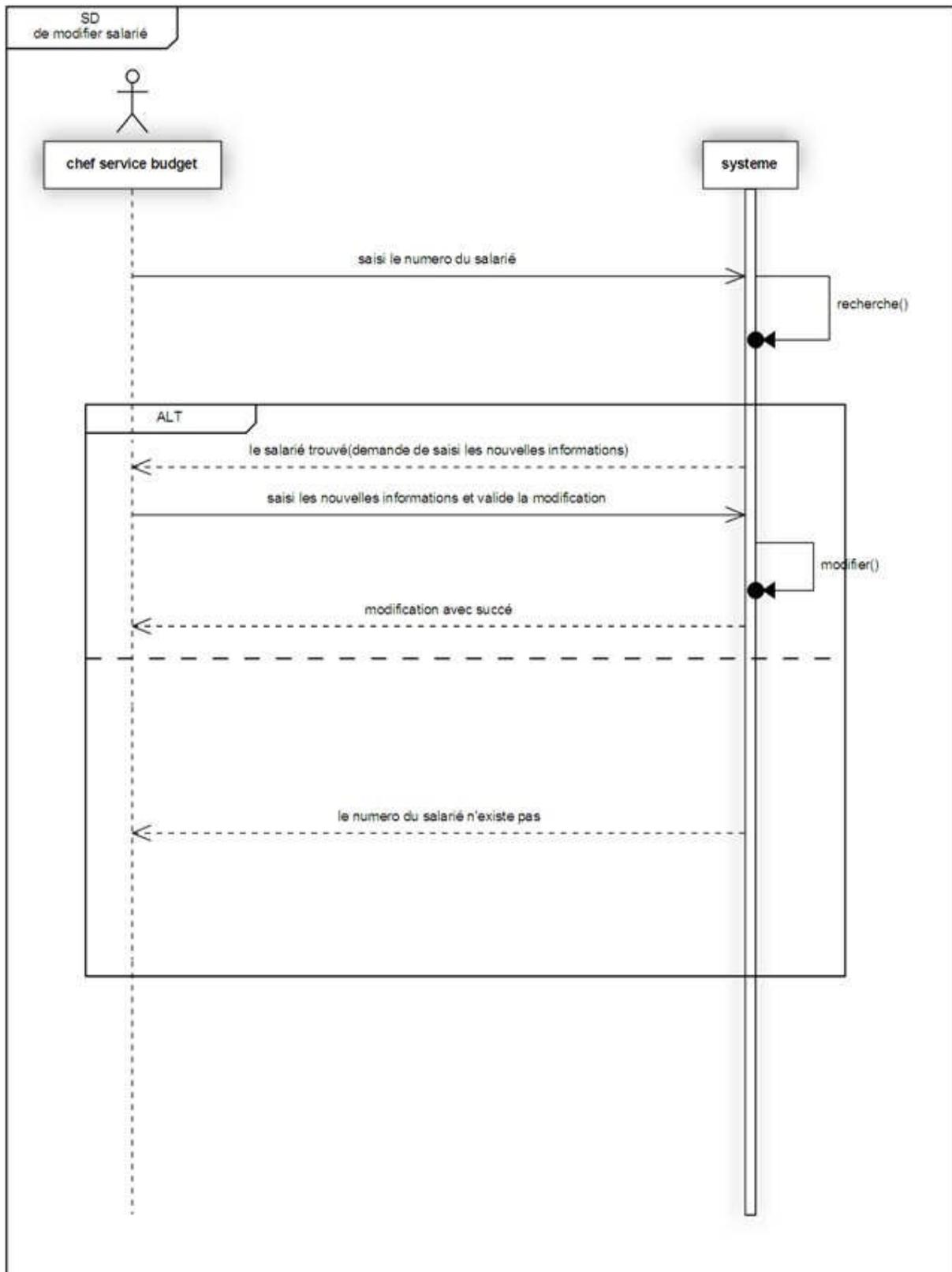
### 5.9 Ajouter salarié



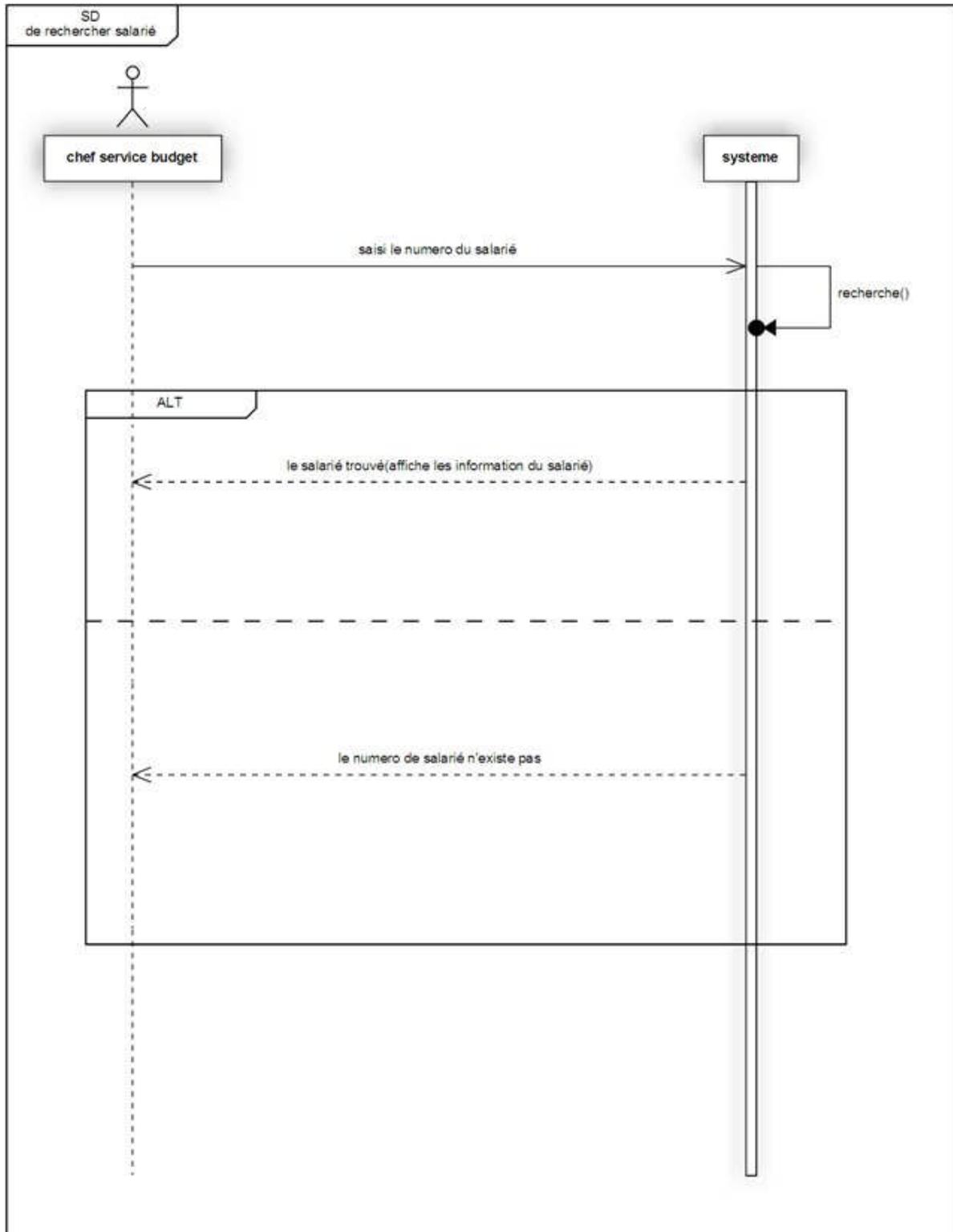
5.10. Supprimer salarié



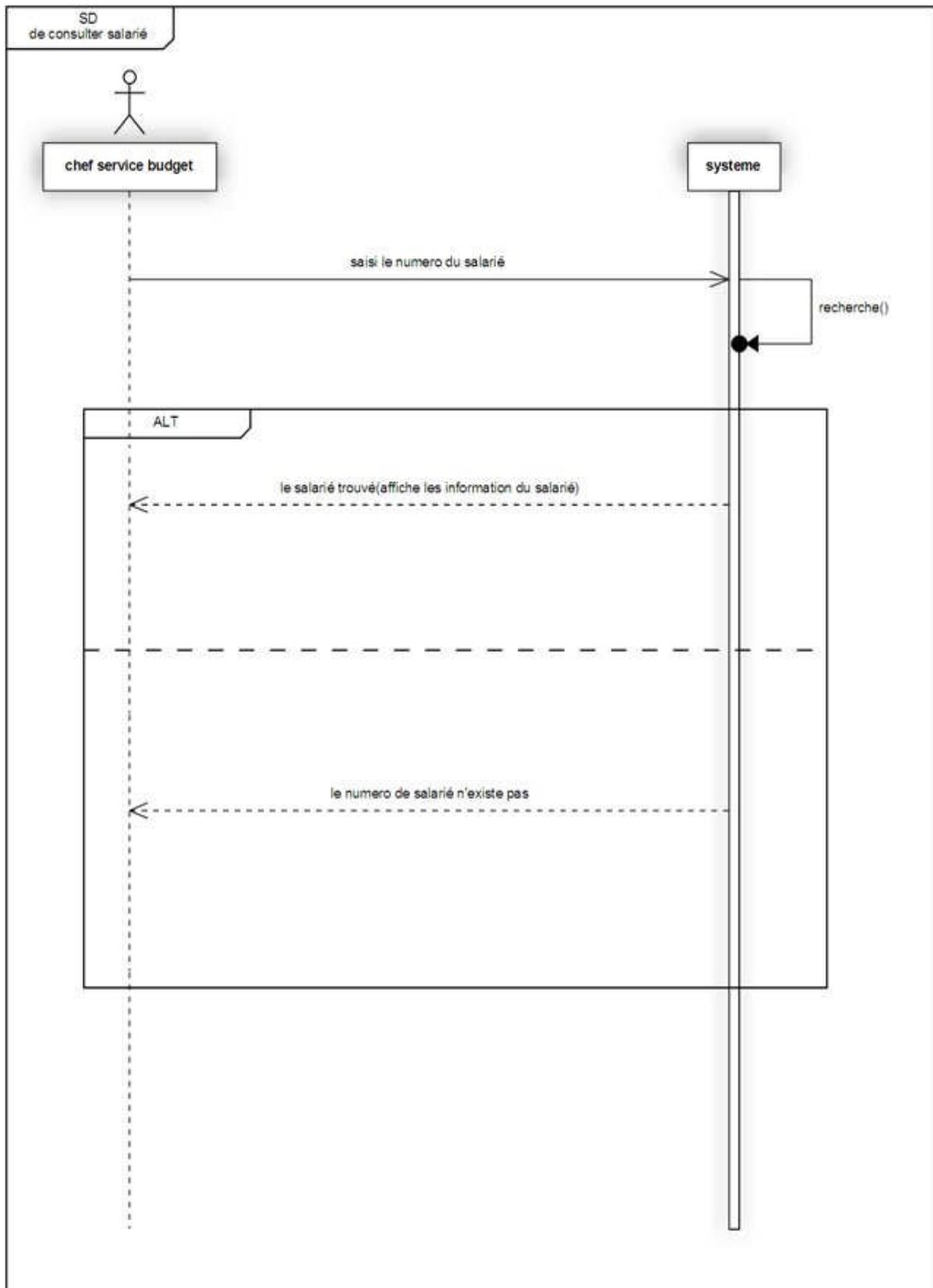
### 5.11. Modifier salarié



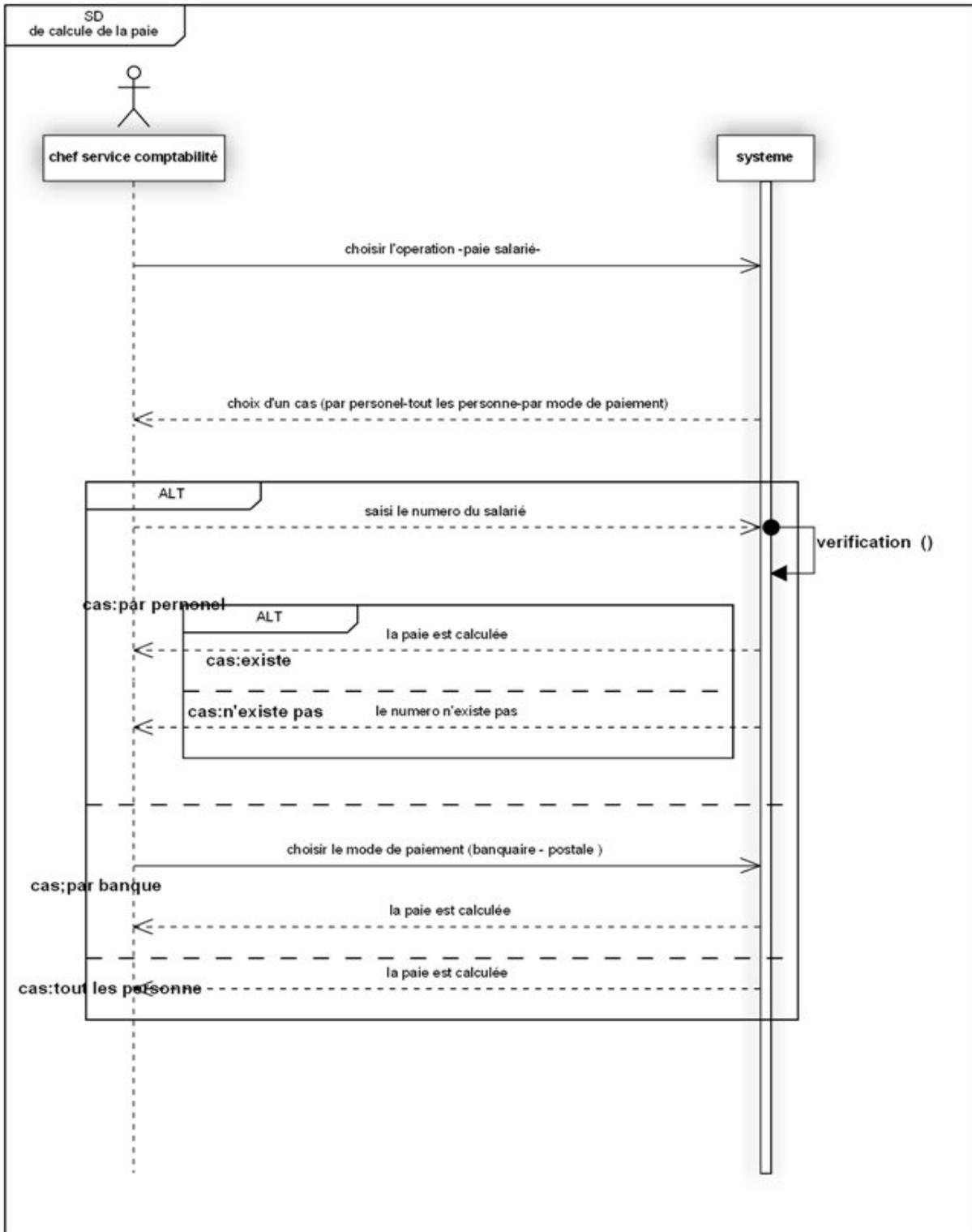
5.12. Rechercher salarié



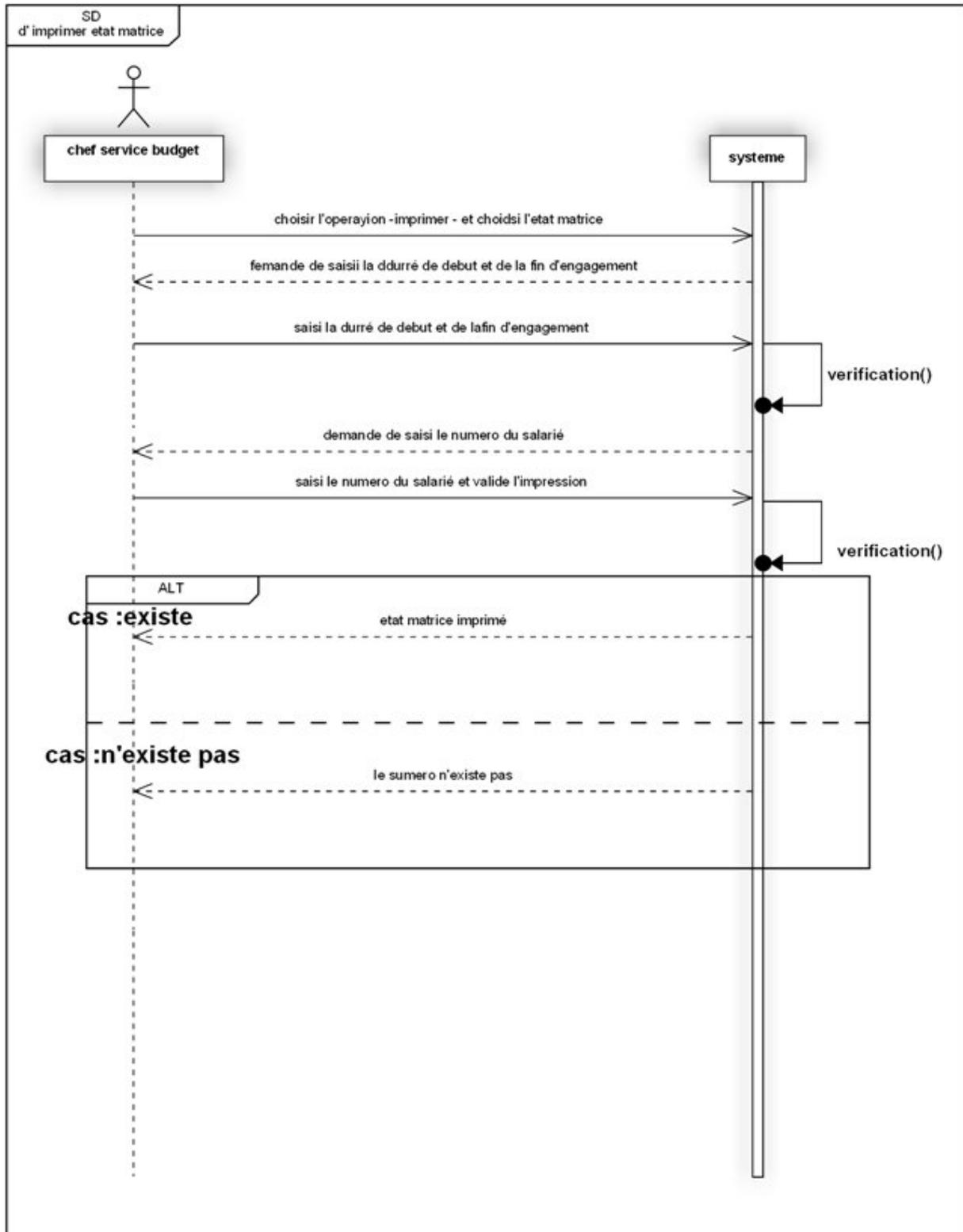
5.13. Consulter salarié



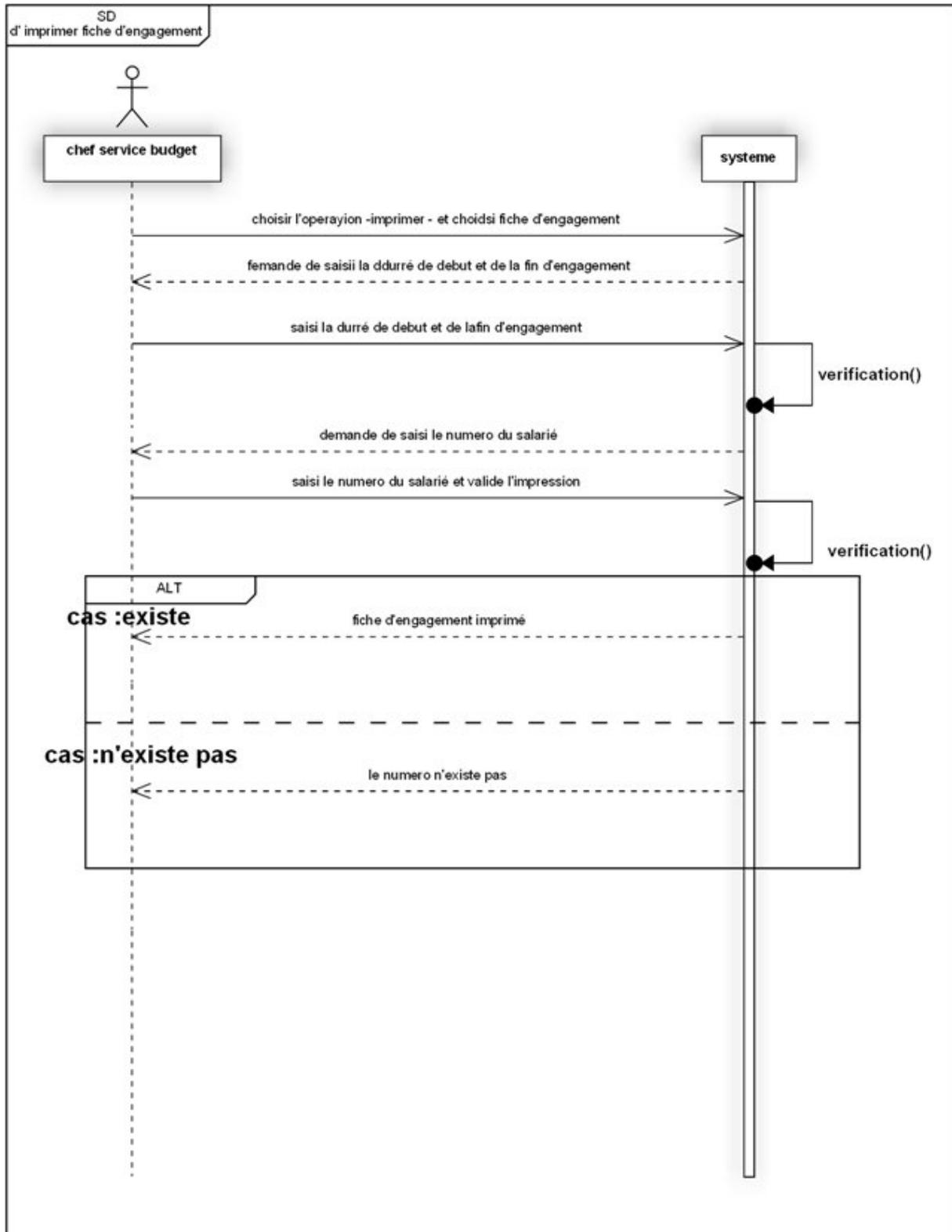
5.14. Calcul de la paie



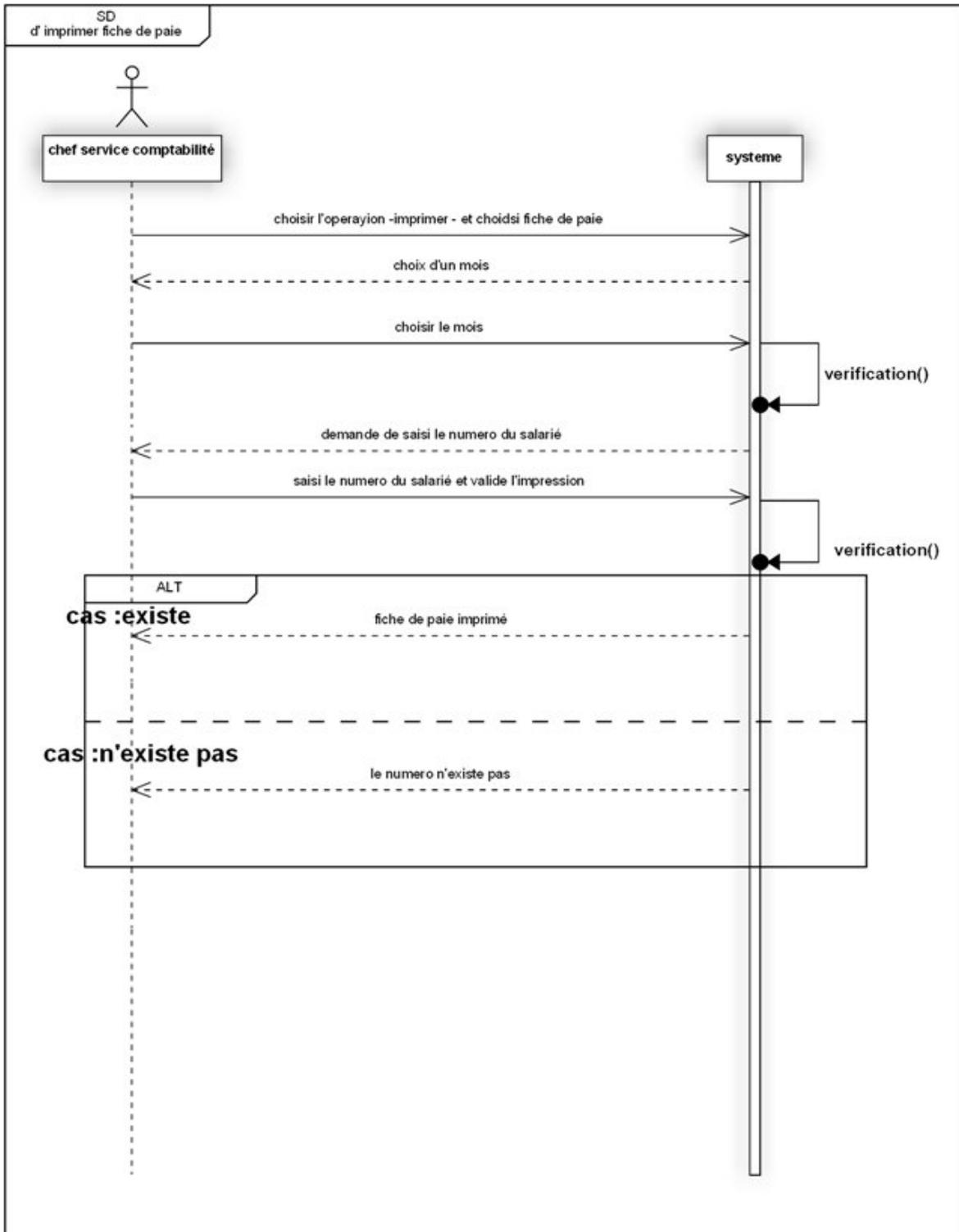
5.15. Imprimer état matrice



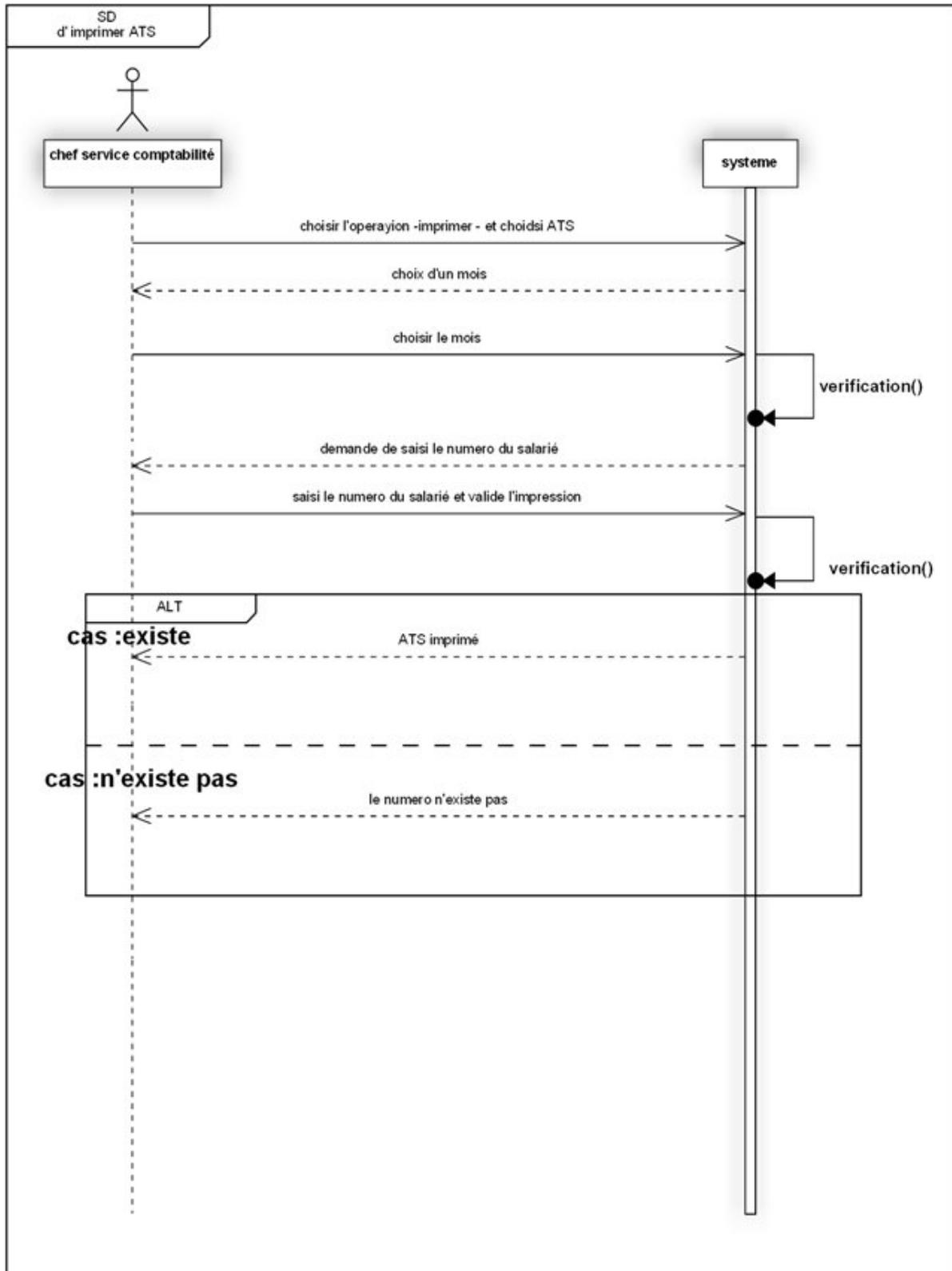
5.16. Imprimer fiche d'engagement



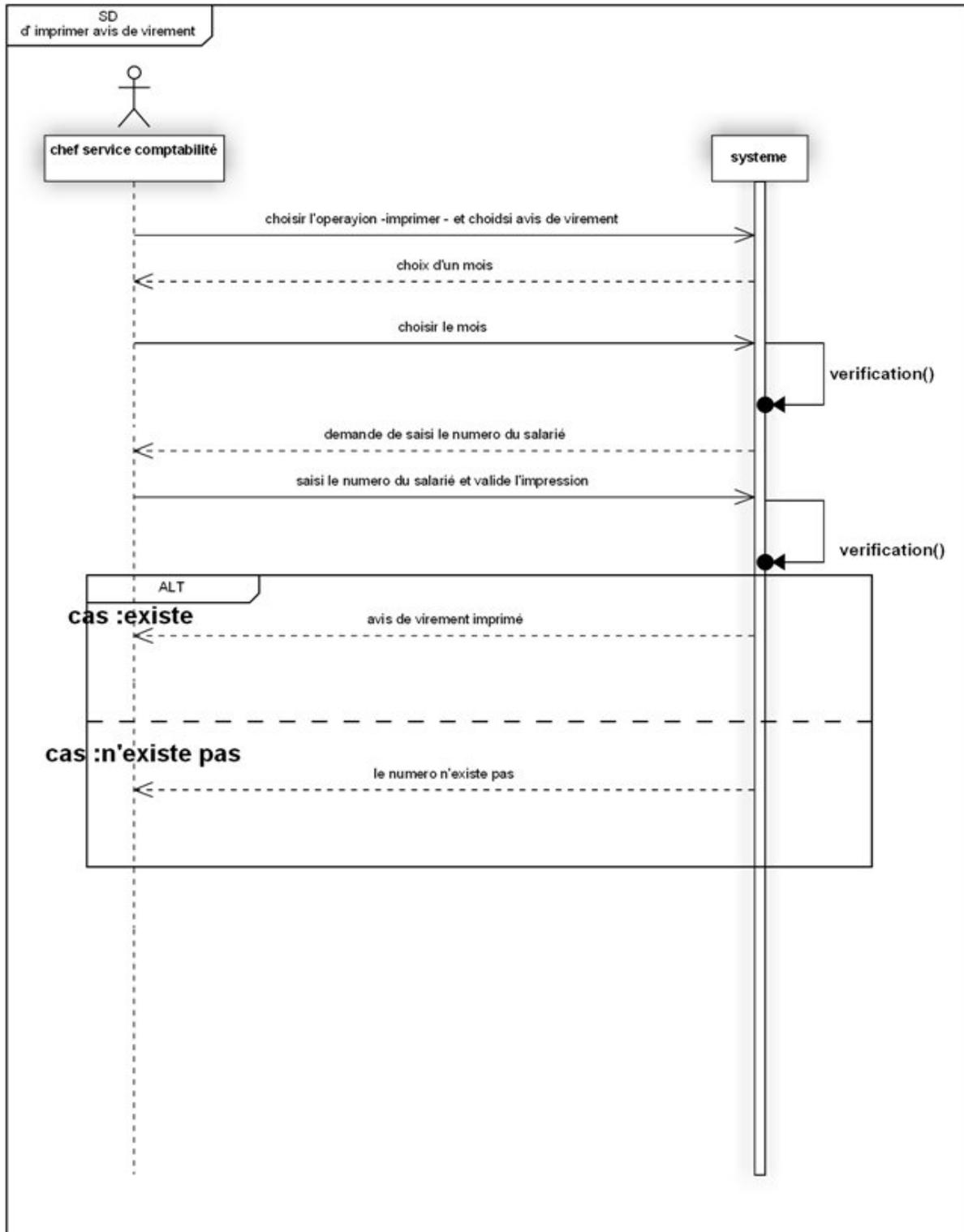
5.17. Imprimer fiche de paie



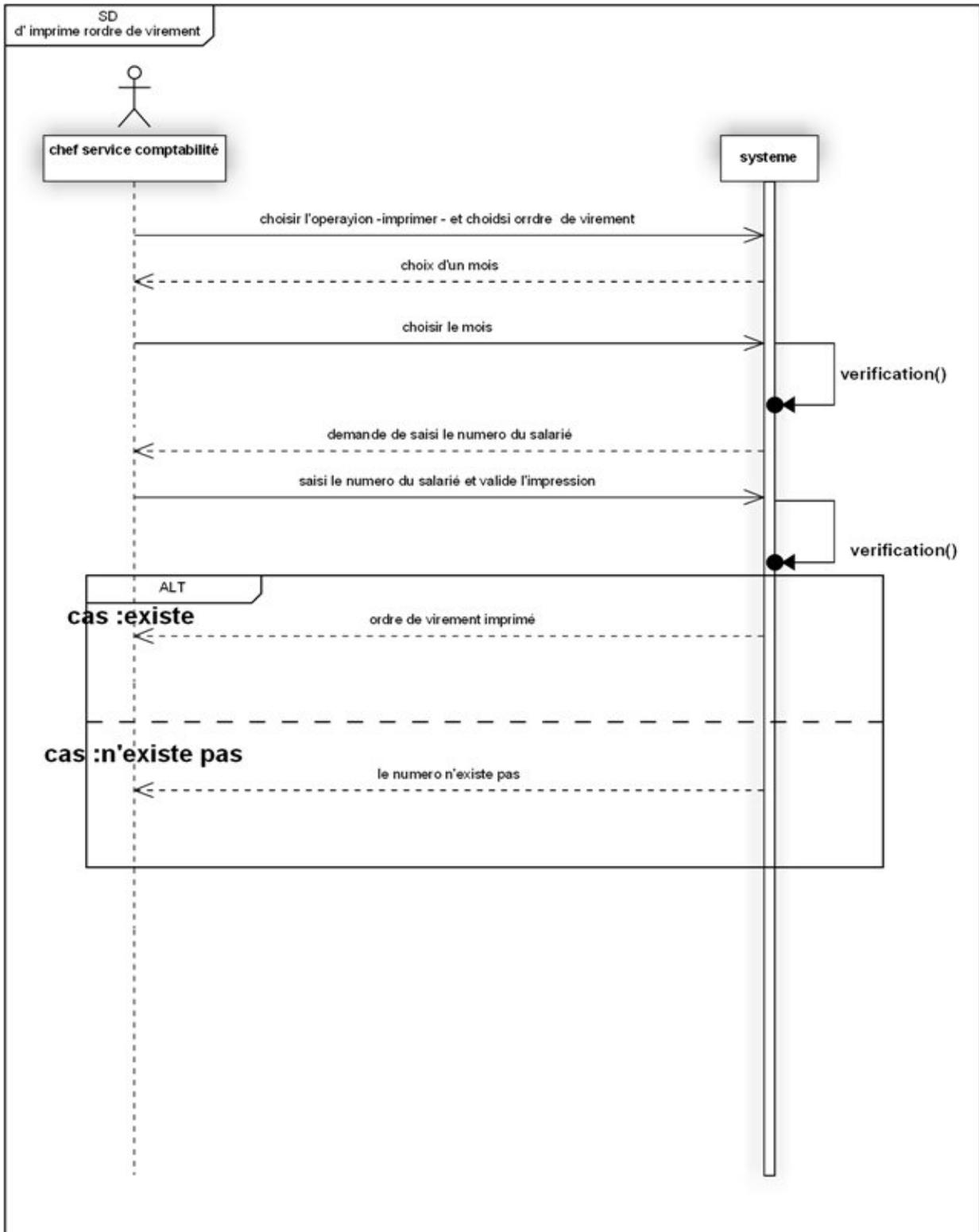
5.18. Imprimer ATS



5.19. Imprimer avis de virement



5.20. Ordre de virement



## *Chapitre 03 : phase d'analyse*

### *Introduction*

- 1. analyse de domaine*
- 2. notre modèle de domaine*
- 3. diagramme de classes  
participante*

## 1. analyse 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 appelé modèle du domaine. Ce modèle doit définir les classes qui modélisent les entités ou concepts présents dans le domaine (on utilise aussi le terme métier) de l'application.

### 1.1. Affectation des attributs aux classes

**Salarié** {num\_sal,nom\_s ,prenom\_s ,adresse\_s,date\_N,num\_s\_s,grade,sexe,date\_rct,  
situation\_f}

**grille** {num\_case\_grl, catégorie, echelon,groupe,val\_ind}

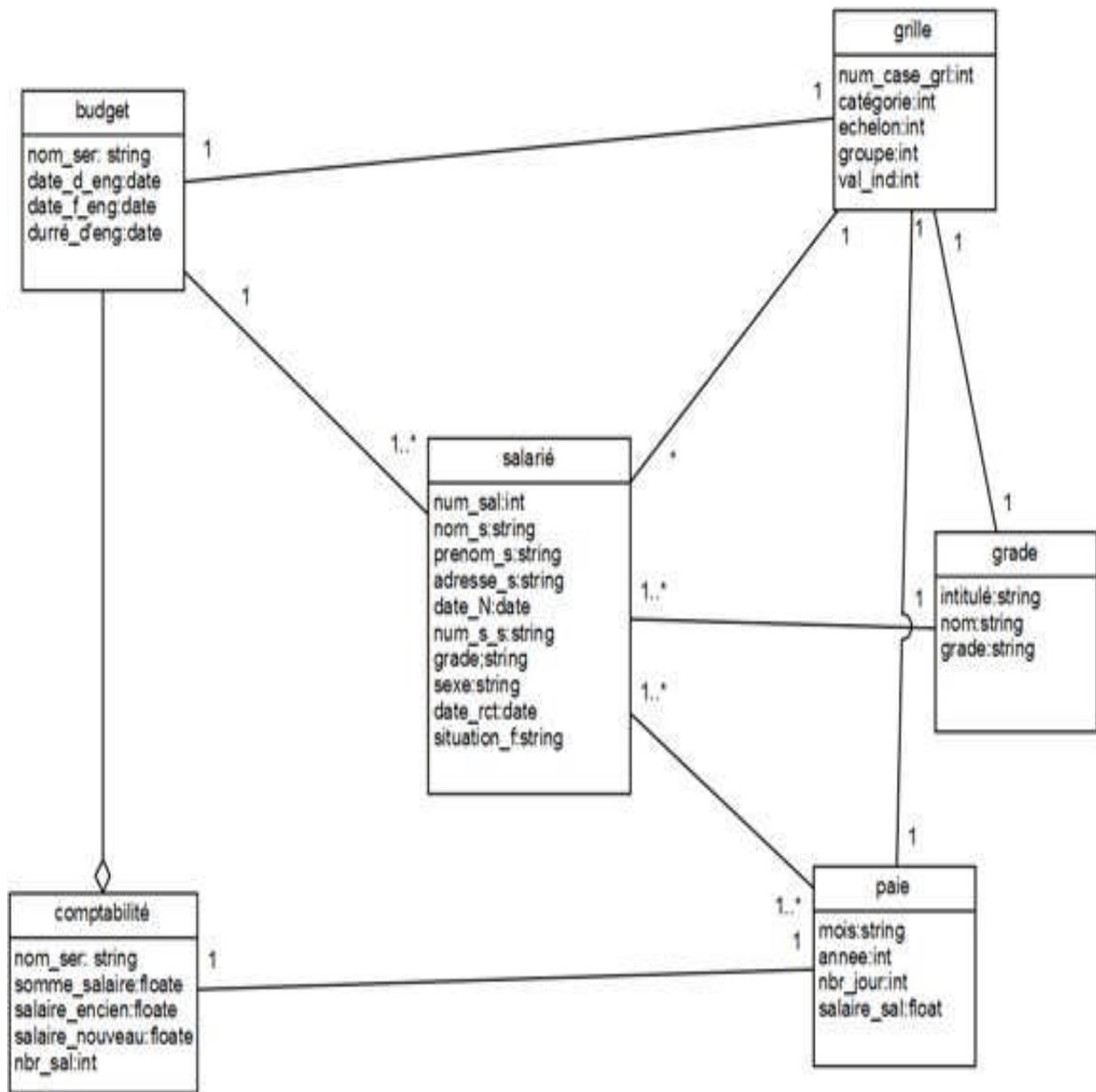
**grade** {intitulé,nom,grade }

**paie** {mois,annee,nbr\_jour,salaire\_sal}

**Budget** {nom\_ser,durre\_d'eng,date\_d\_eng,date\_f\_eng}

**comptabilité** {nom\_ser,somme\_salaire,nbr\_sal,salaire\_encien,salaire\_nouveau}

## 2. Modèle de domaine



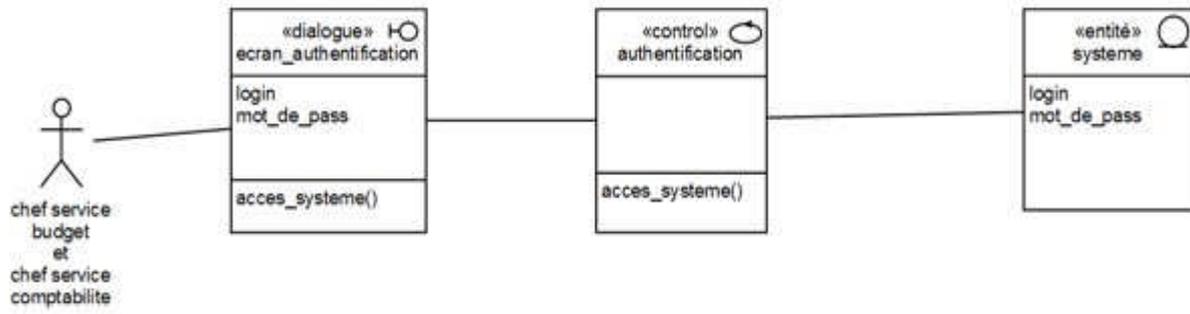
### **3. Diagramme de classe participante**

Le diagramme de classes participantes est particulièrement important puisqu'il effectue la jonction entre les classes de cas d'utilisation, et du domaine.

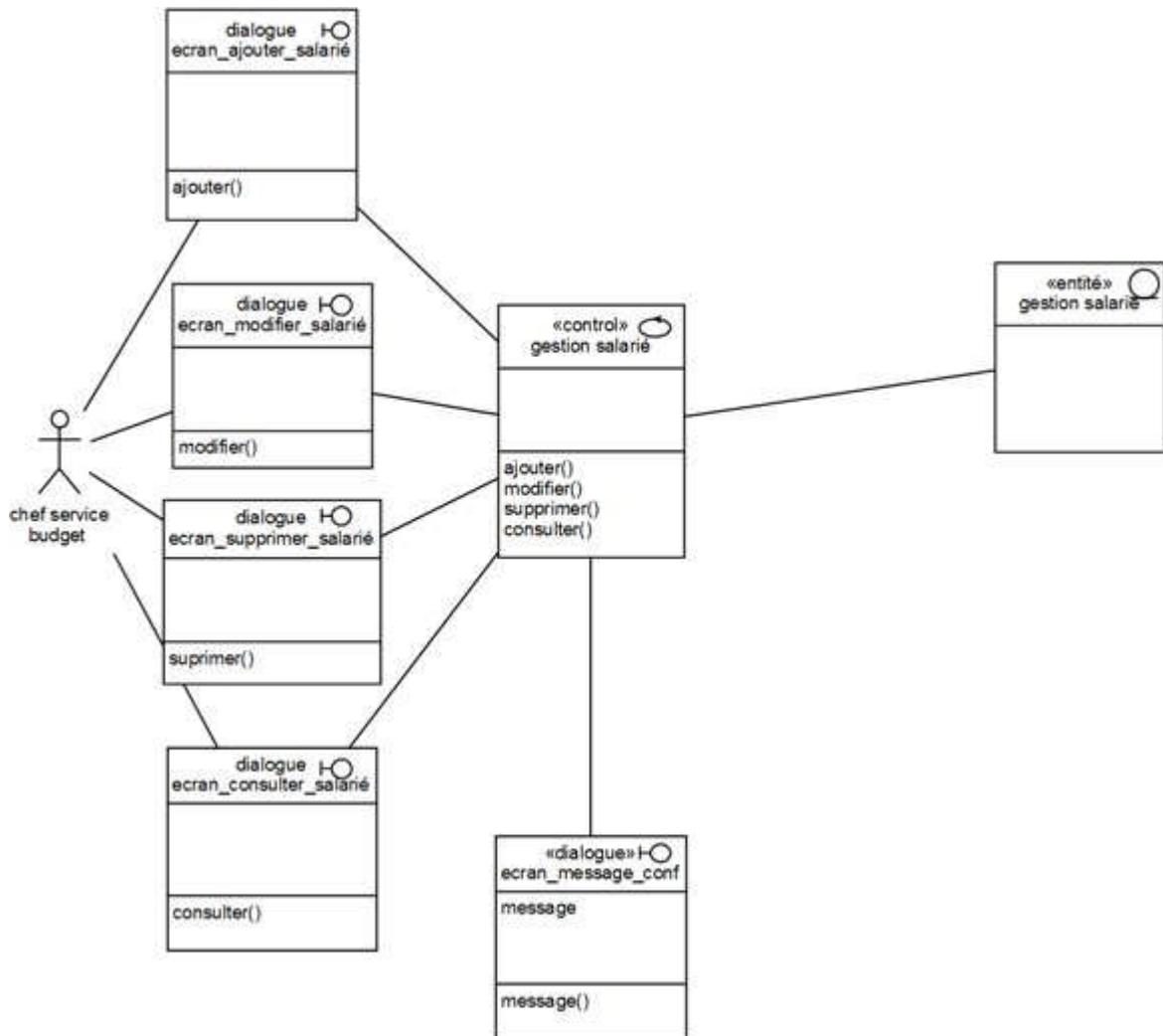
Il n'est pas souhaitable que les utilisateurs interagissent directement avec les instances des classes des domaines par le biais de l'interface graphique. En effet le modèle de domaine doit être indépendant des utilisateurs et de l'interface graphique. De même, l'interface graphique du logiciel doit pouvoir évaluer sans répercussion sur le cœur de l'application.

#### 4.1. S'authentifier

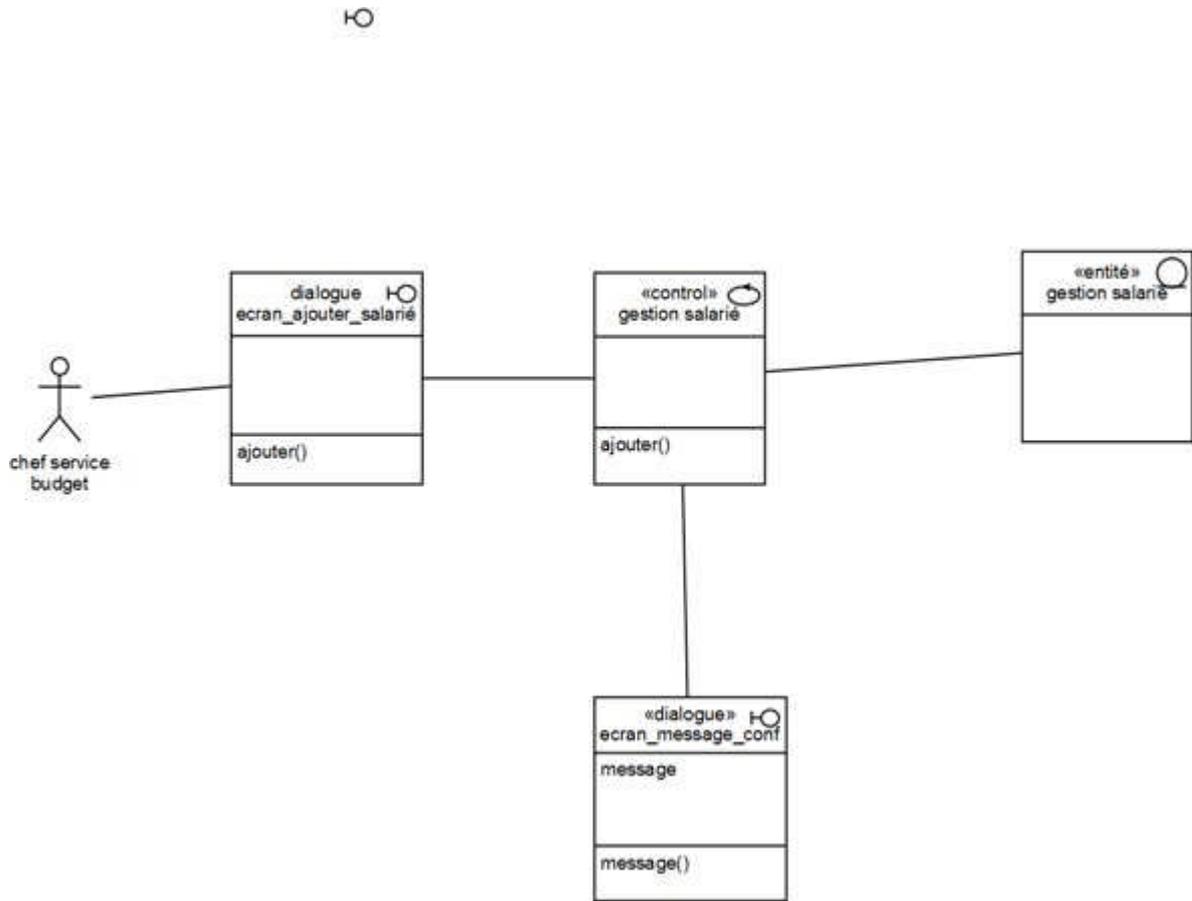
IO



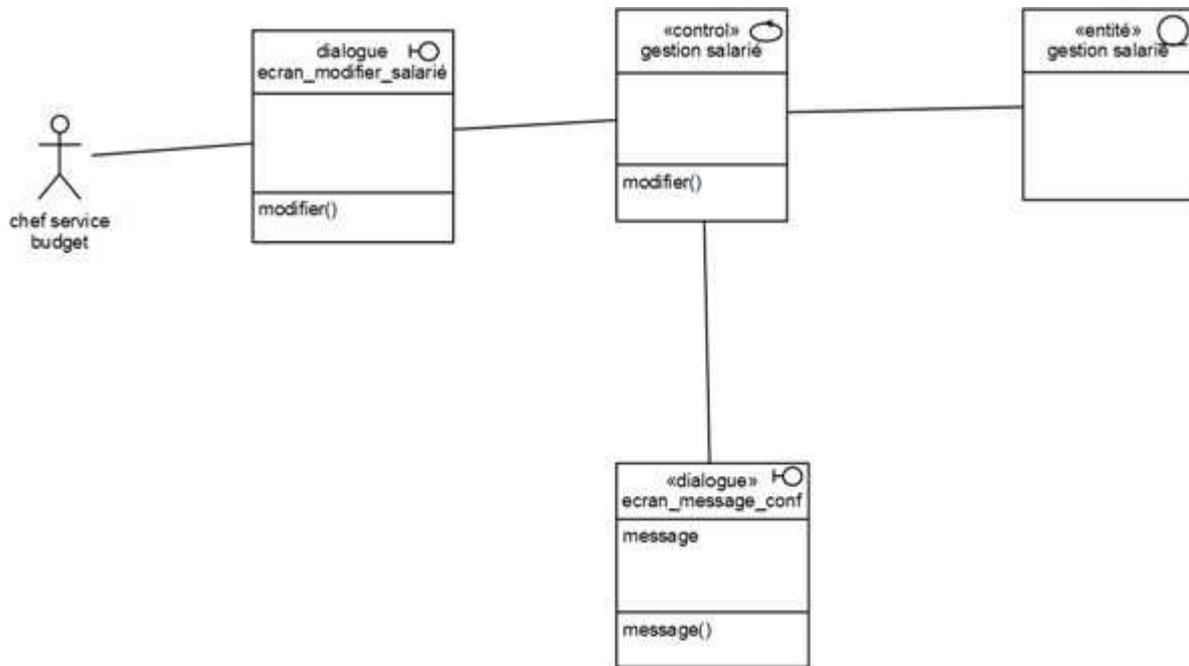
4.2. Gestion salarié



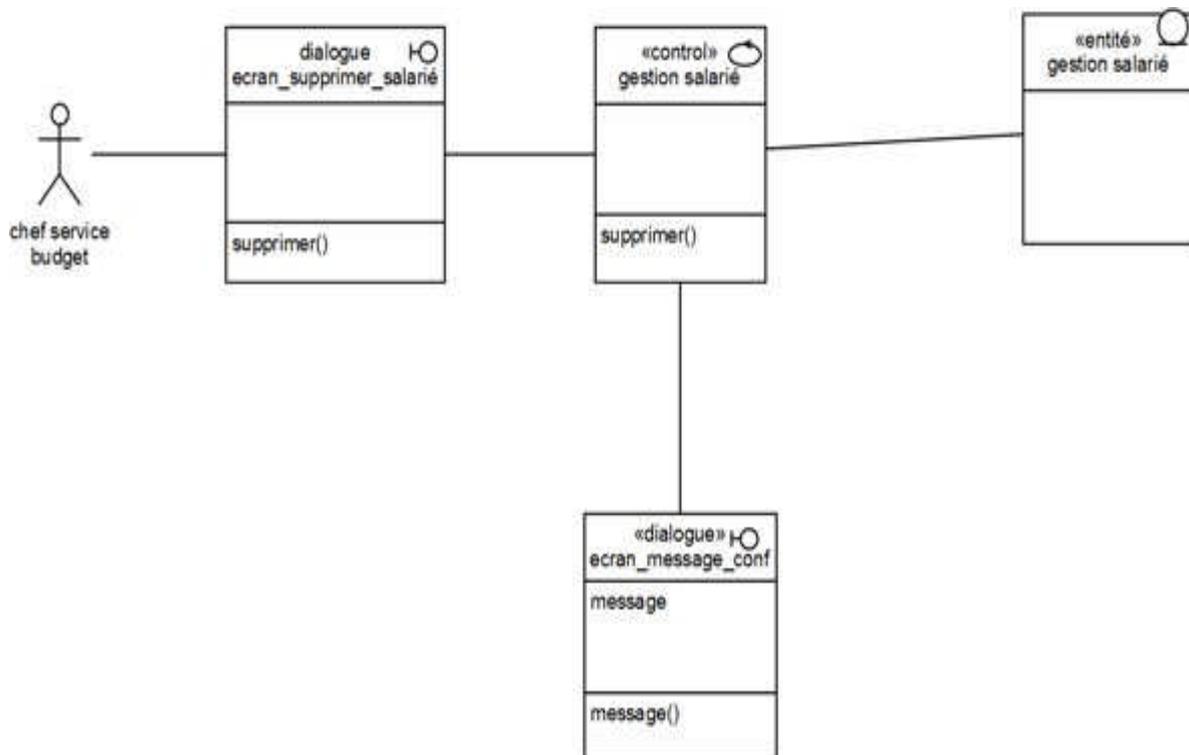
4.3. Ajouter salarié



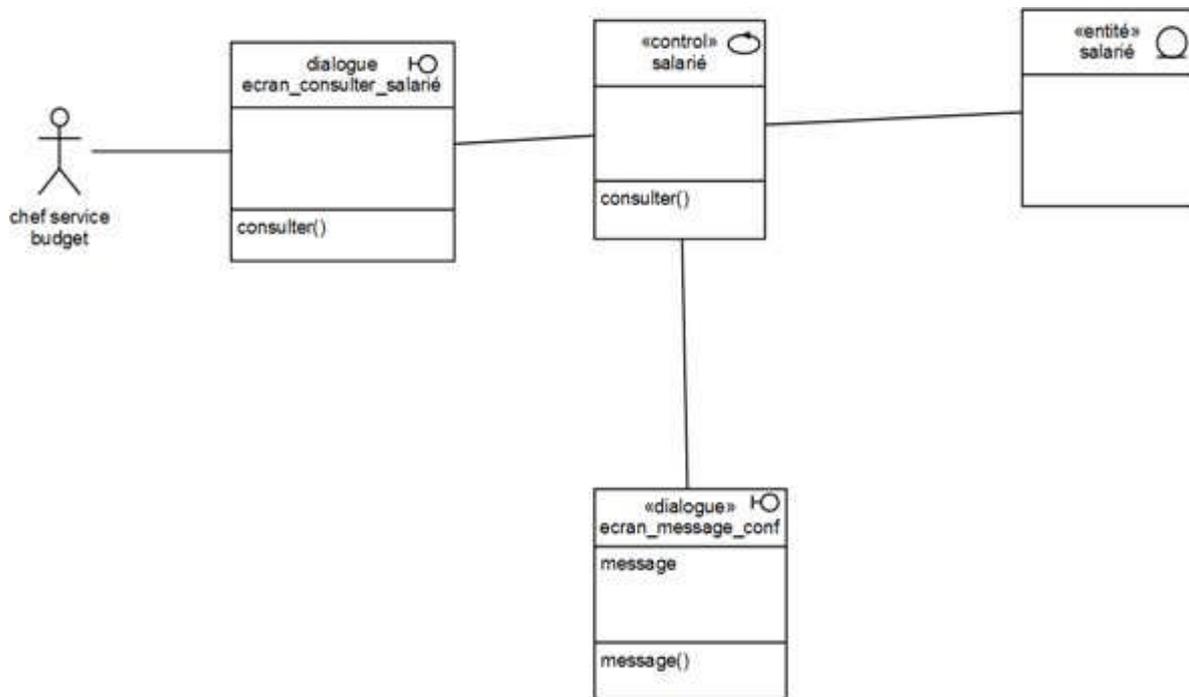
4.4. Modifier salarié



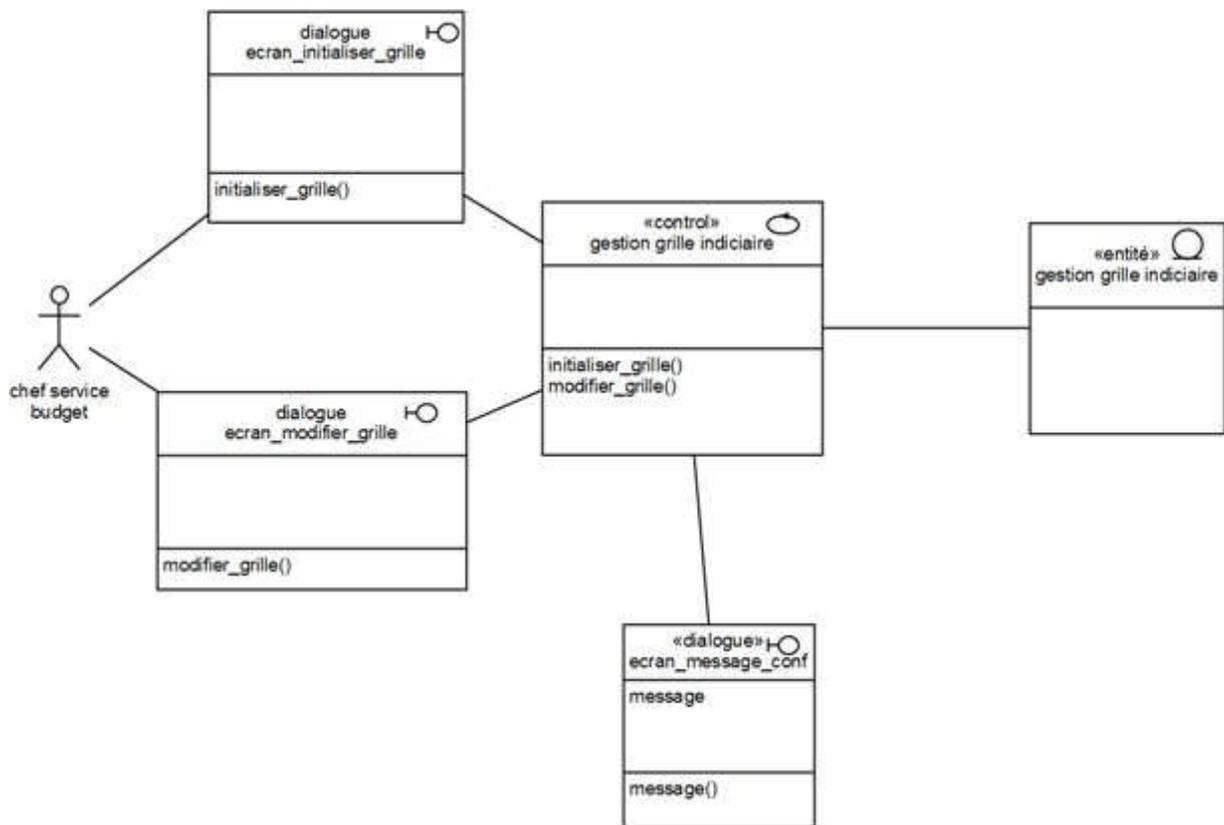
4.5. Supprimer salarié



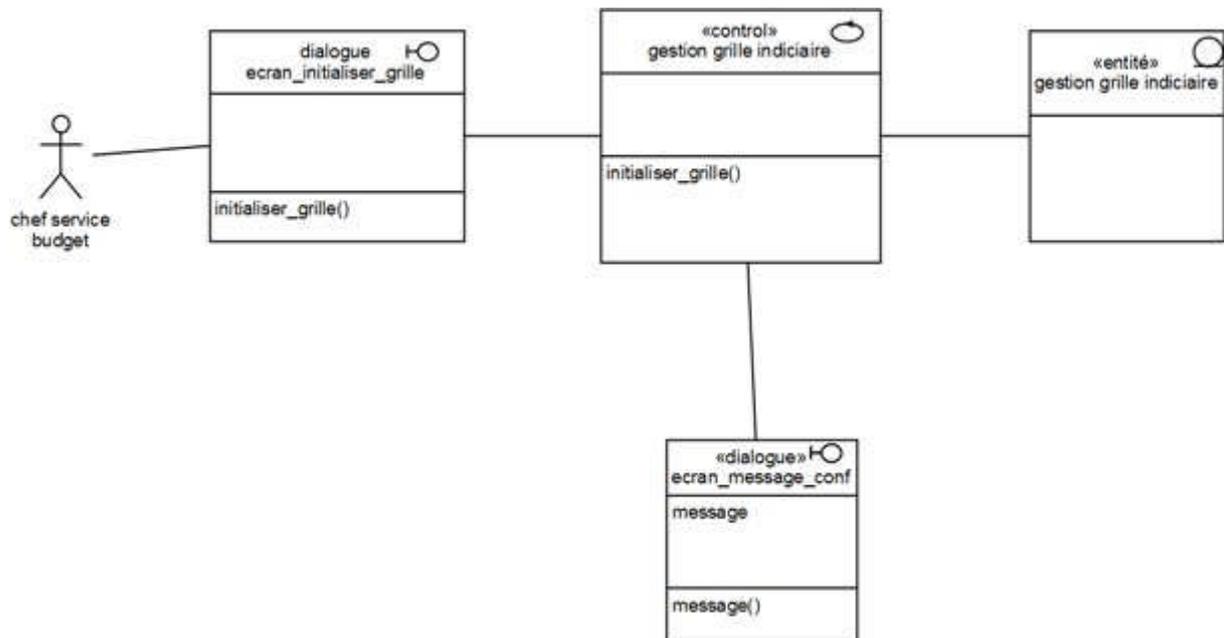
4.6. Consulter salarié



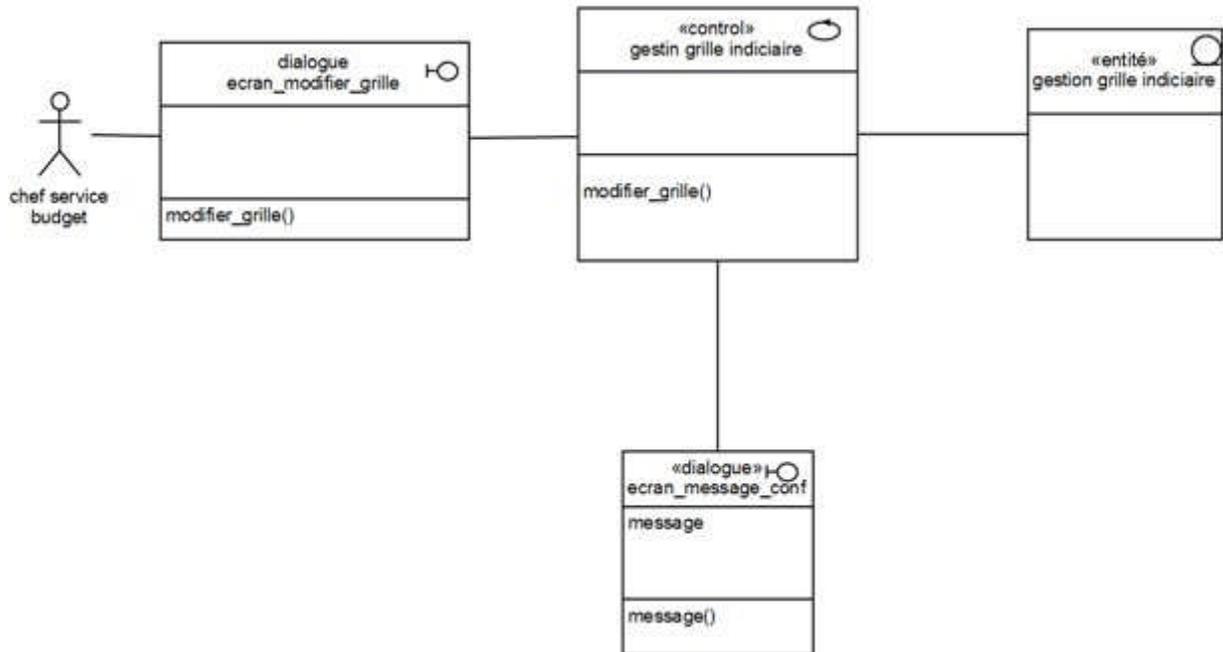
4.7. Gestion grille indiciaire



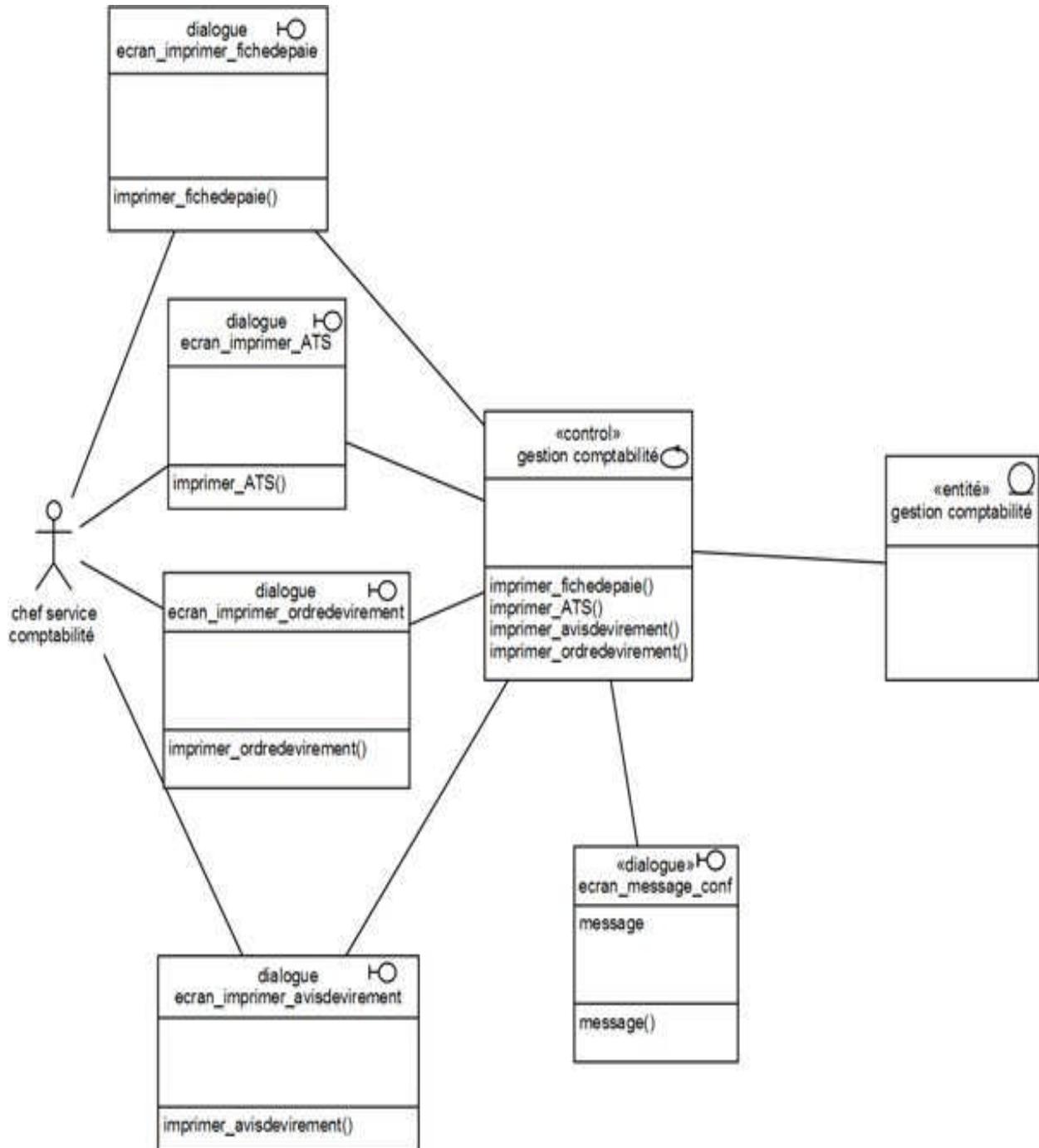
4.8. Initialiser grille indiciaire



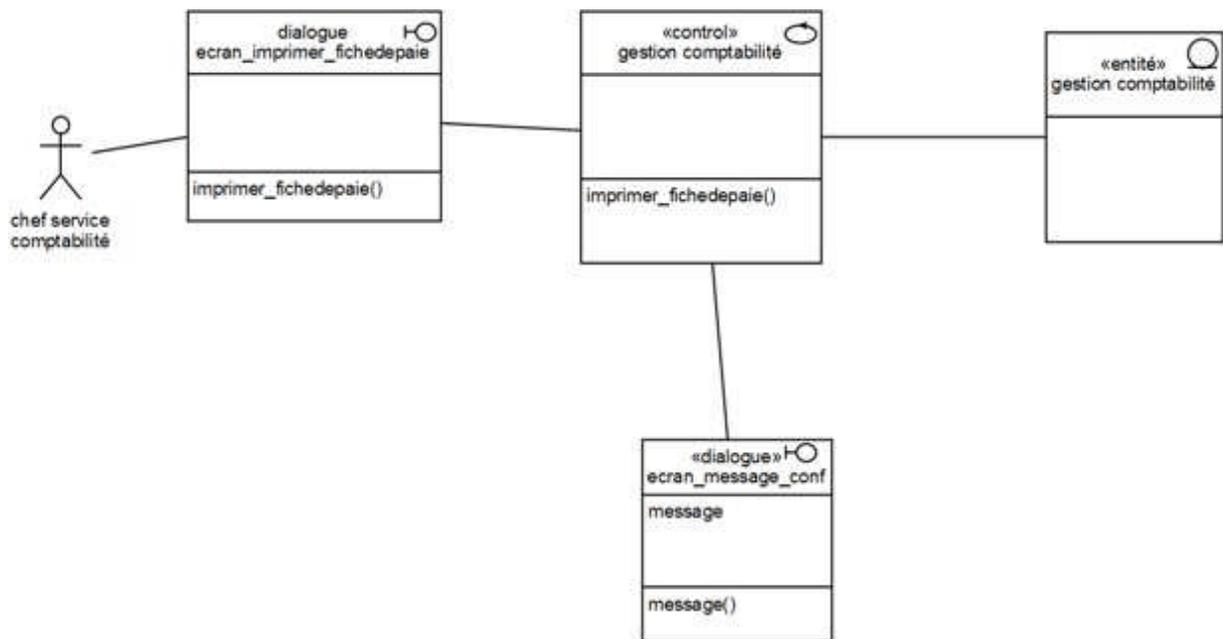
4.9. Modifier grille indiciaire



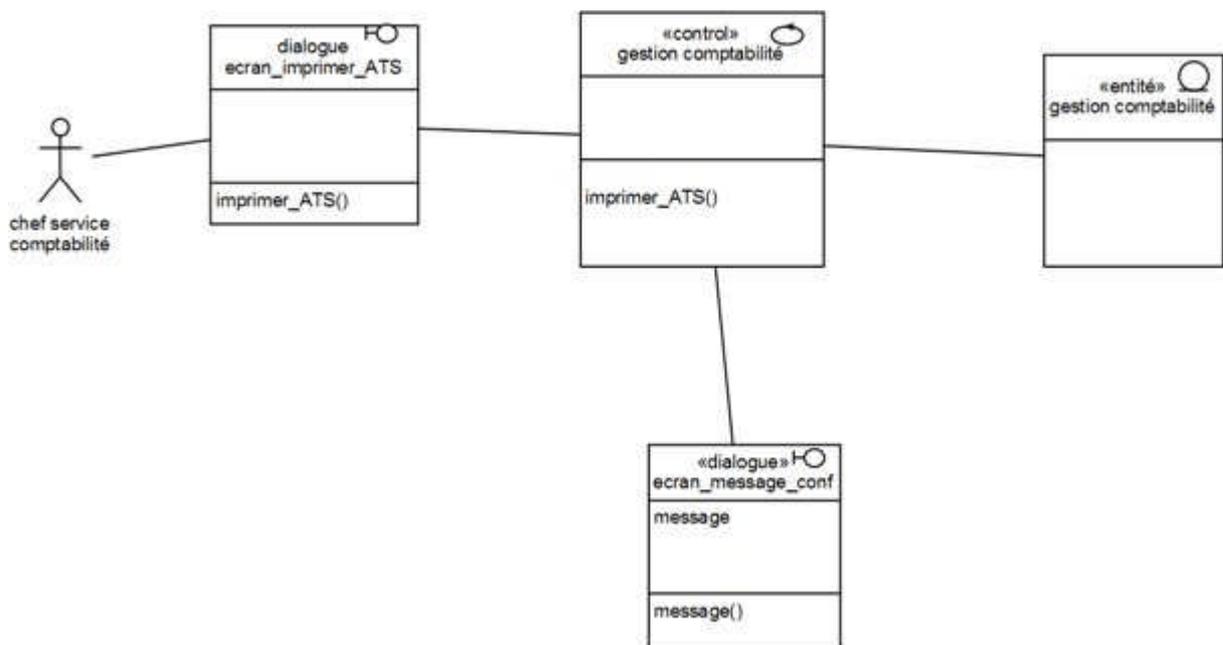
4.10. Gestion comptabilité



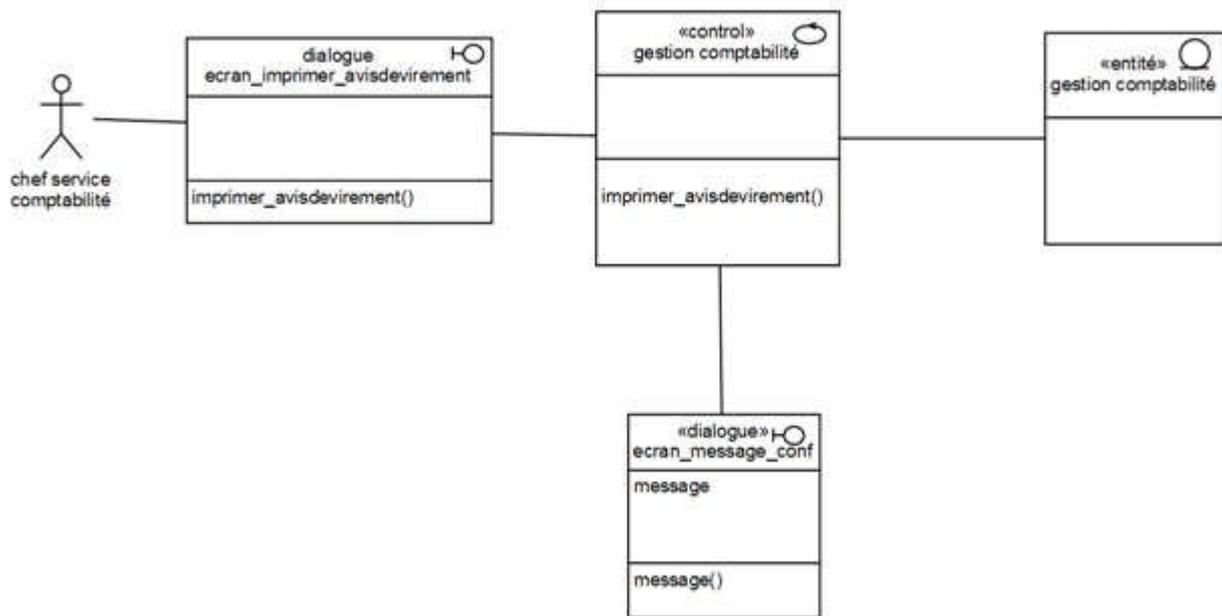
4.11. [Imprimer fiche de paie](#)



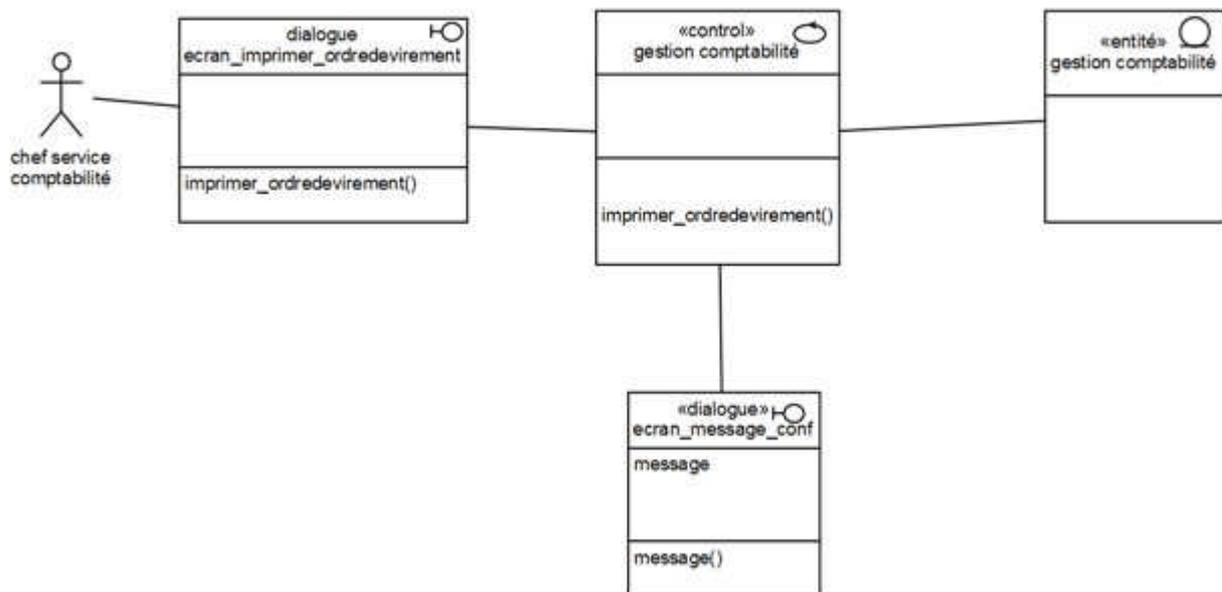
4.12. [Imprimer ATS](#)



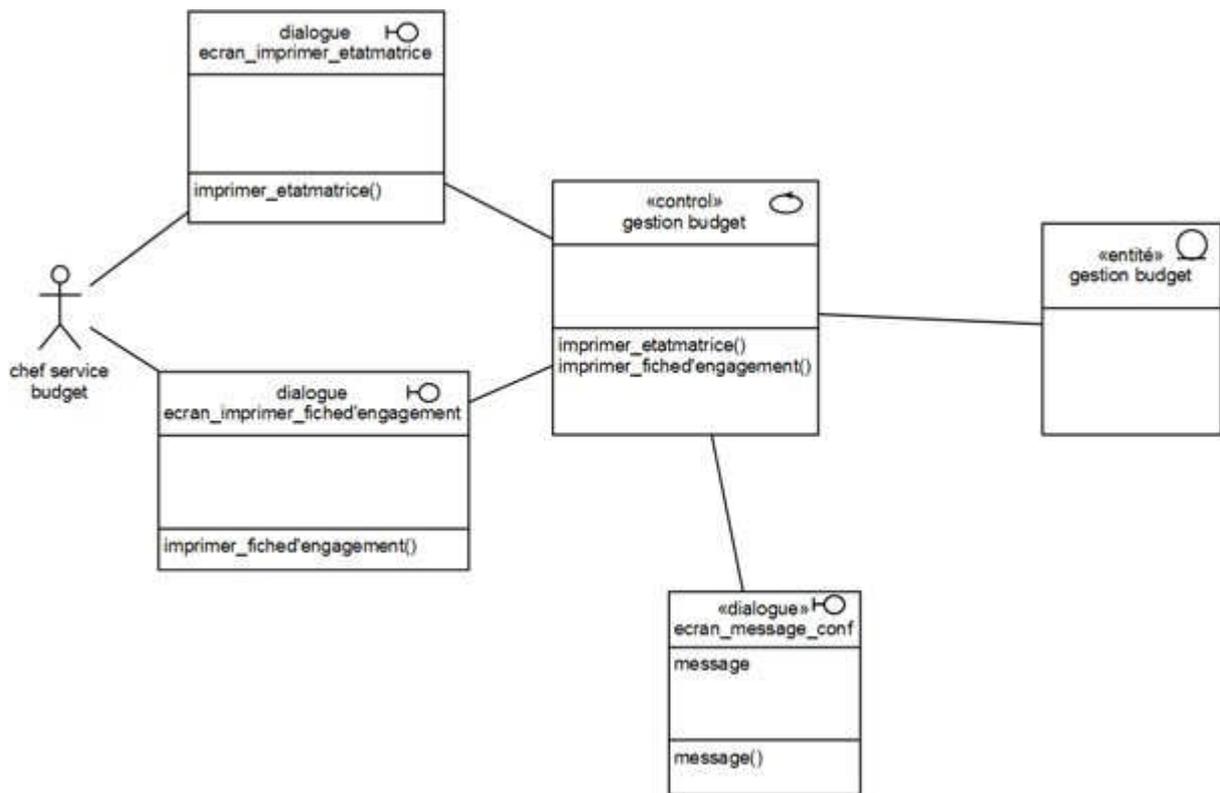
4.13. [Imprimer avis de virement](#)



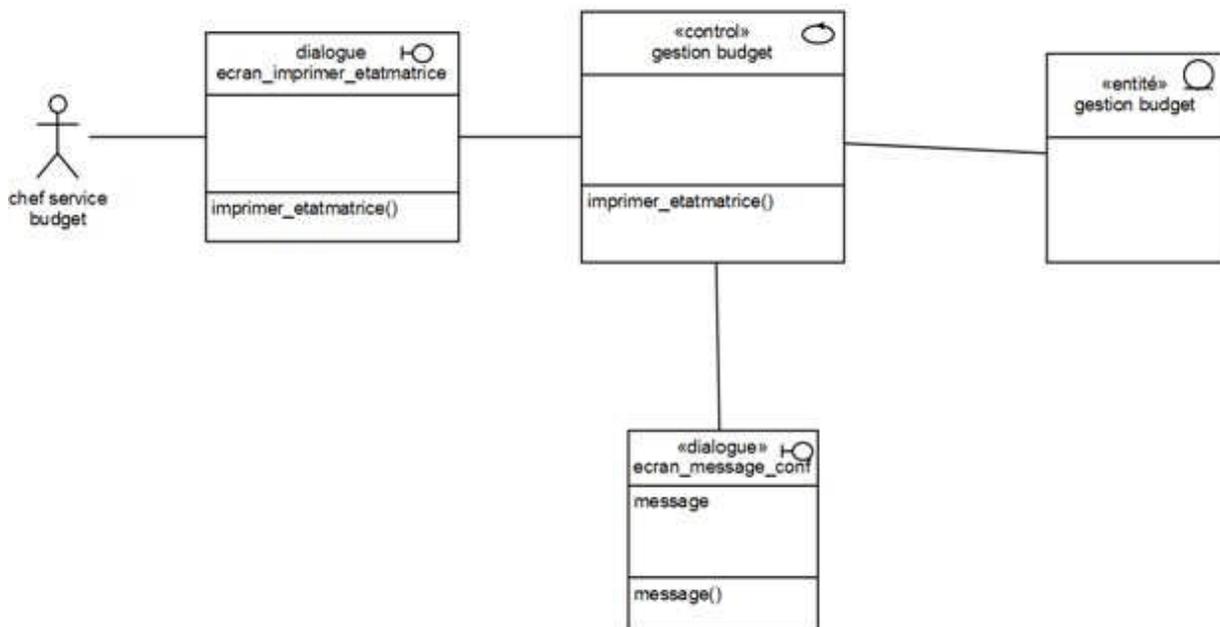
4.14. [Imprimer ordre de virement](#)



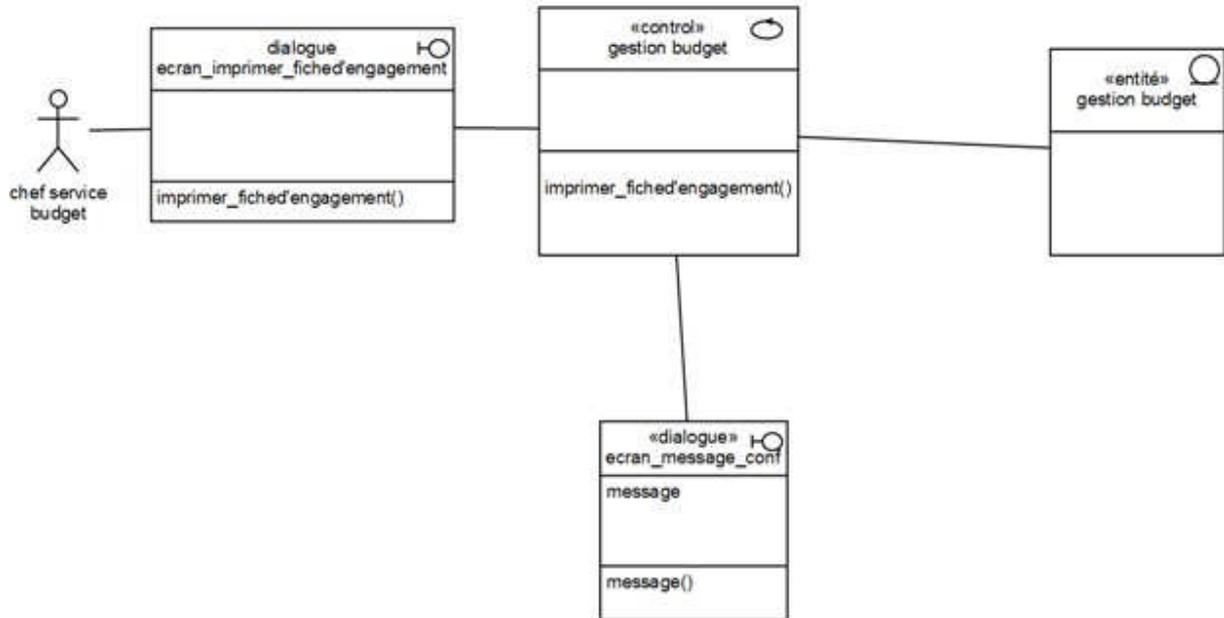
4.15. Gestion budget



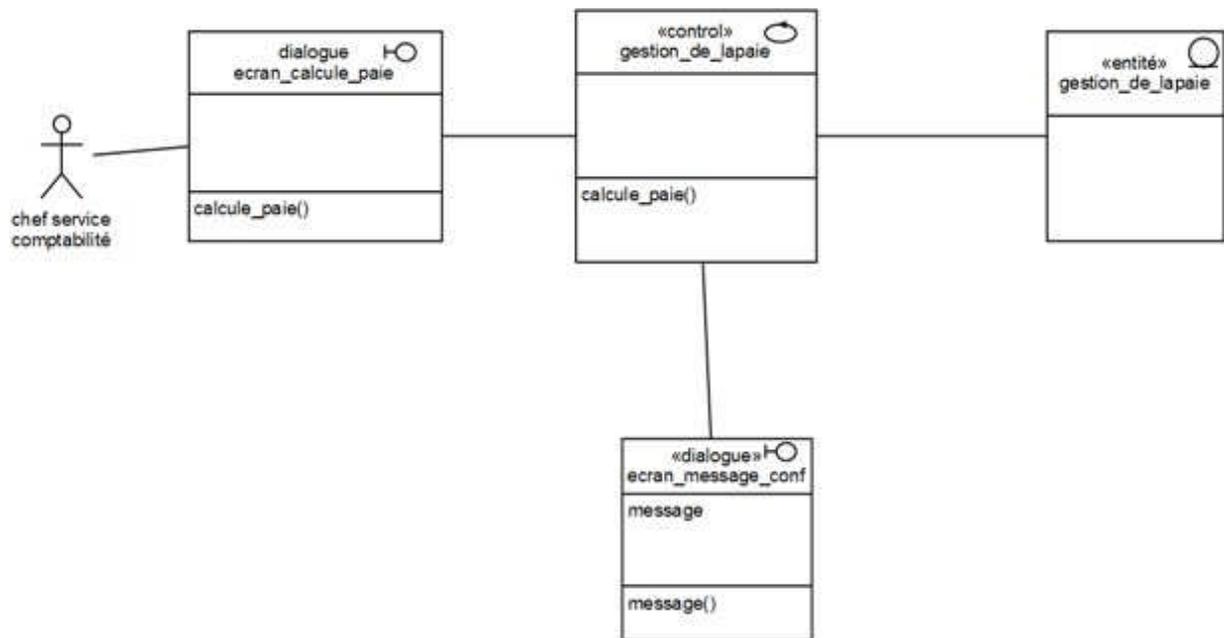
4.16. Imprimer état matrice



4.17. [Imprimer fiche d'engagement](#)



4.18. [Gestion de la paie](#)



*Chapitre 04 : phase de  
conception*

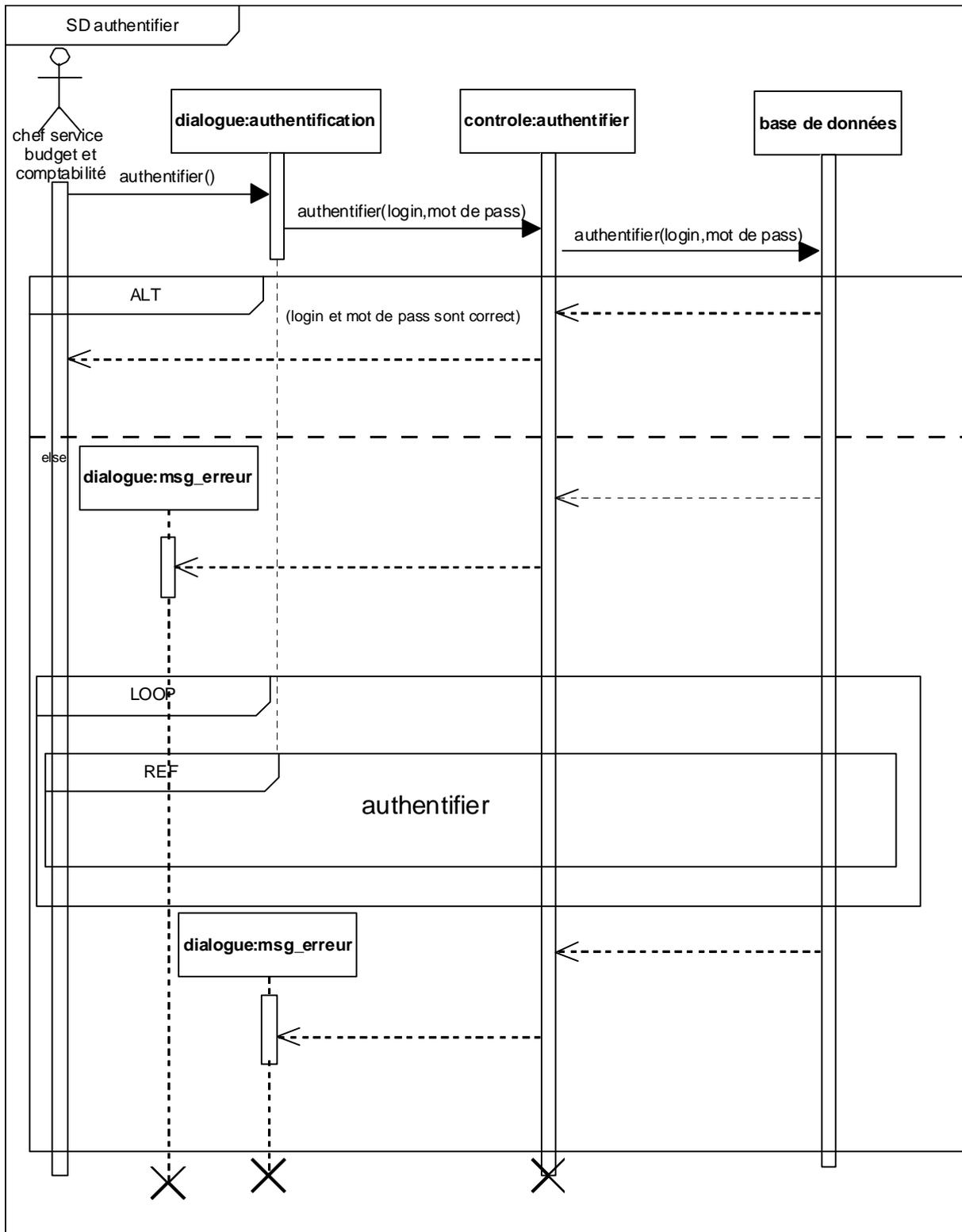
*Introduction*

- 1. diagramme de séquence*
- 2. diagramme de classes de  
conception*

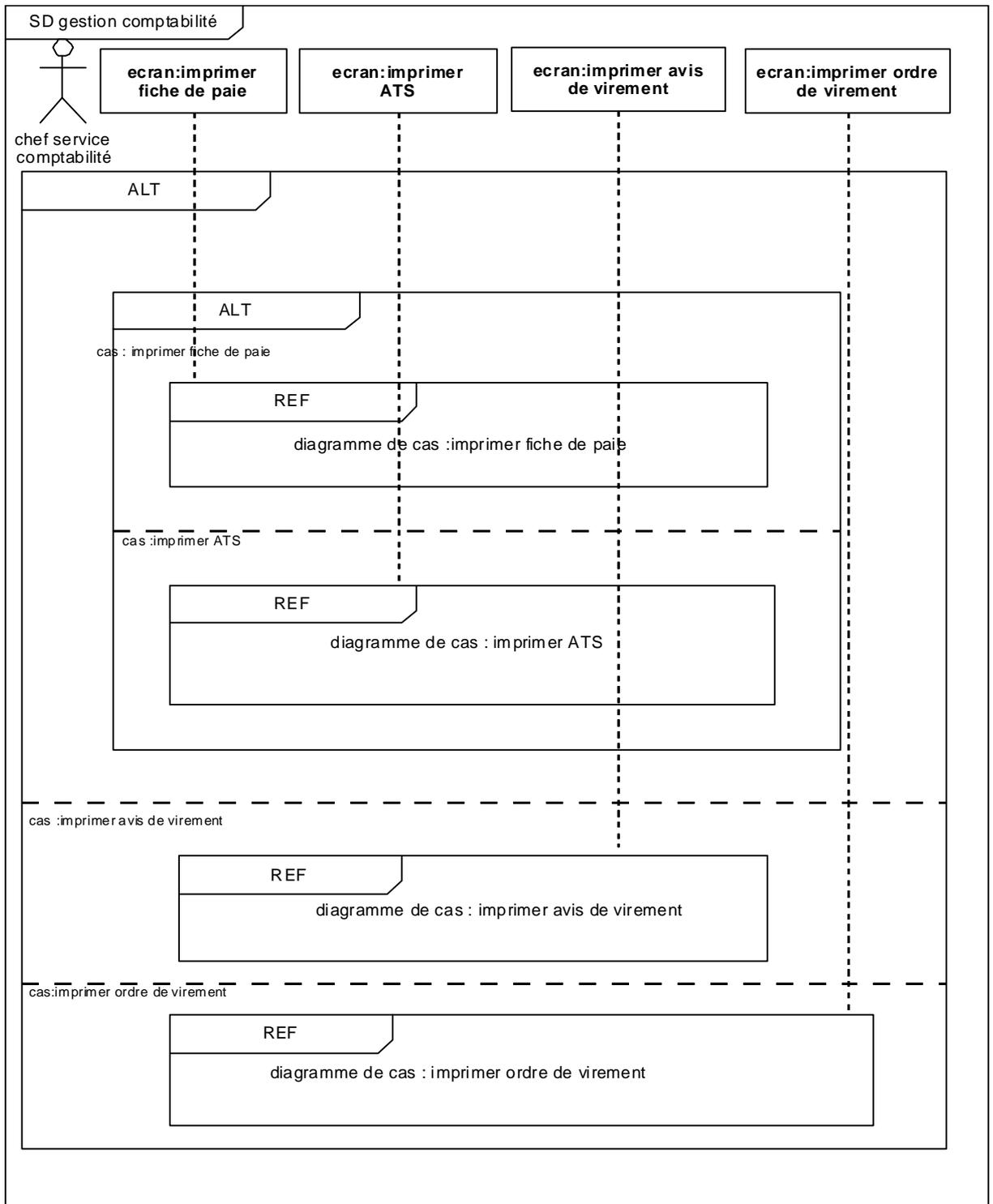
## 1. Diagramme de séquence

Maintenant, il faut attribuer précisément les responsabilités de comportement, dégagée par le diagramme de classe participante. Les résultats de cette réflexion sont présentés sous la forme de diagramme de séquence. 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 indépendant des choix technologique qui seront faits ultérieurement.

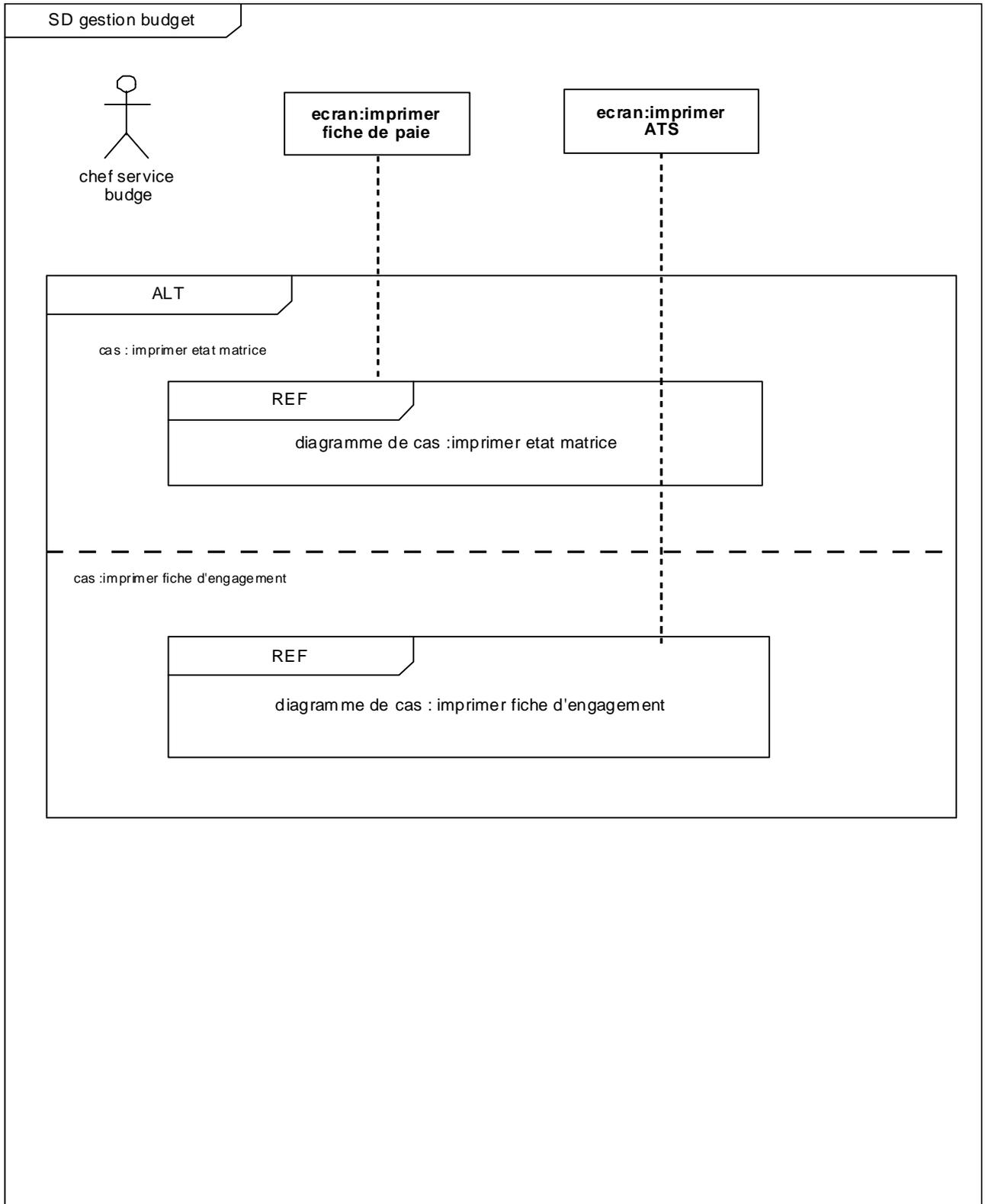
1.2. Authentifier



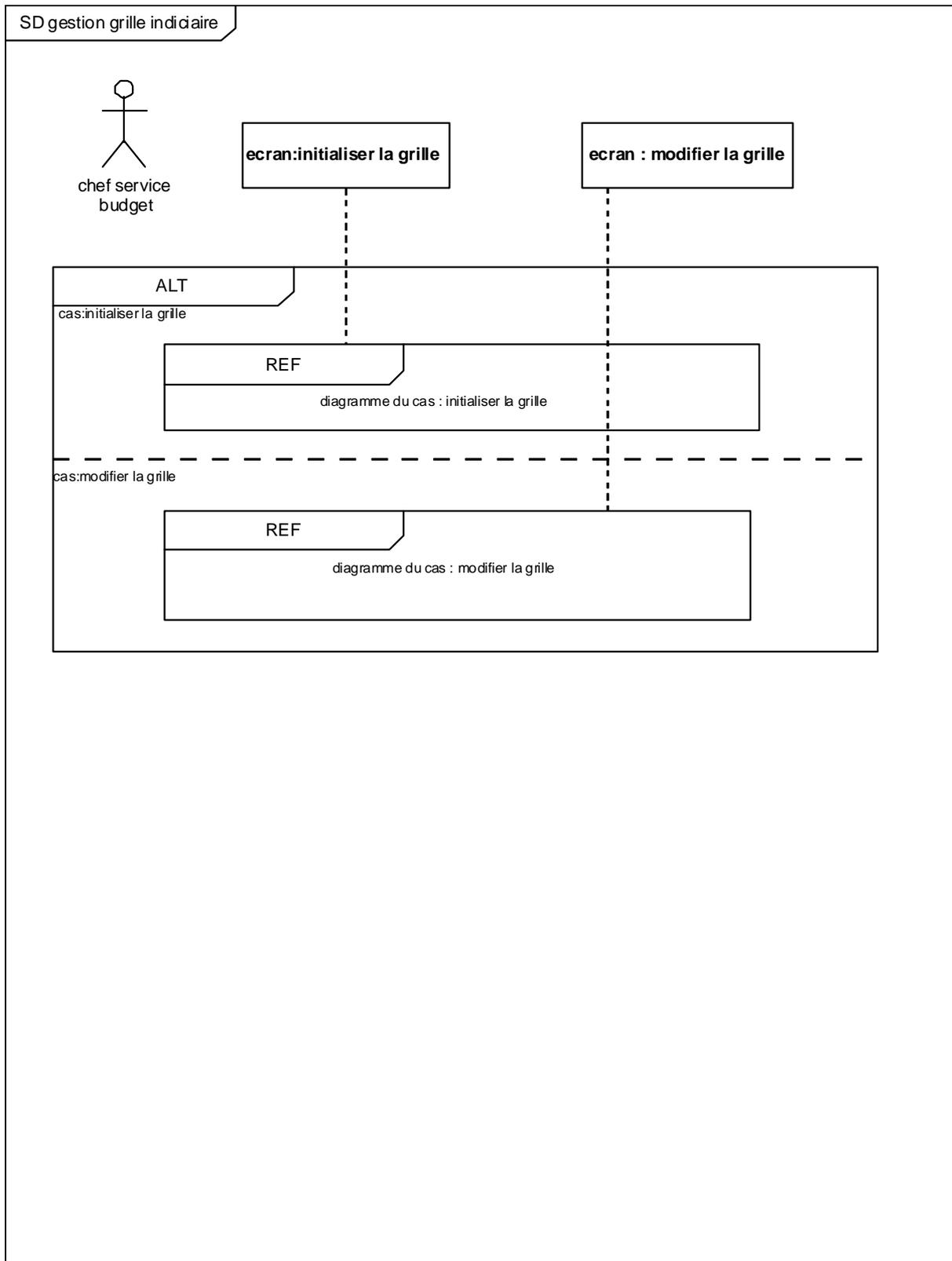
1.3. Gestion comptabilité



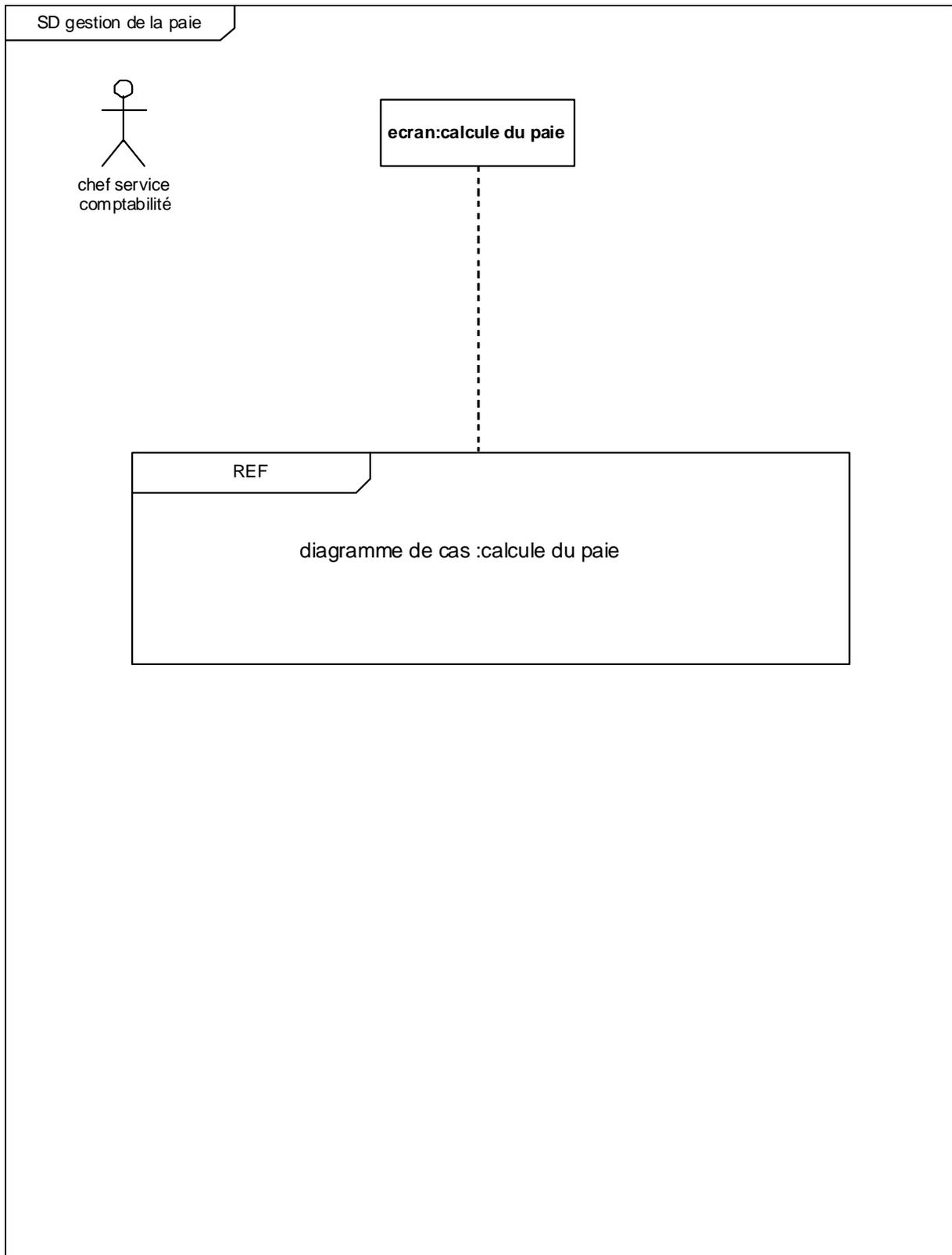
1.4. Gestion budget



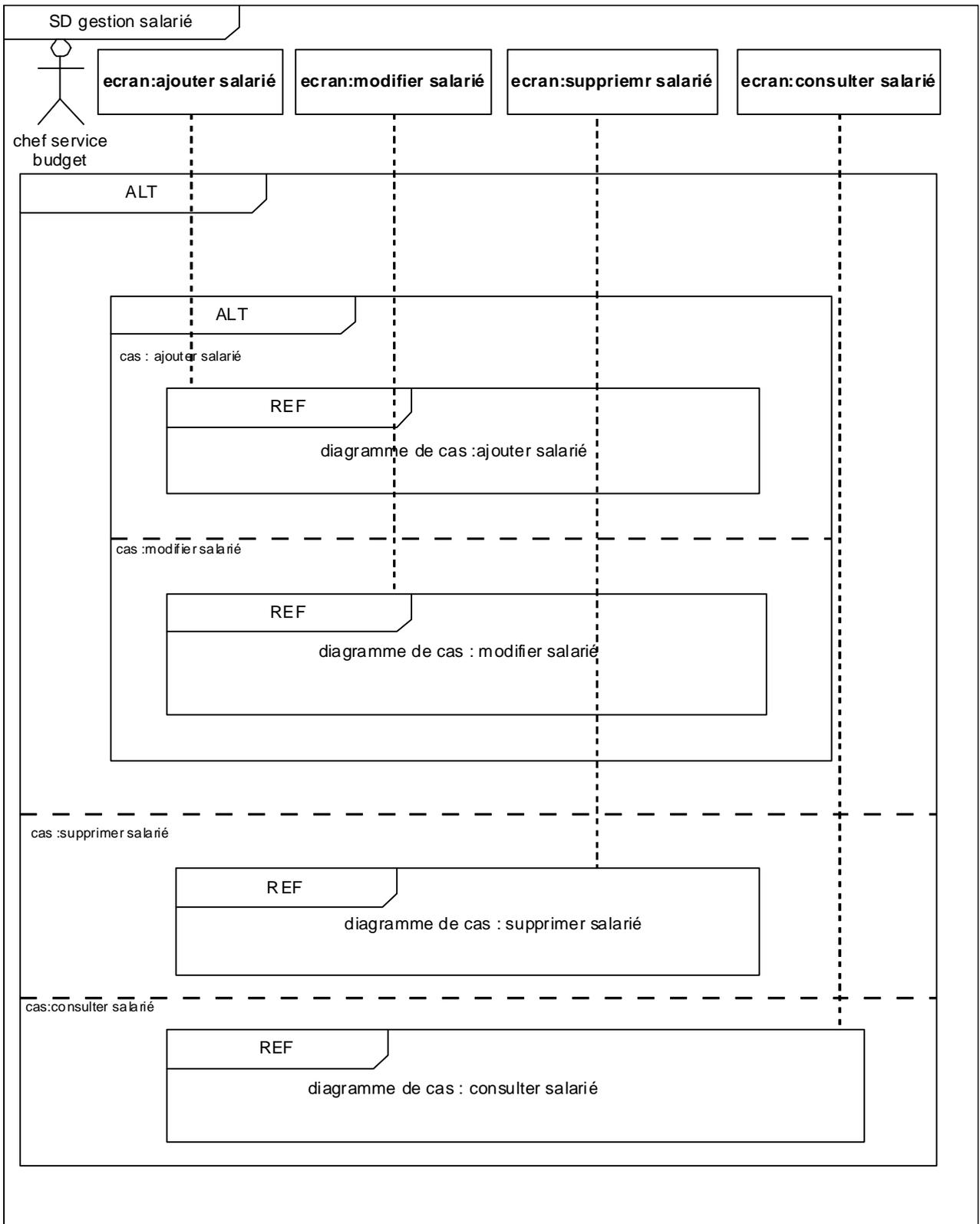
1.5. Gestion grille indiciaire



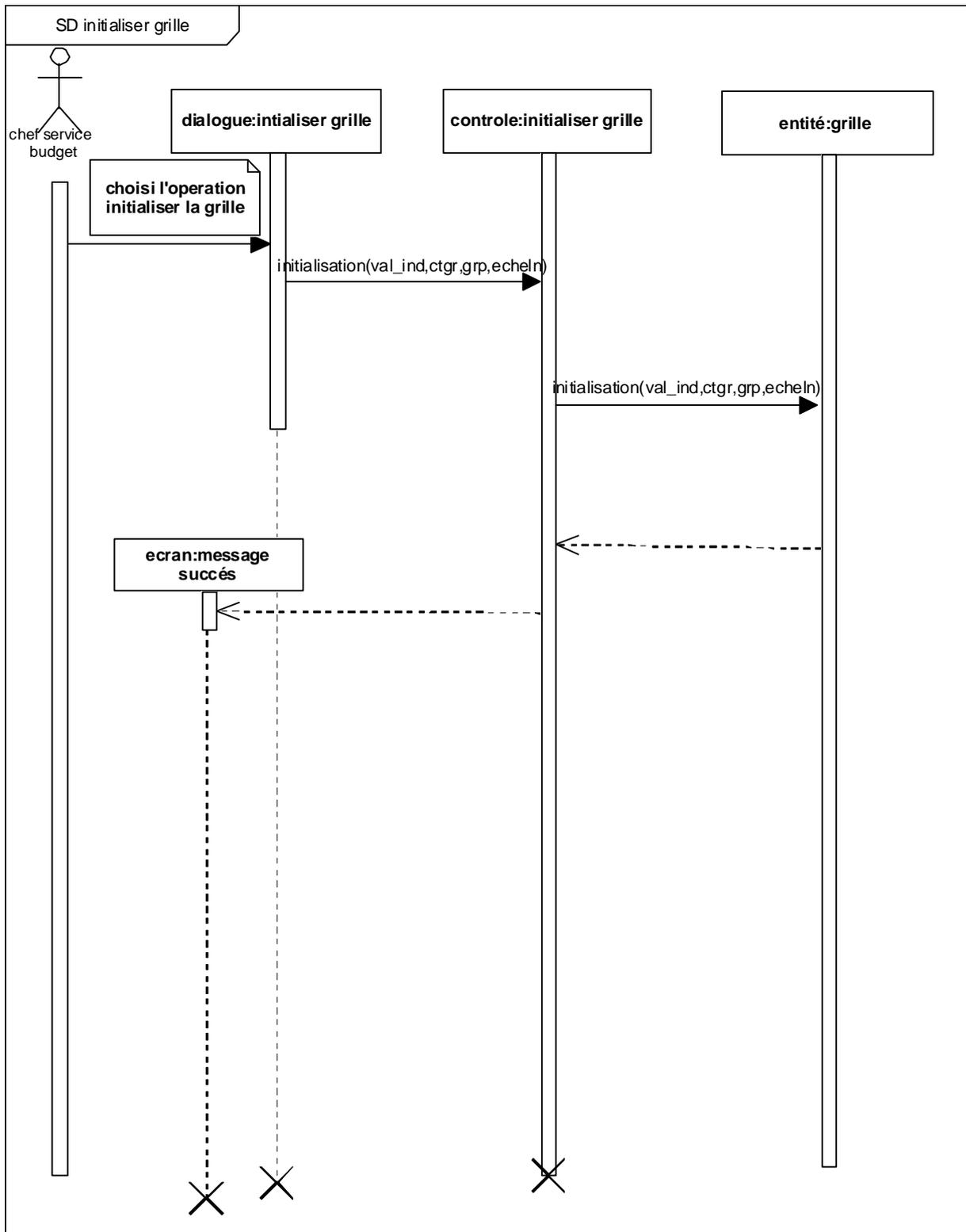
1.6. Gestion de la paie



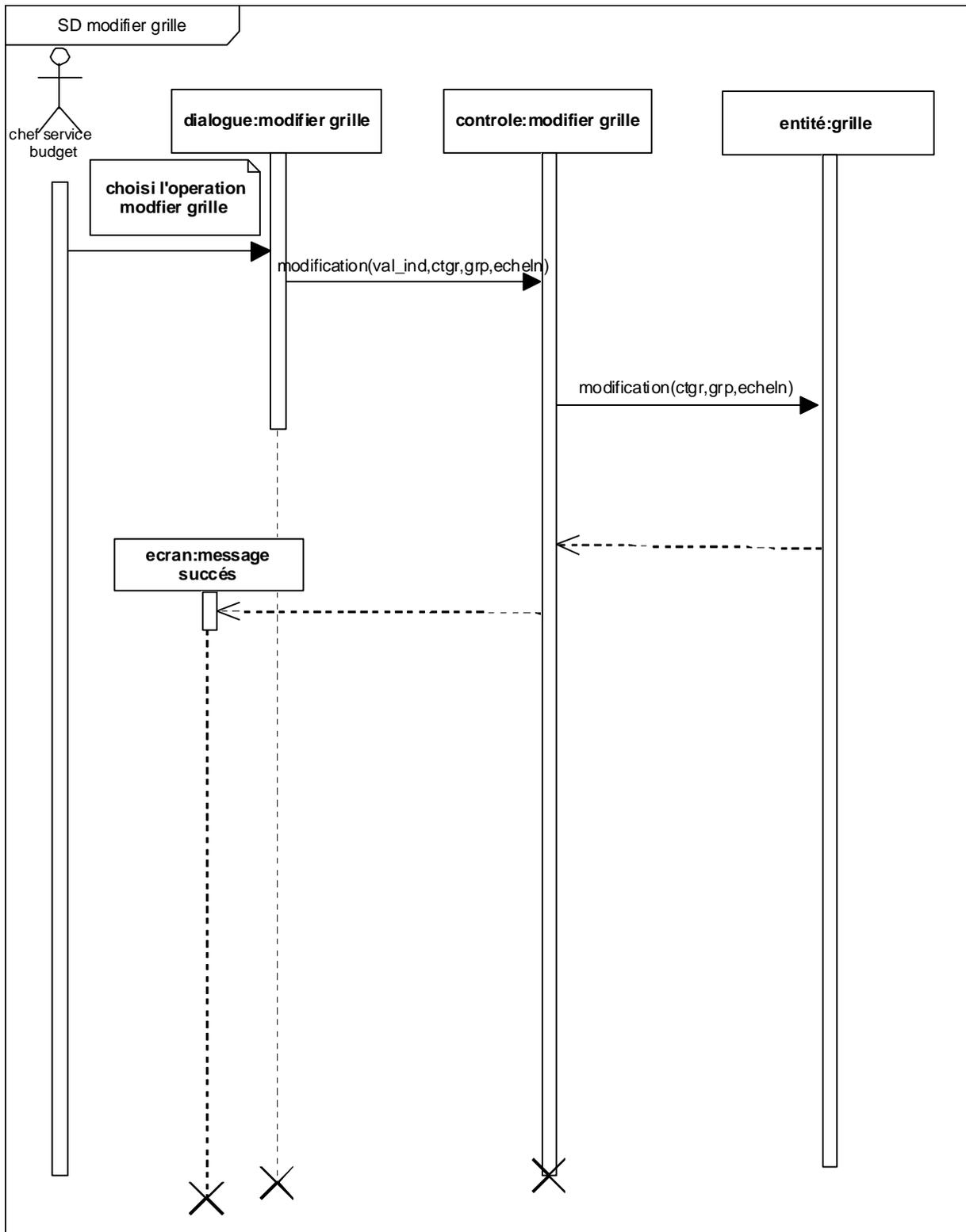
1.7. Gestion salarié



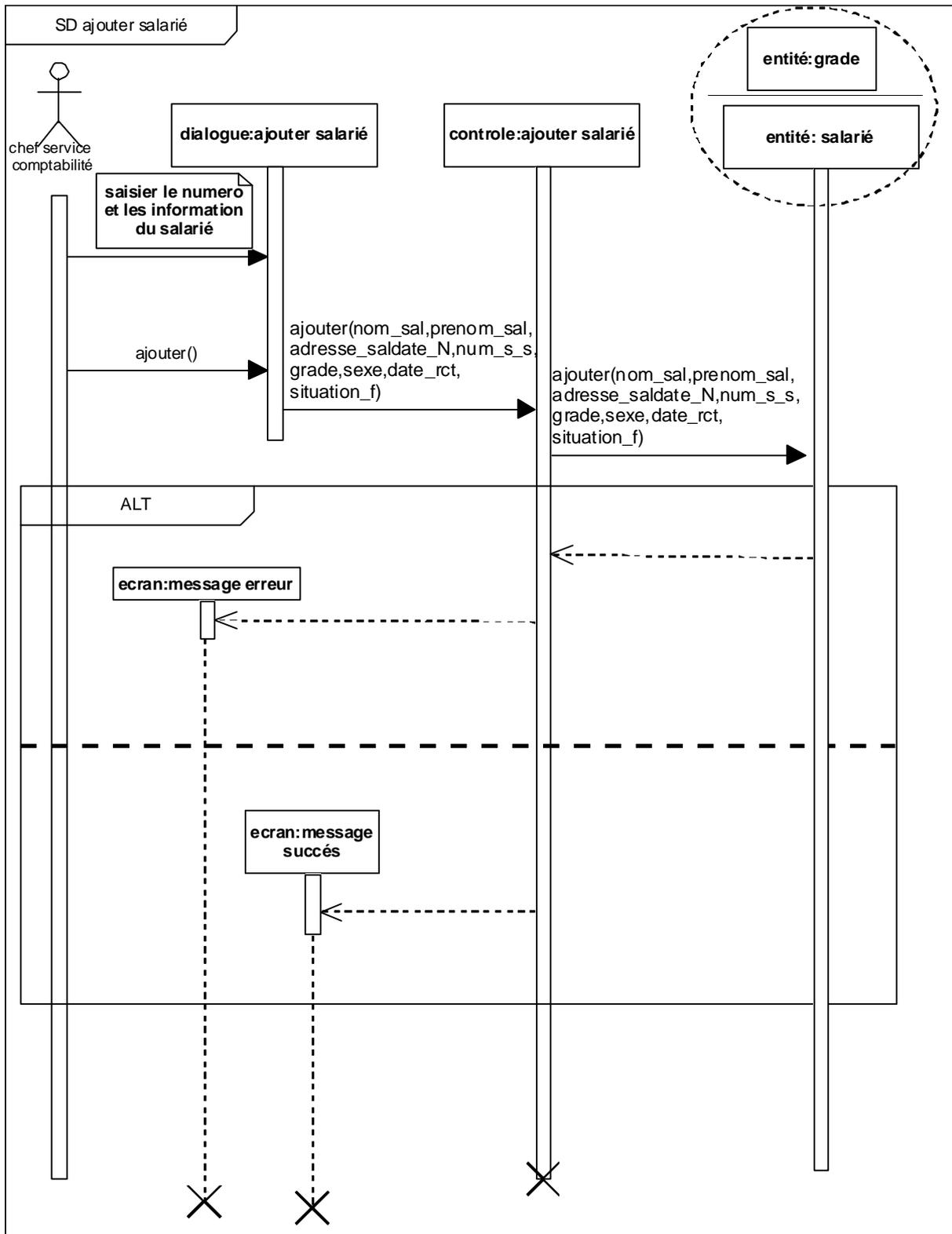
1.8. Initialiser la grille indiciaire



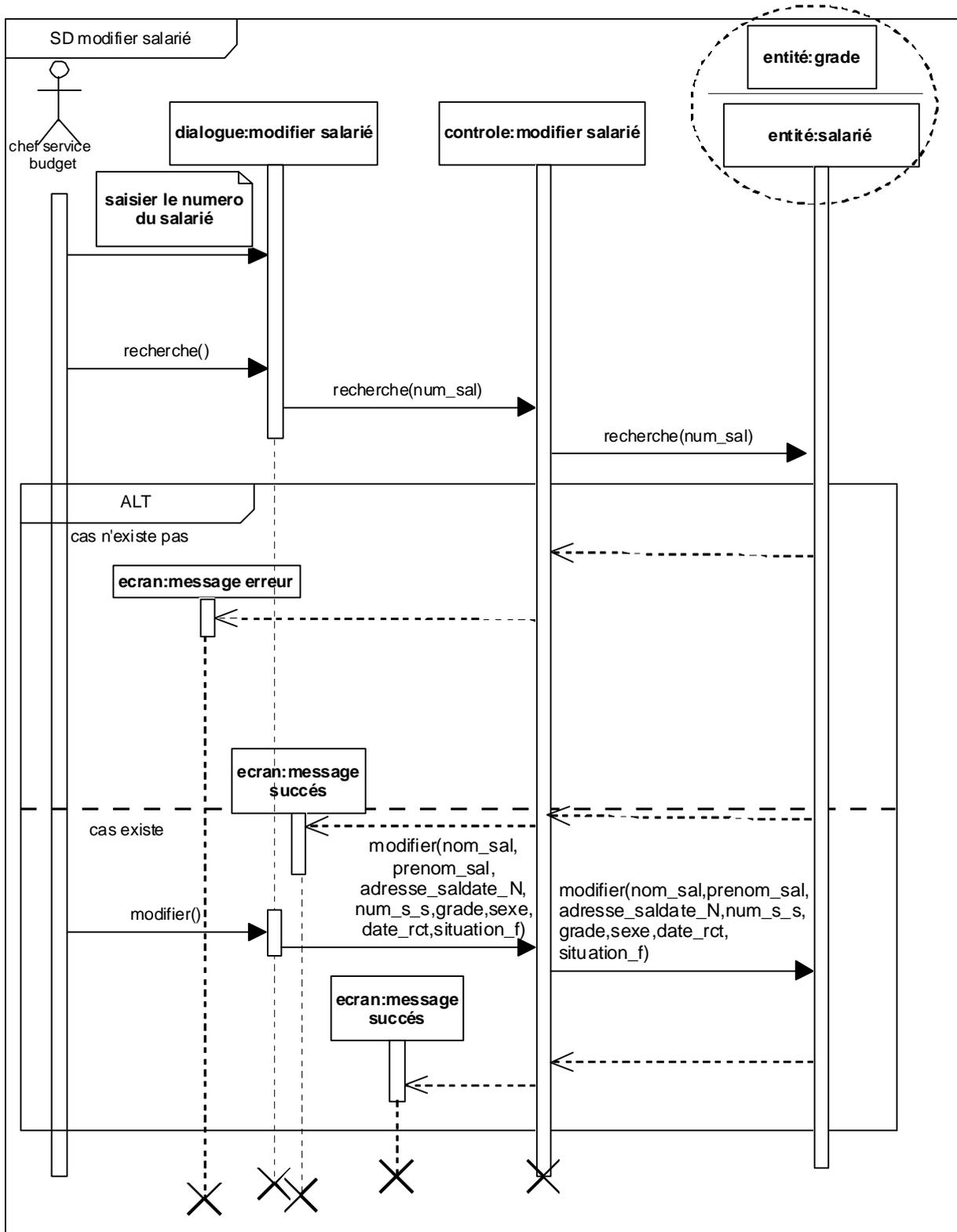
1.9. Modifier grille indiciaire



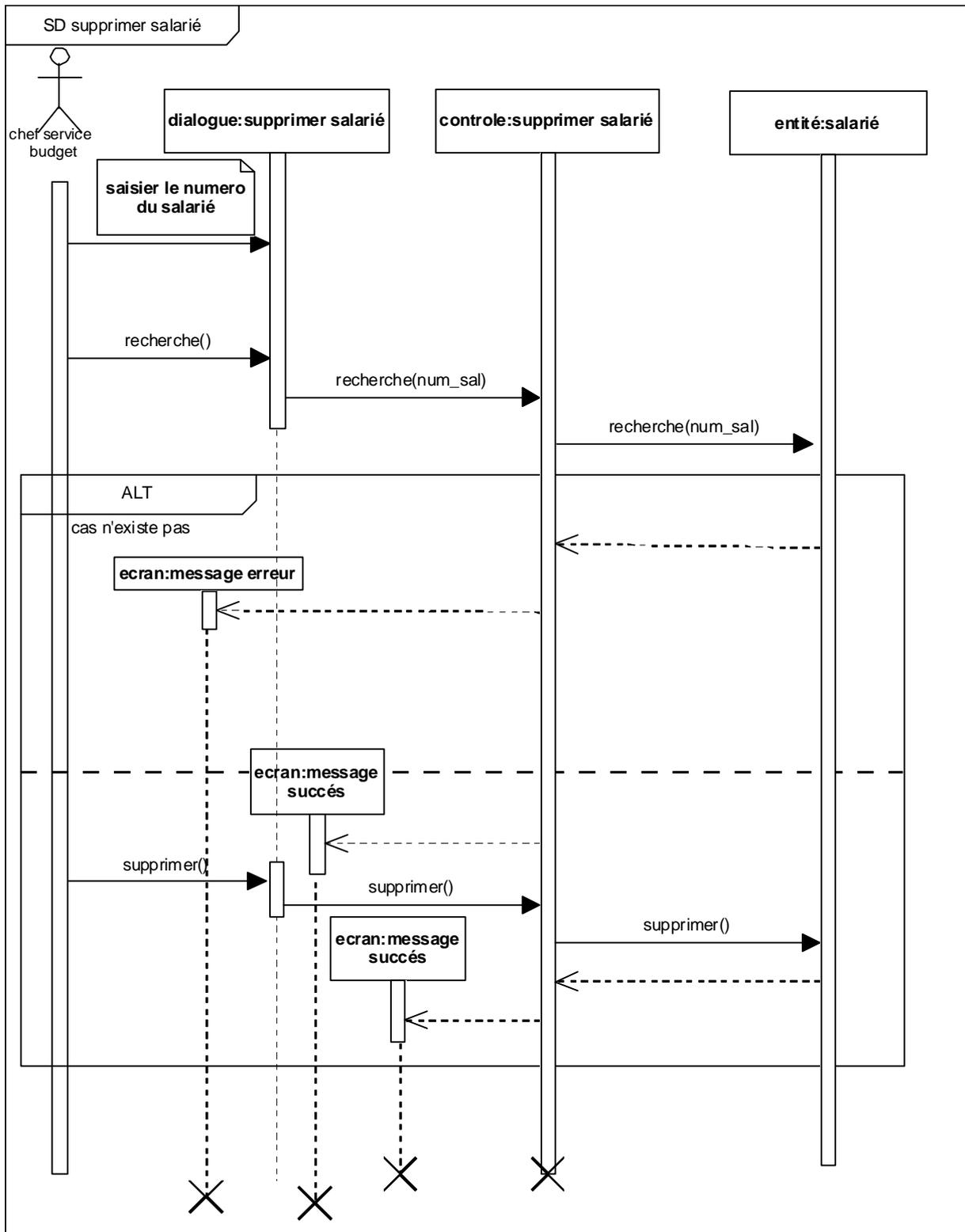
1.10. Ajouter salarié



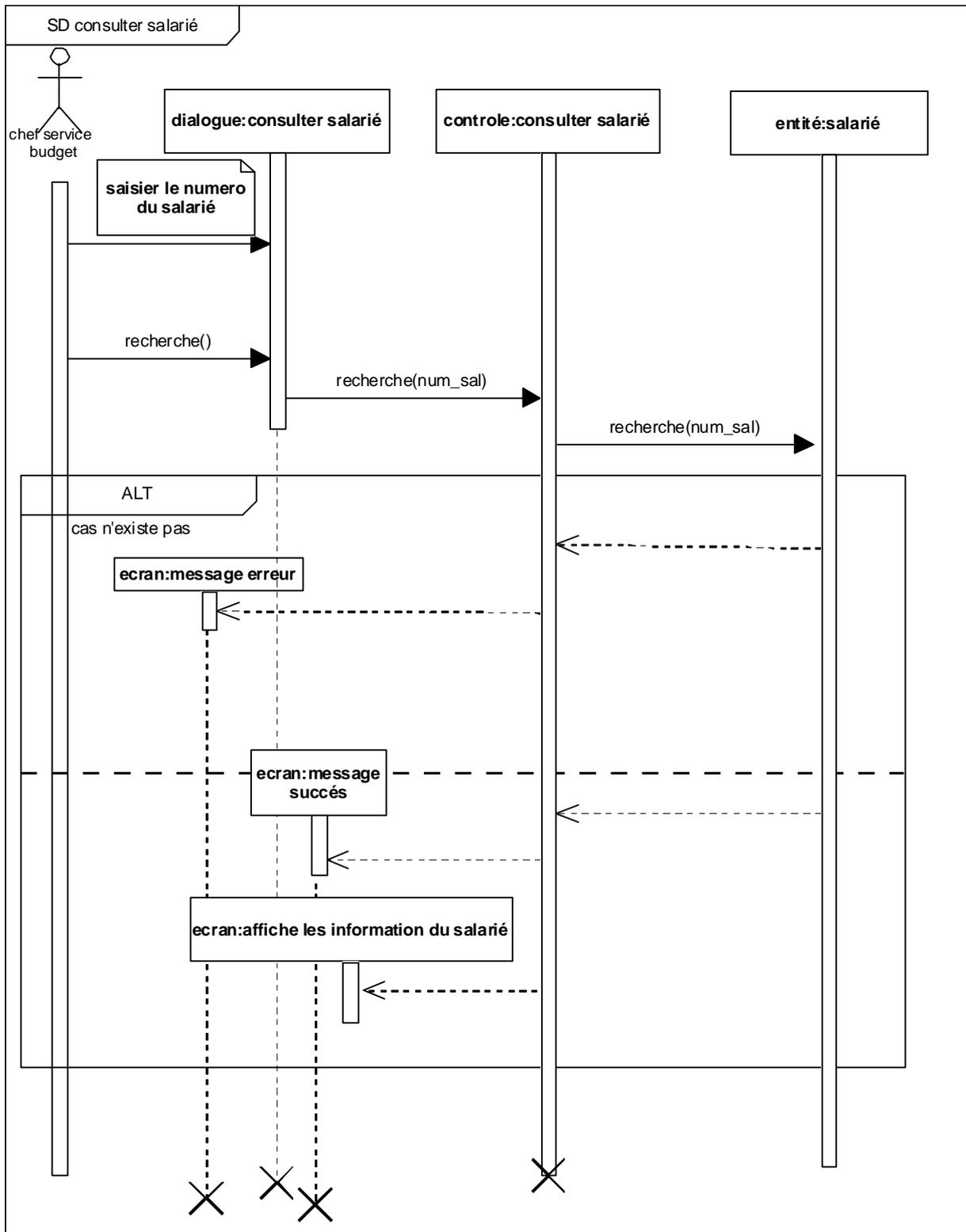
1.11. Modifier salarié



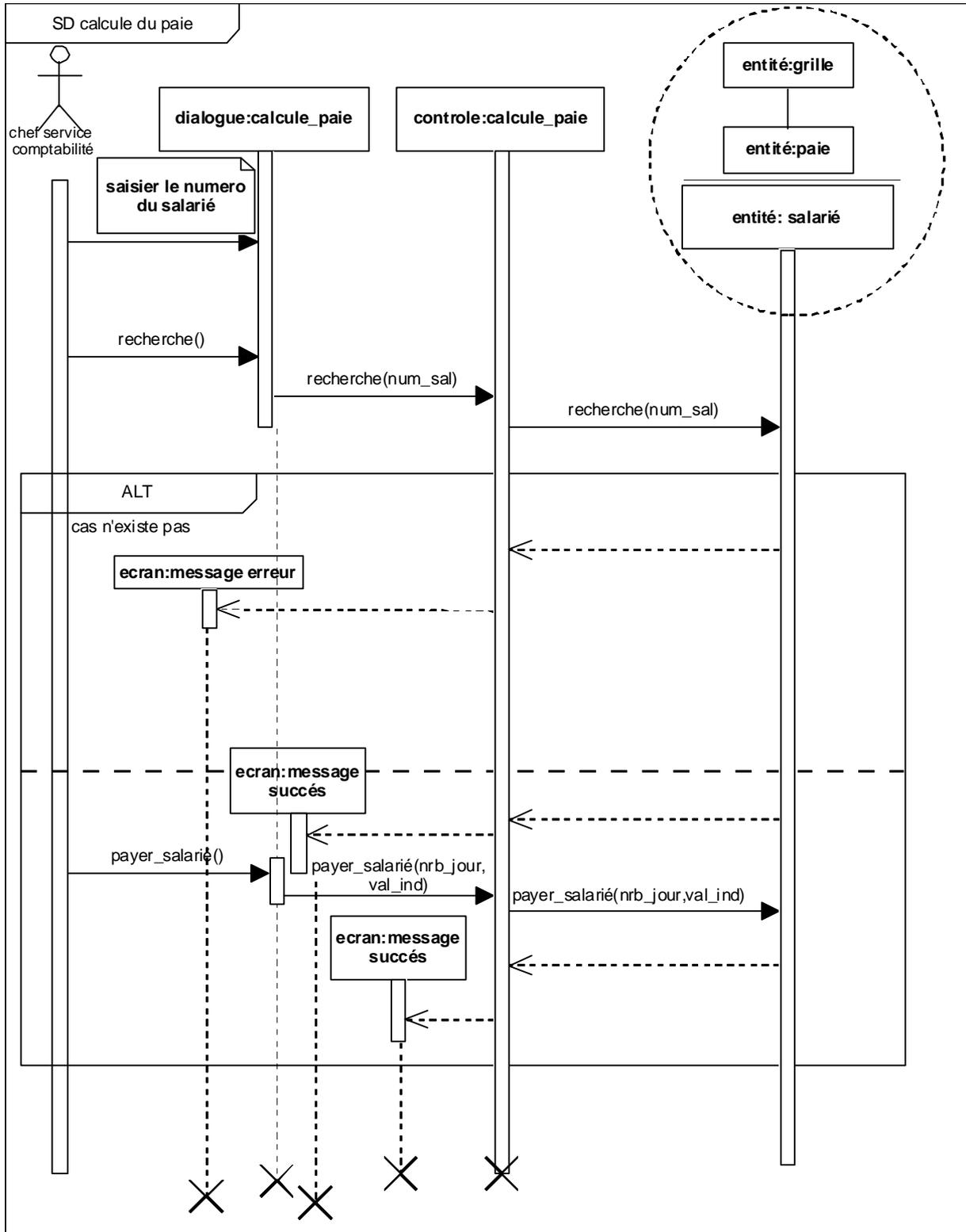
1.12. Supprimer salarié



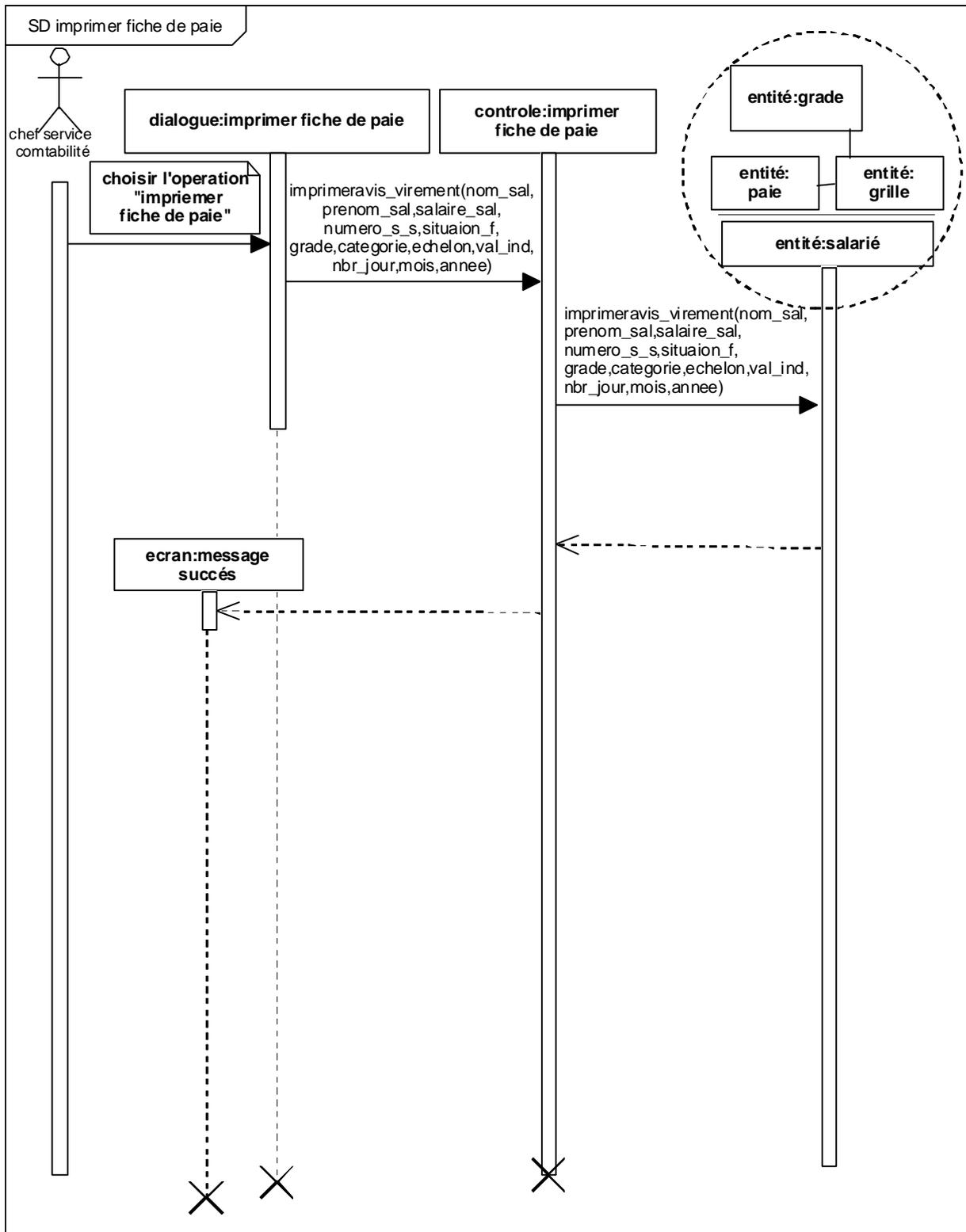
1.13. Consulter salarié



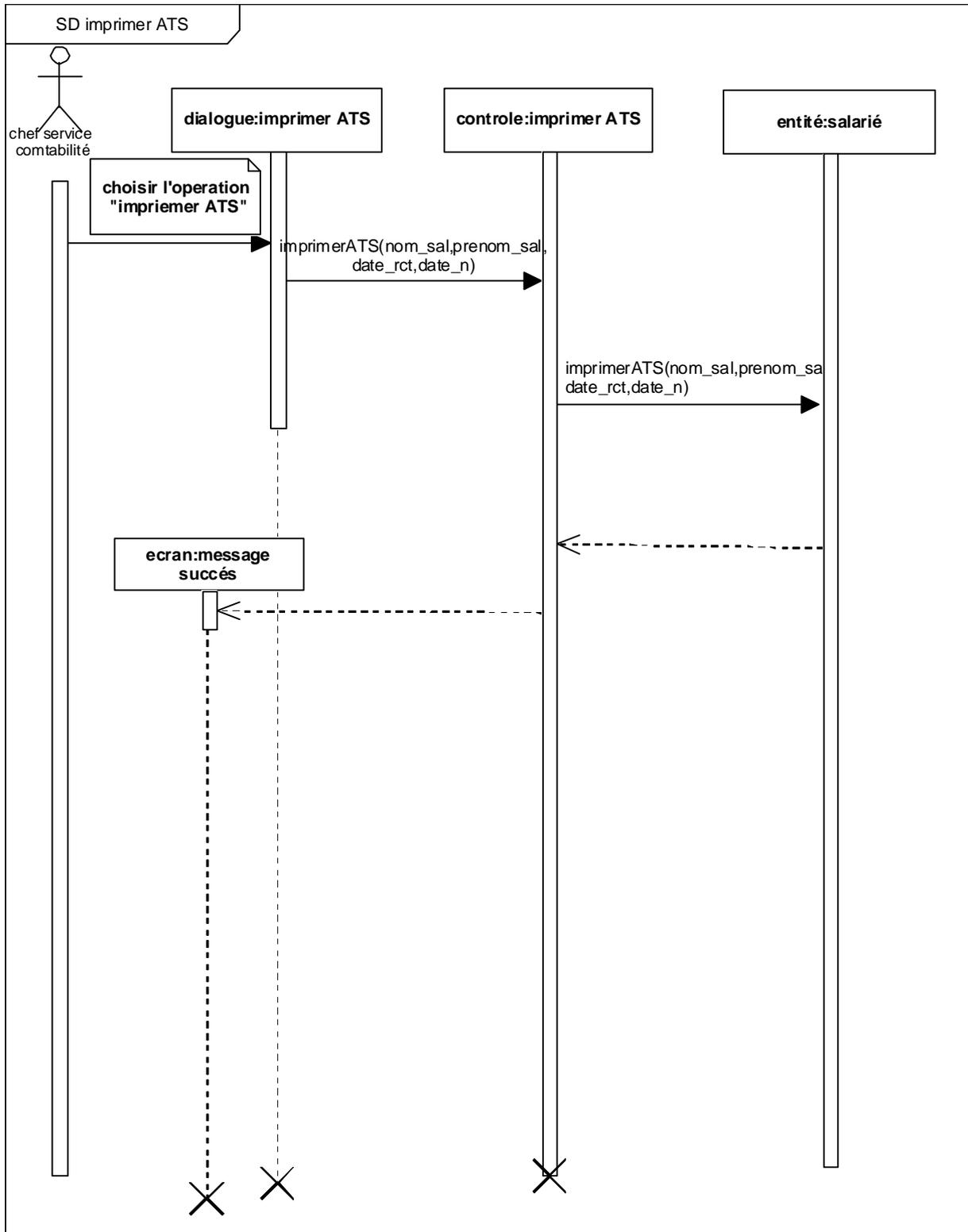
1.14. Calcul de la paie



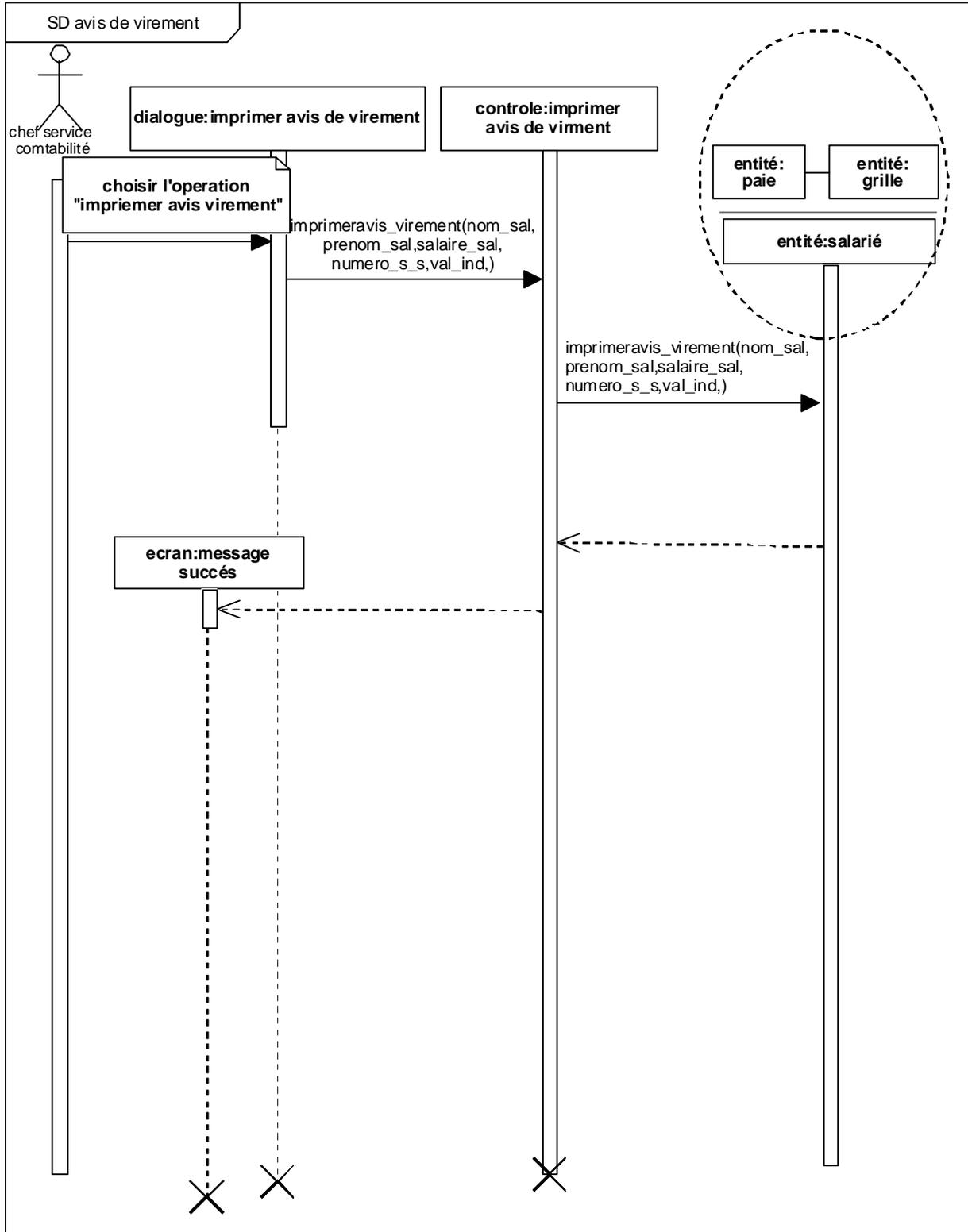
1.15. Imprimer fiche de paie



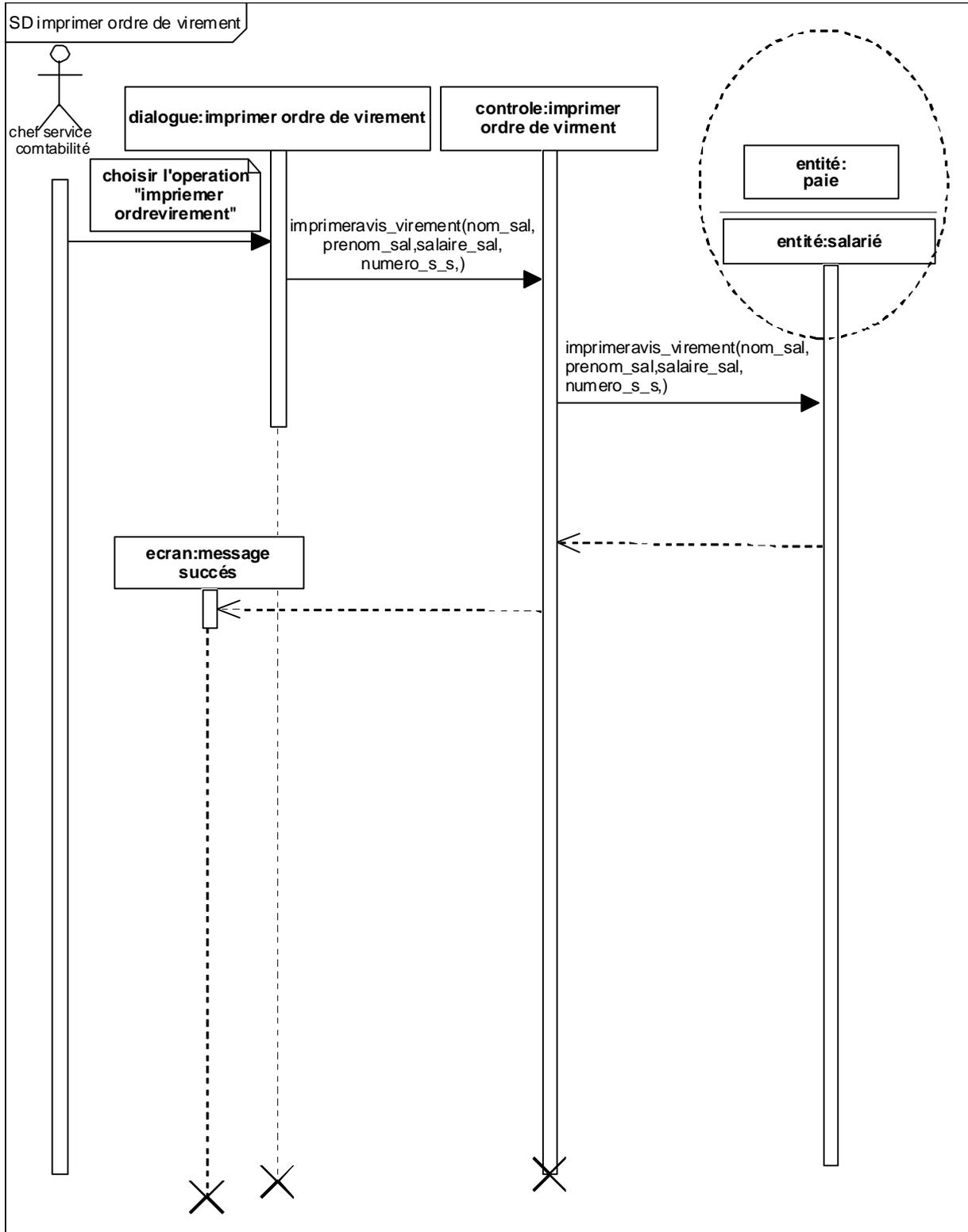
1.16. Imprimer ATS



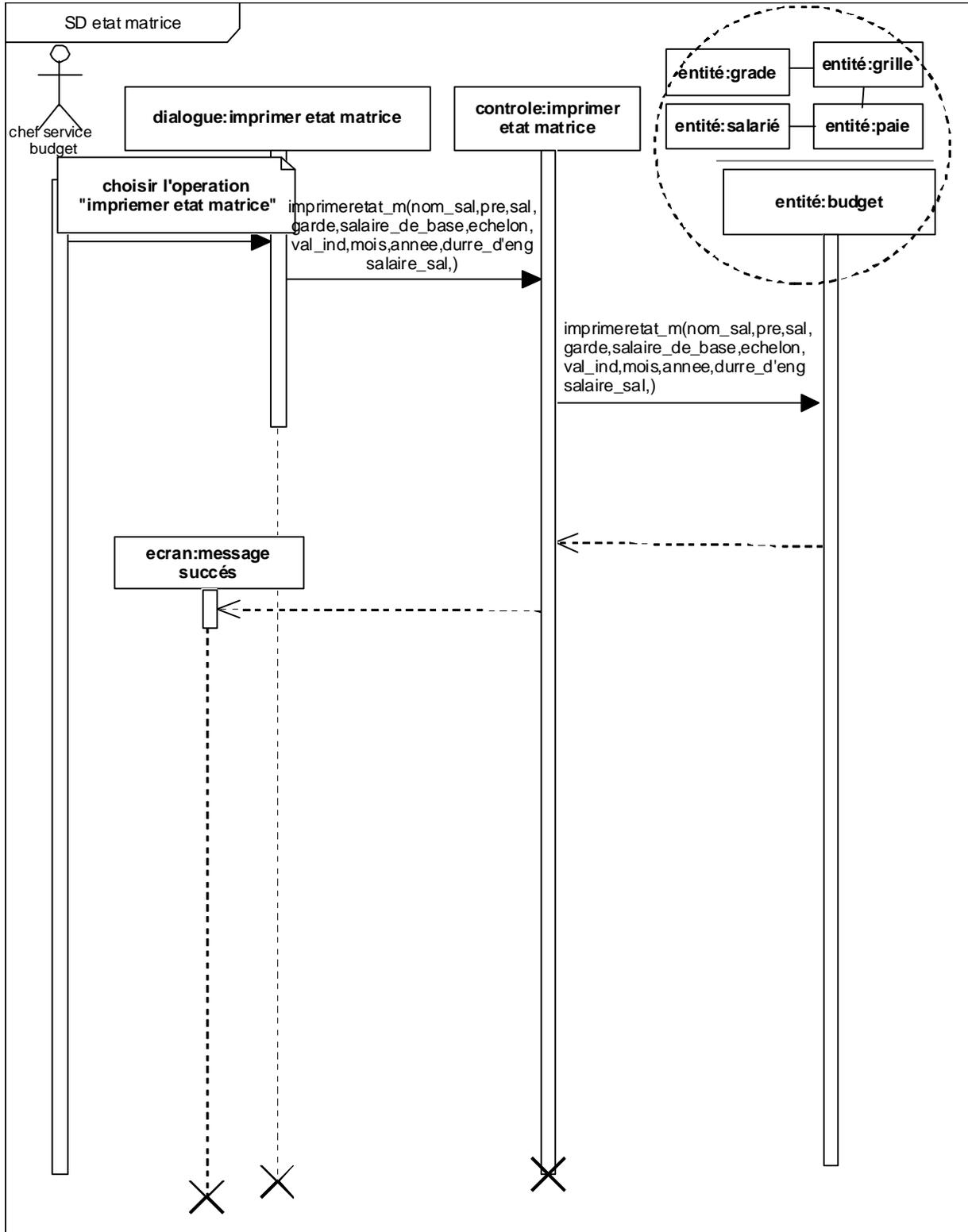
1.17. Avis de virement



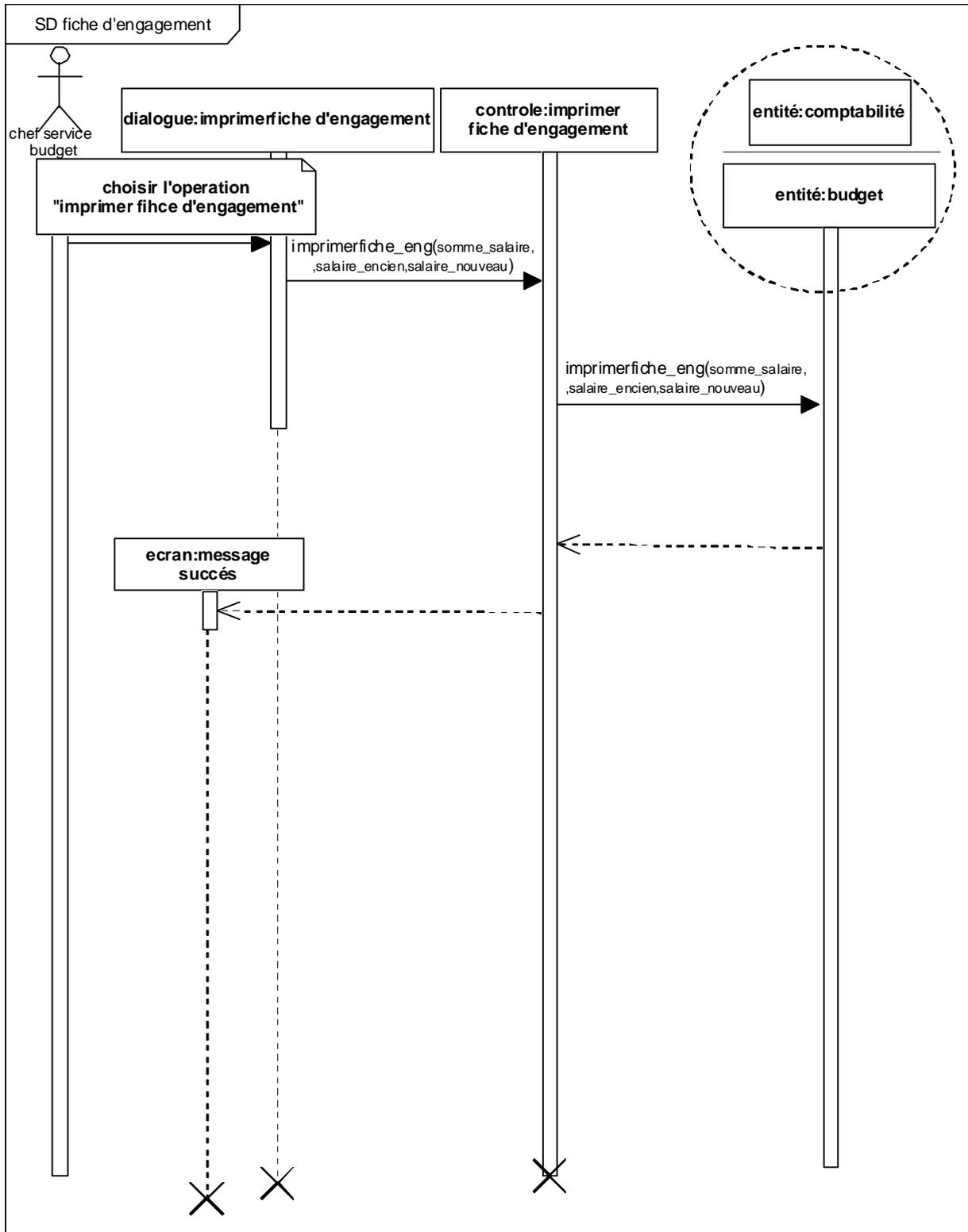
1.18. Ordre de virement



1.19. Imprimer Etat matrice



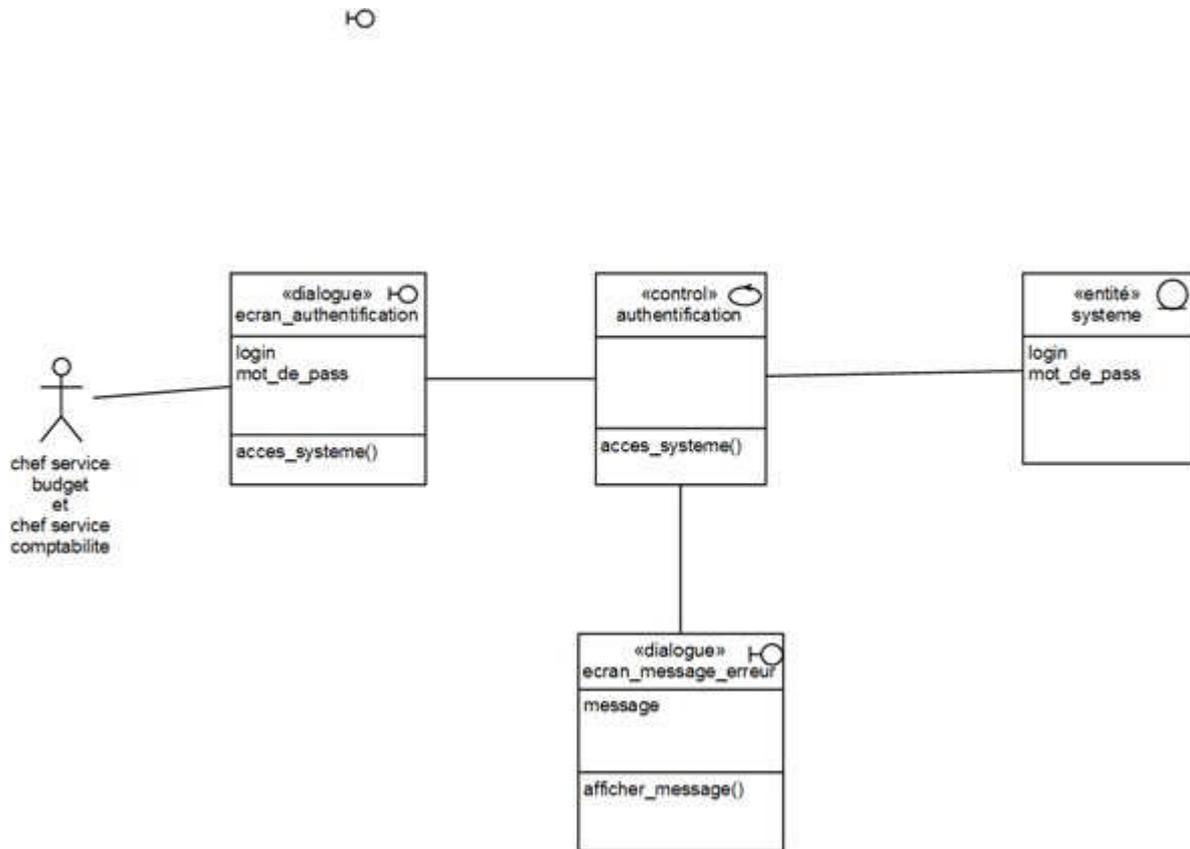
1.20. Imprimer Fiche d'engagement



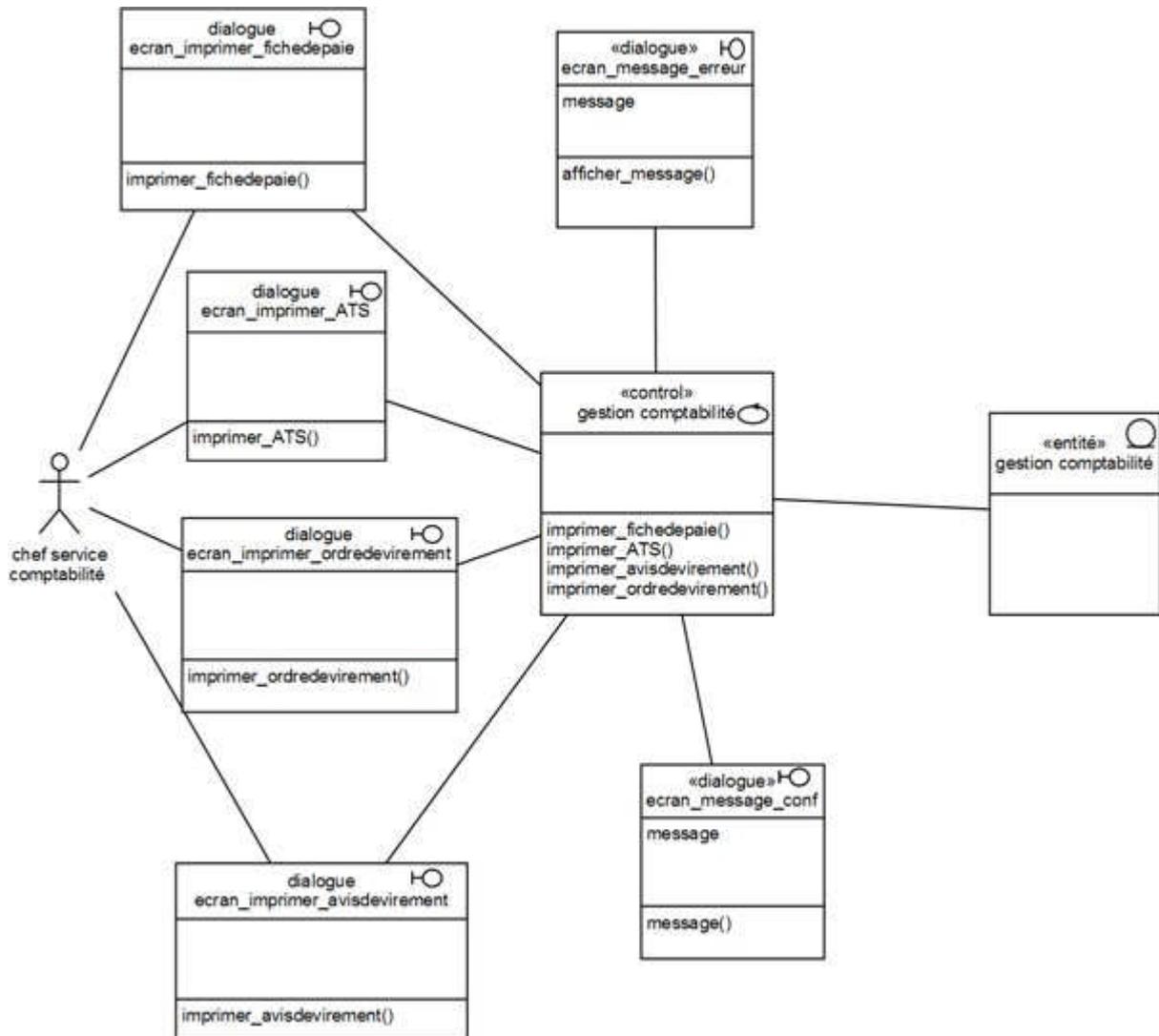
## 2. Diagramme de classes de conception

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 de séquence. Il faut maintenant le compléter en précisant les opérations privées des différentes classes.

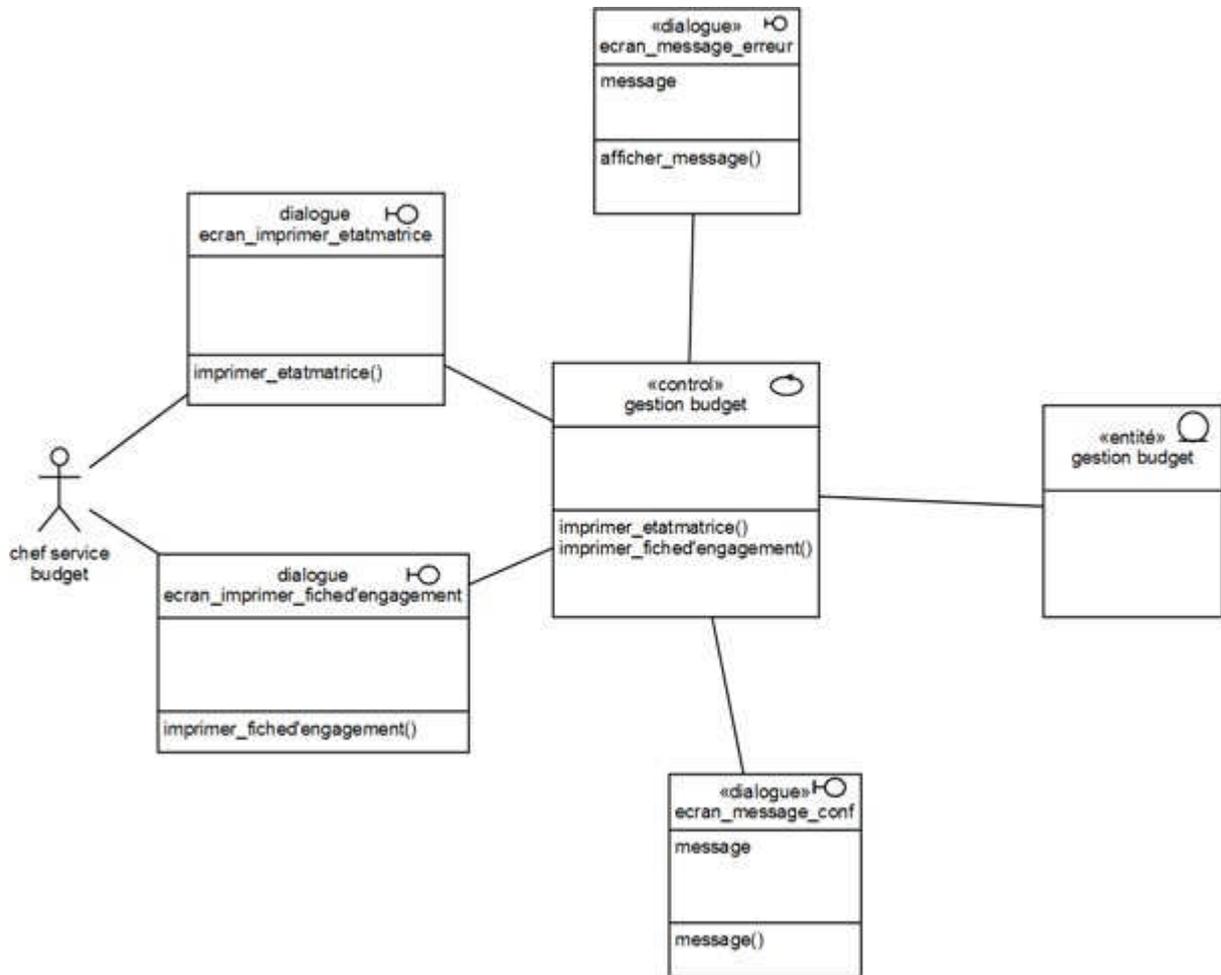
## 2.1. Authentifier



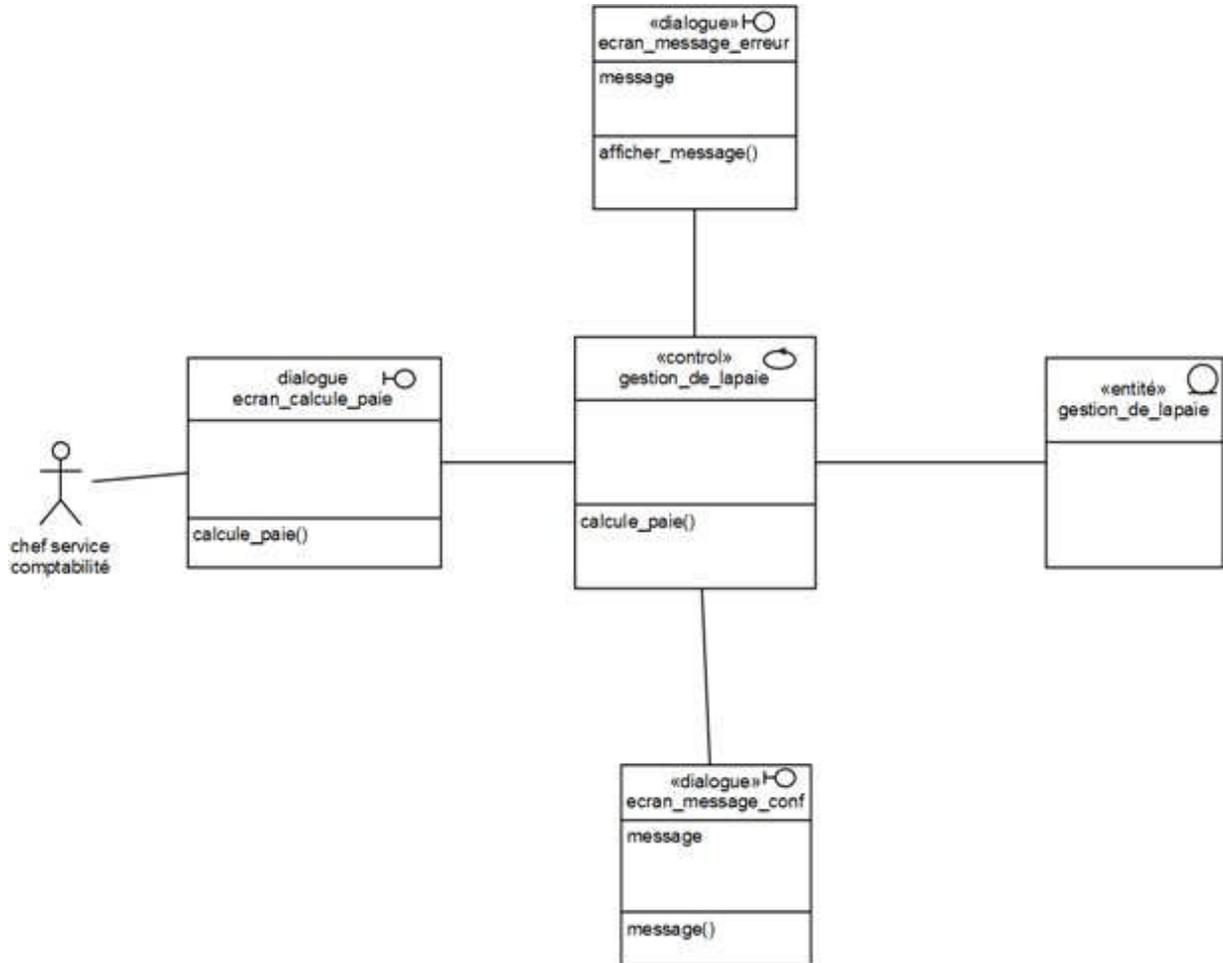
## 2.2. Comptabilité



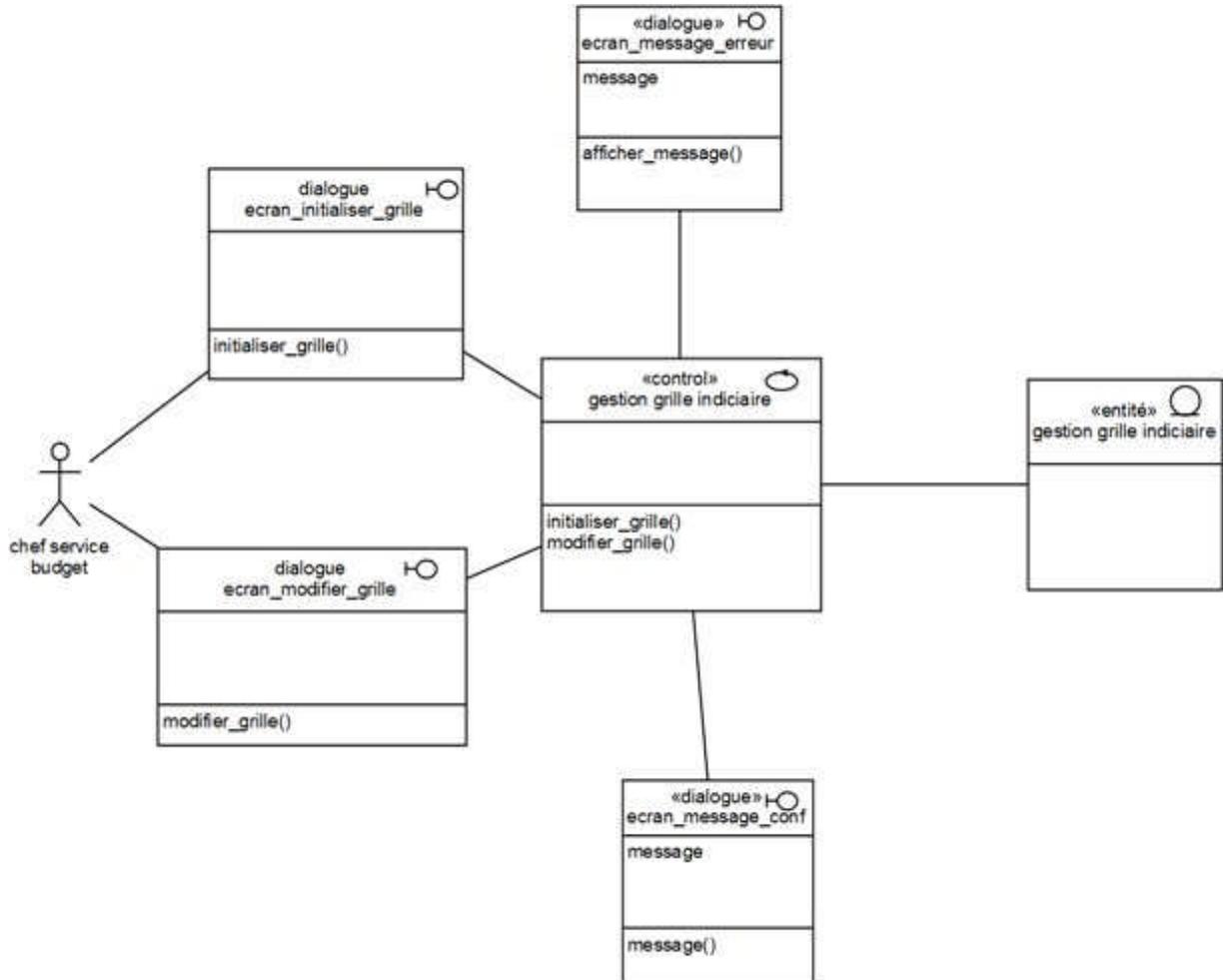
### 2.3. Budget



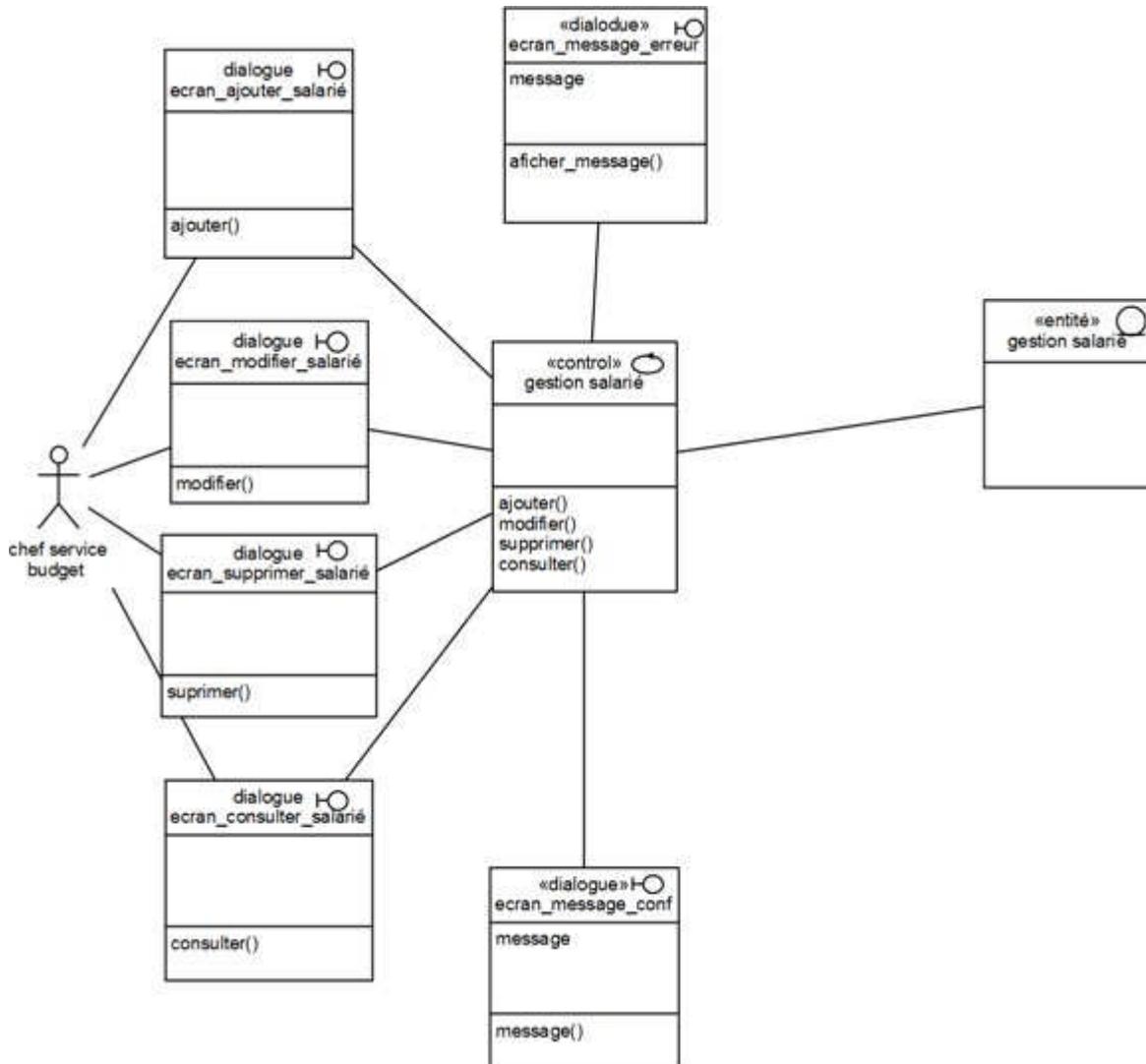
2.4. Paie



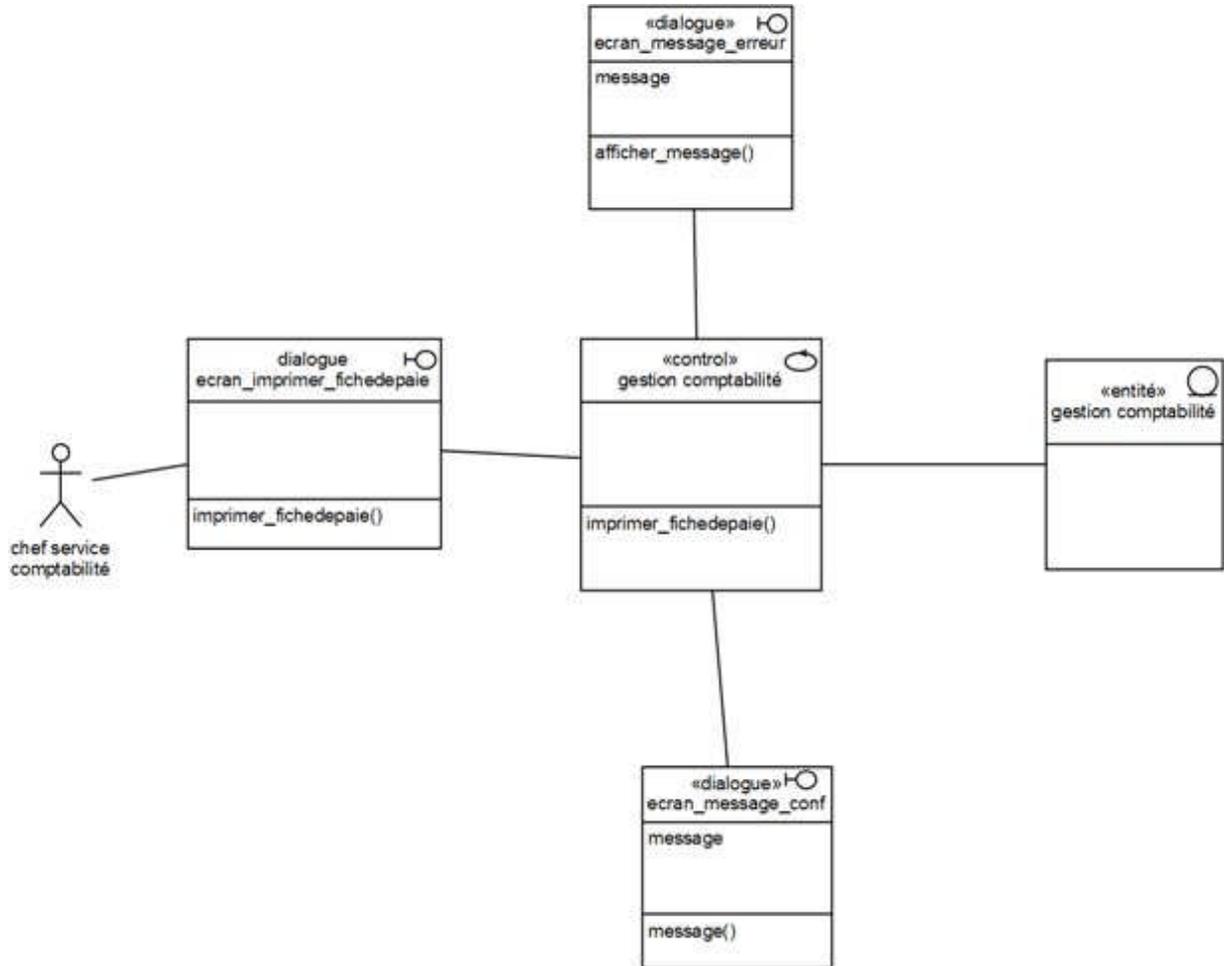
2.5. Grille



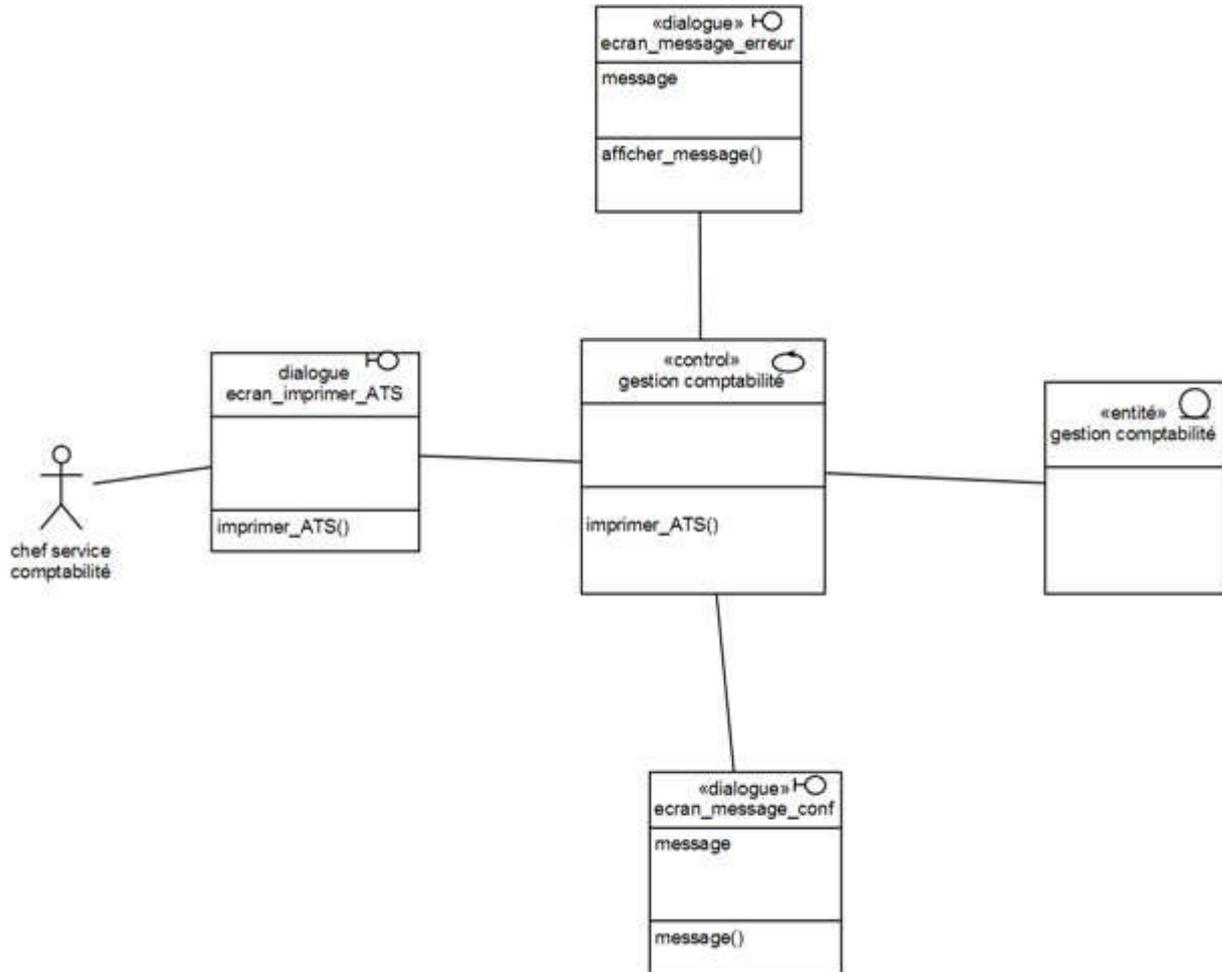
2.6. Salarié



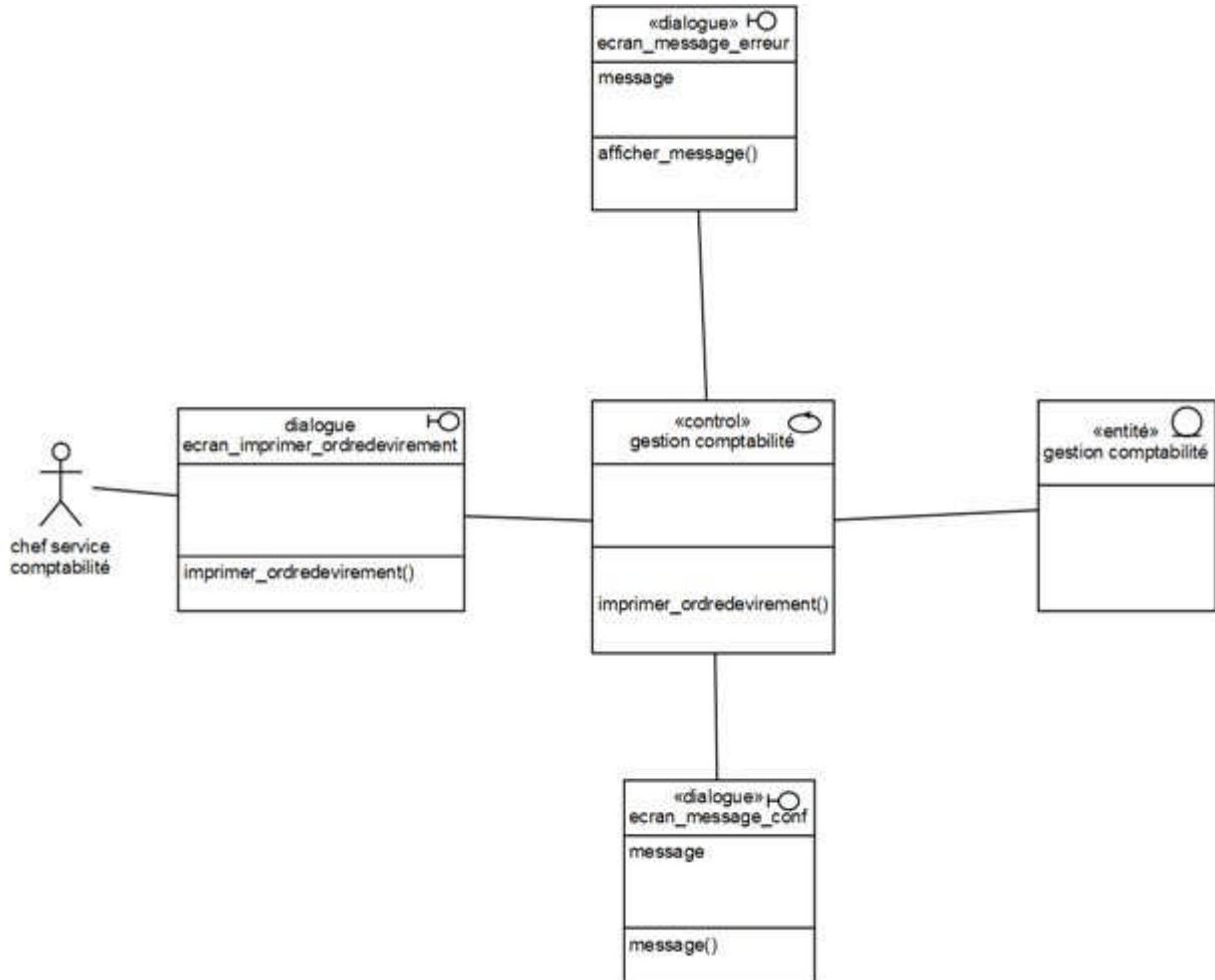
2.7. [Imprimer fiche de paie](#)



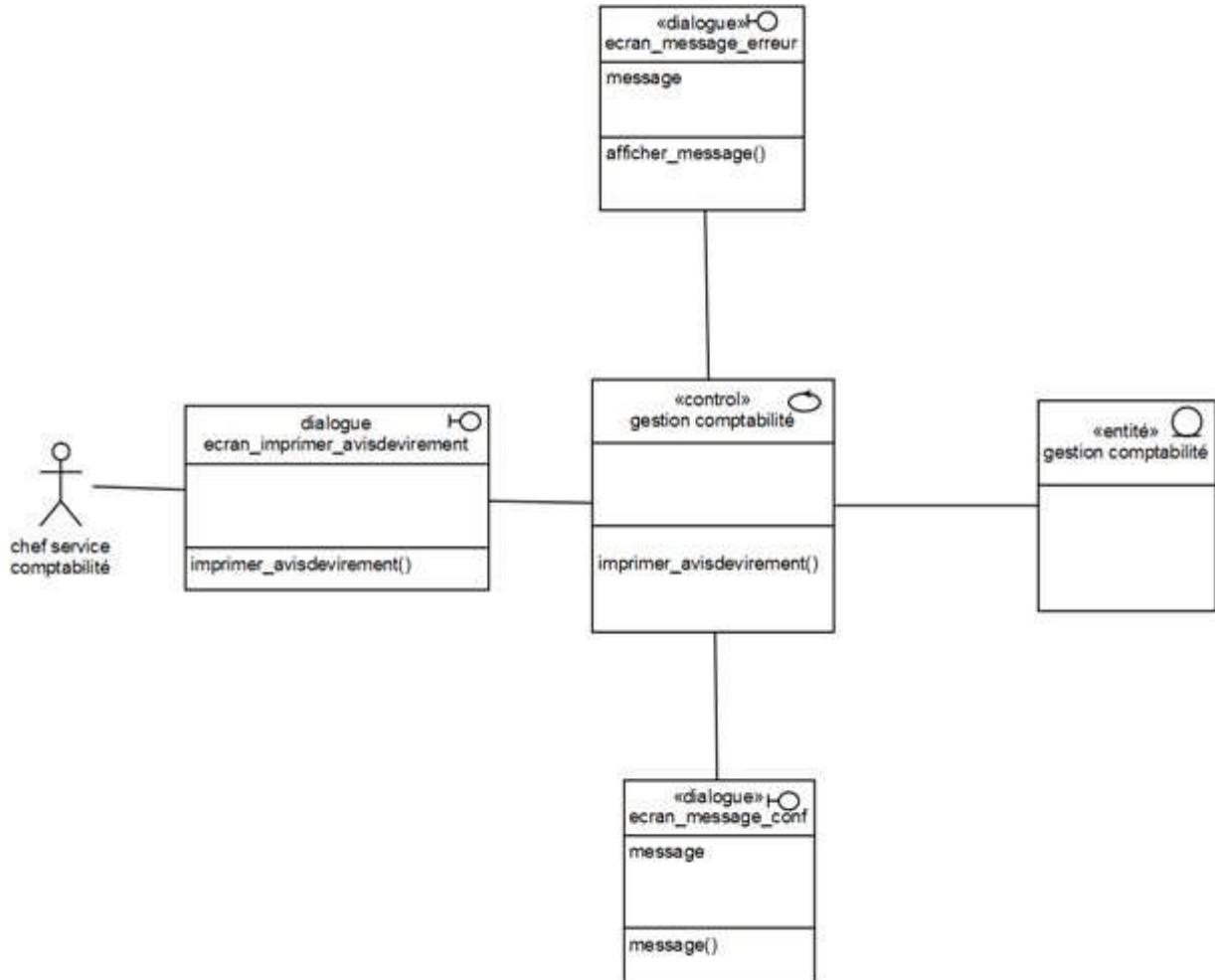
2.8. Imprimer ATS



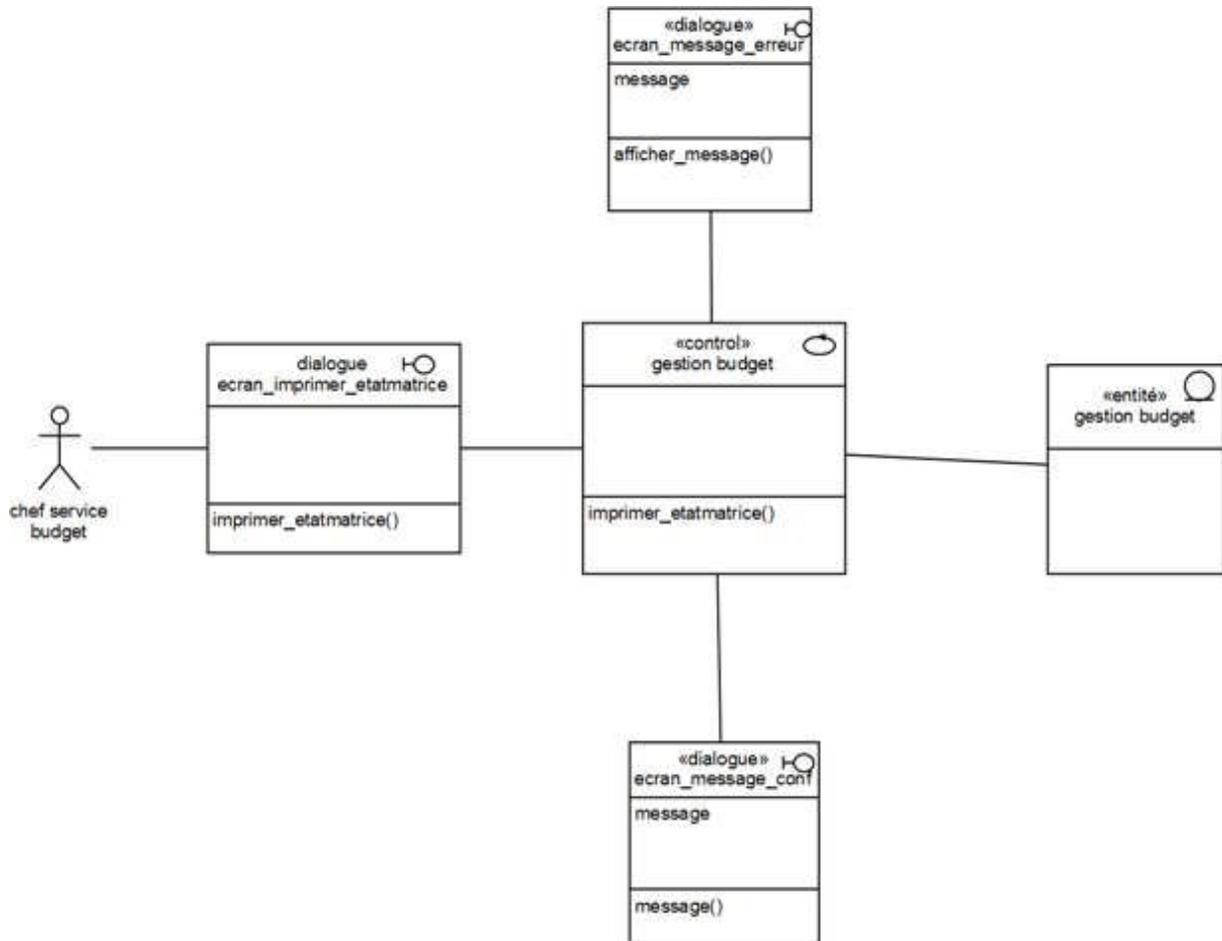
2.9. Imprimer ordre de virement



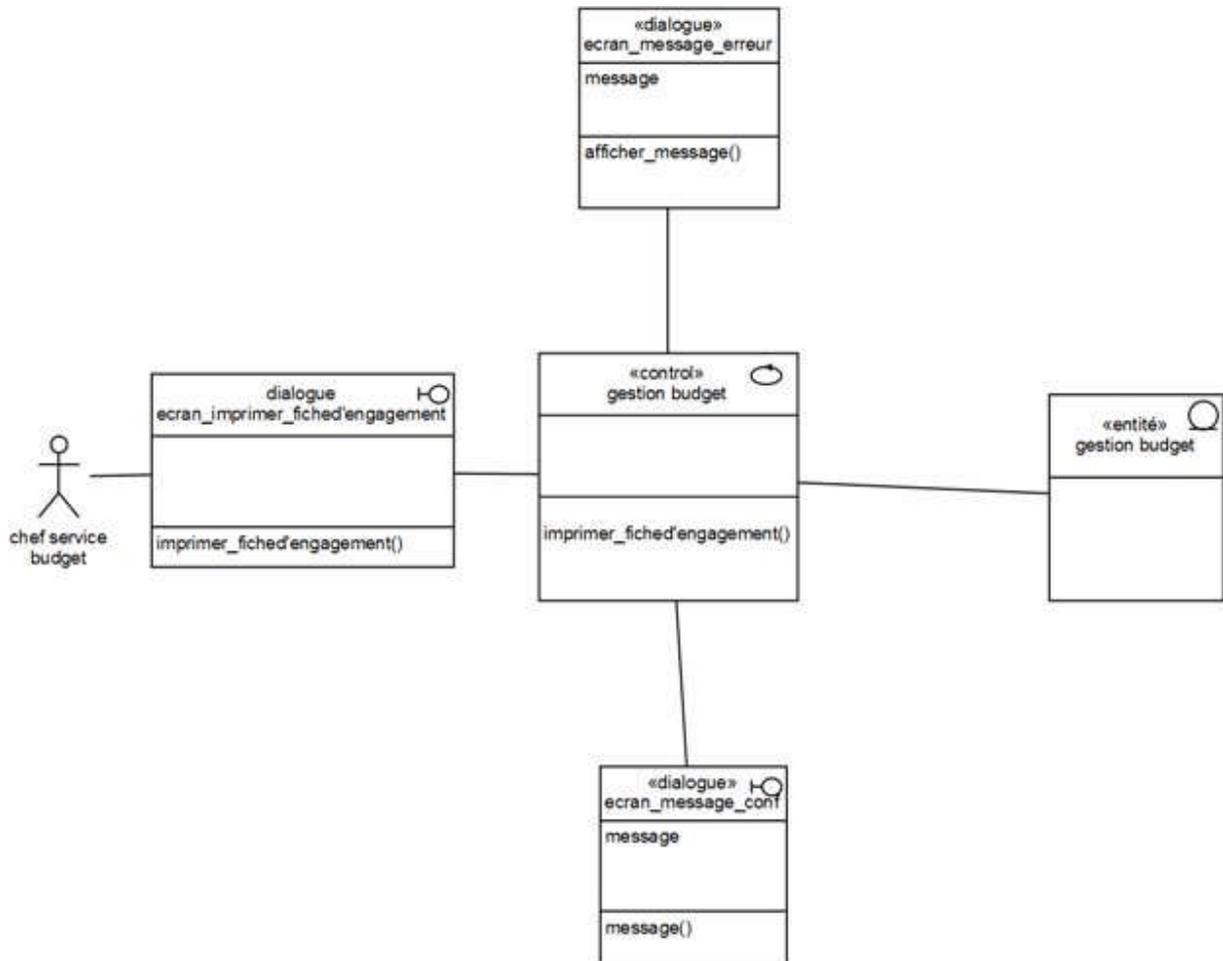
2.10. Imprimer avis de virement



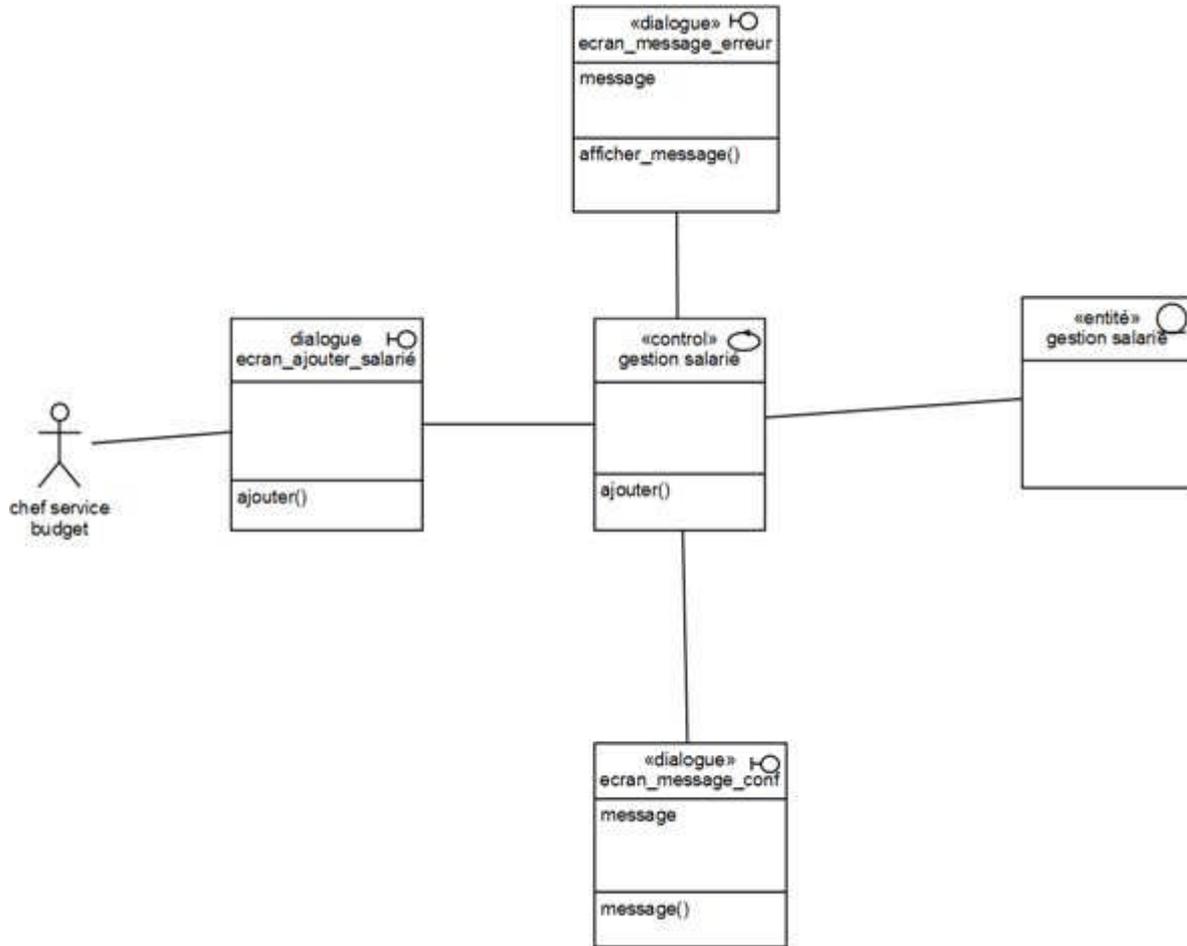
2.11. Imprimer état matrice



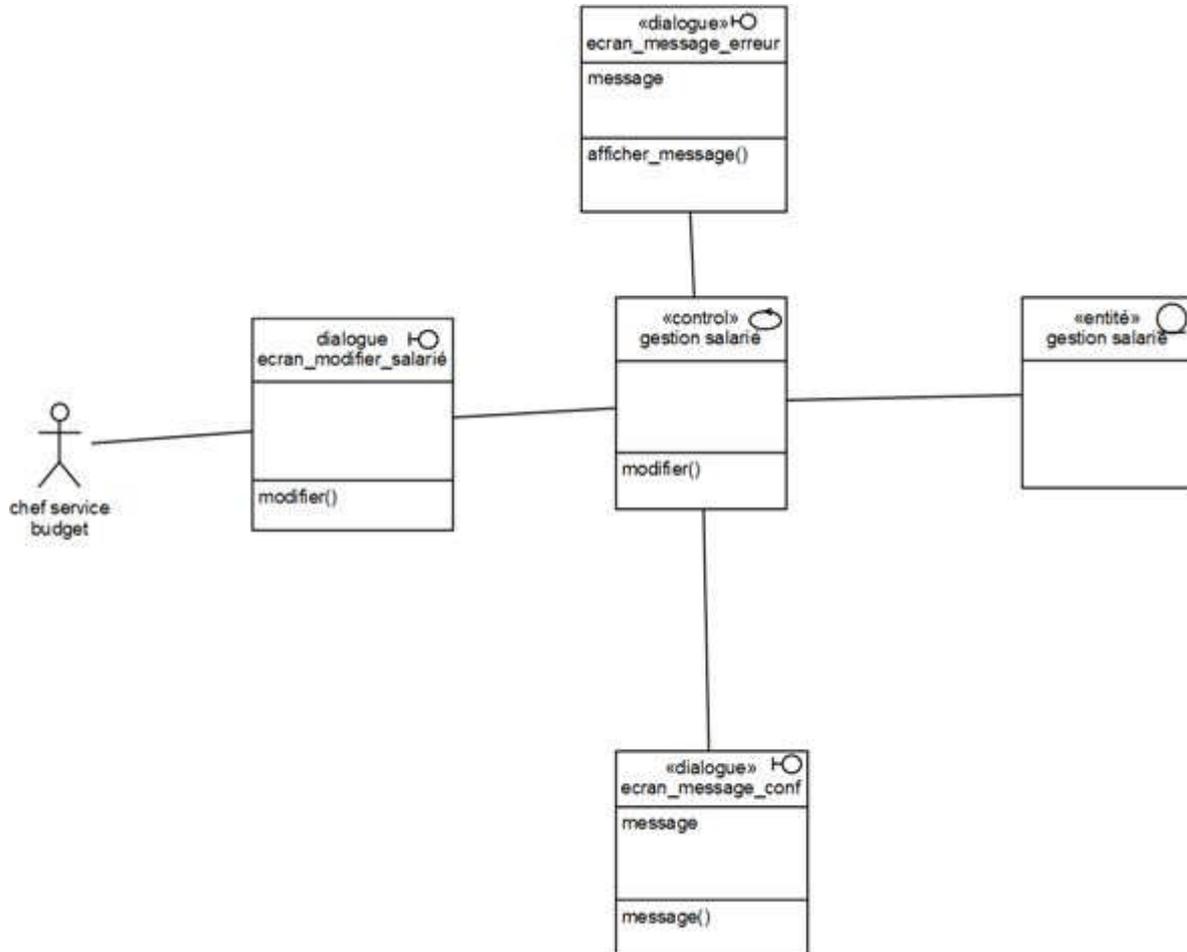
2.12. Fiche d'engagement



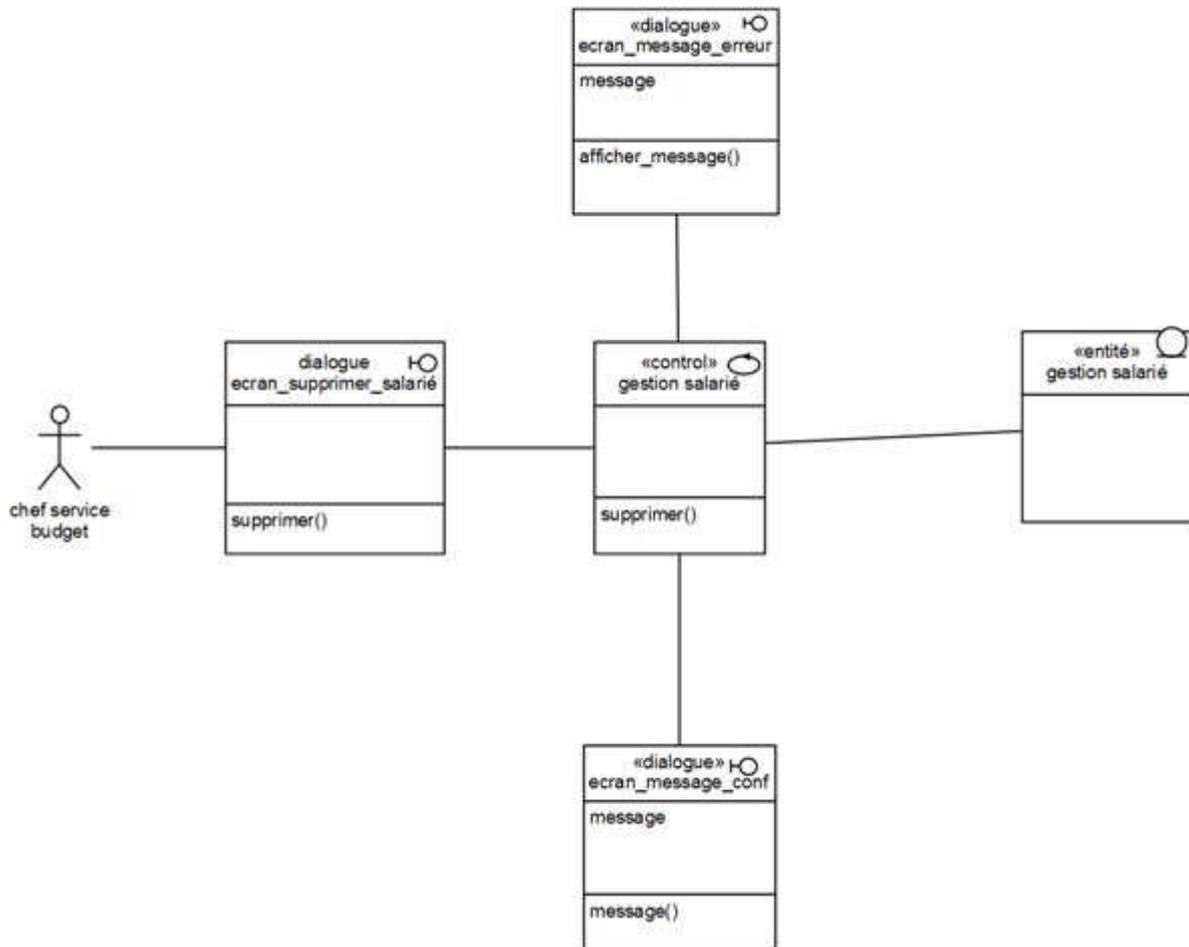
2.13. Ajouter salarié



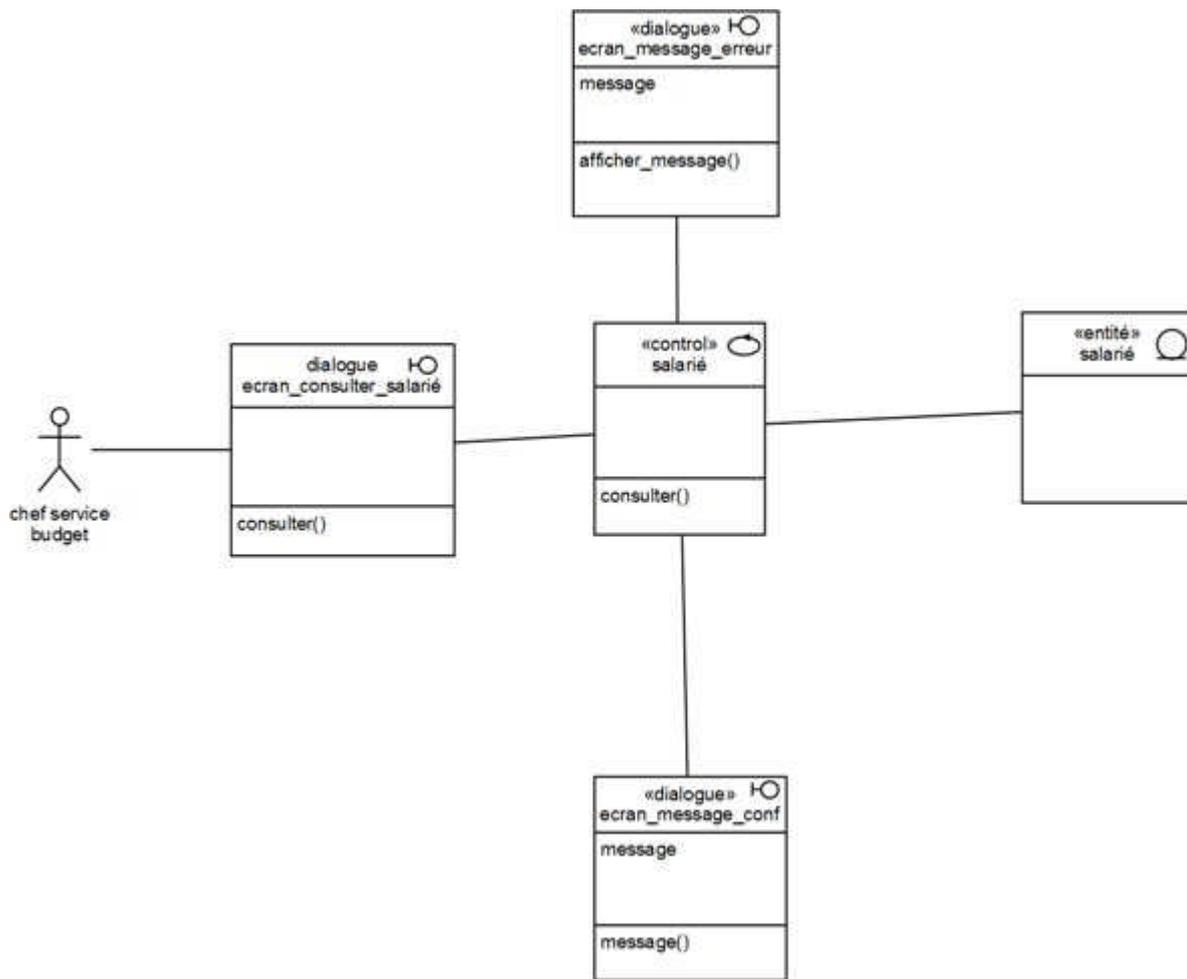
2.14. Modifier salarié



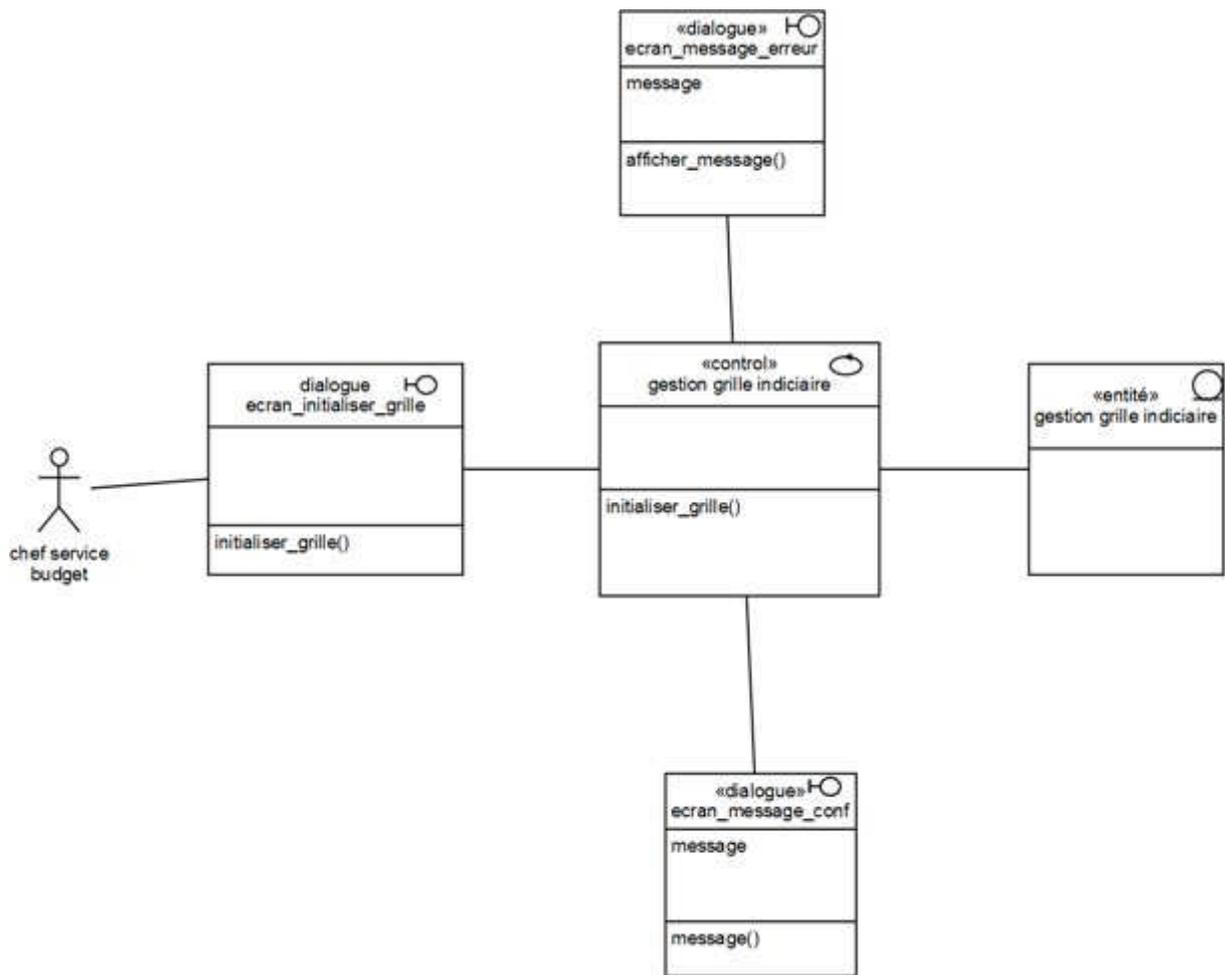
2.15. Supprimer salarié



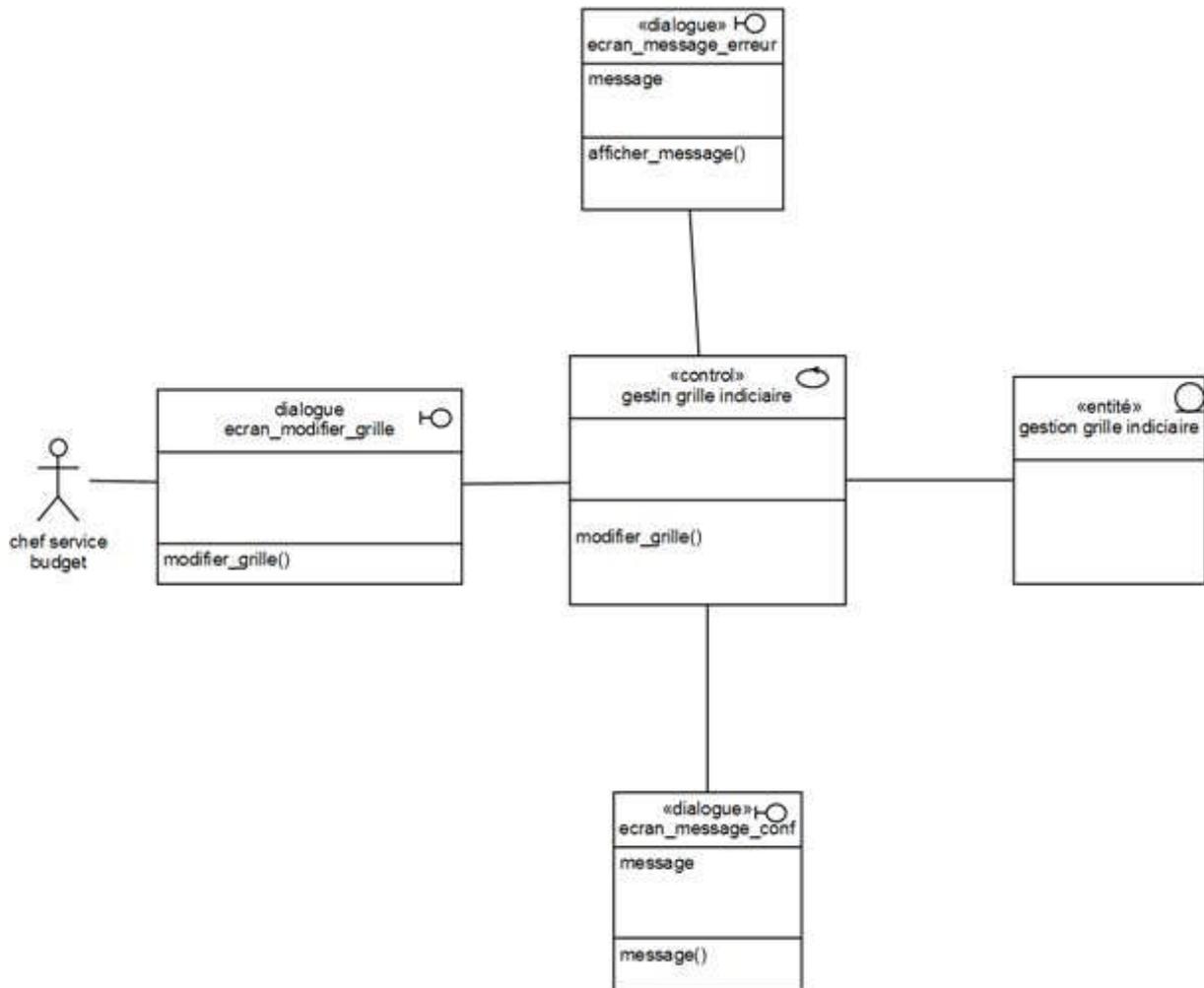
2.16. Consulter salarié



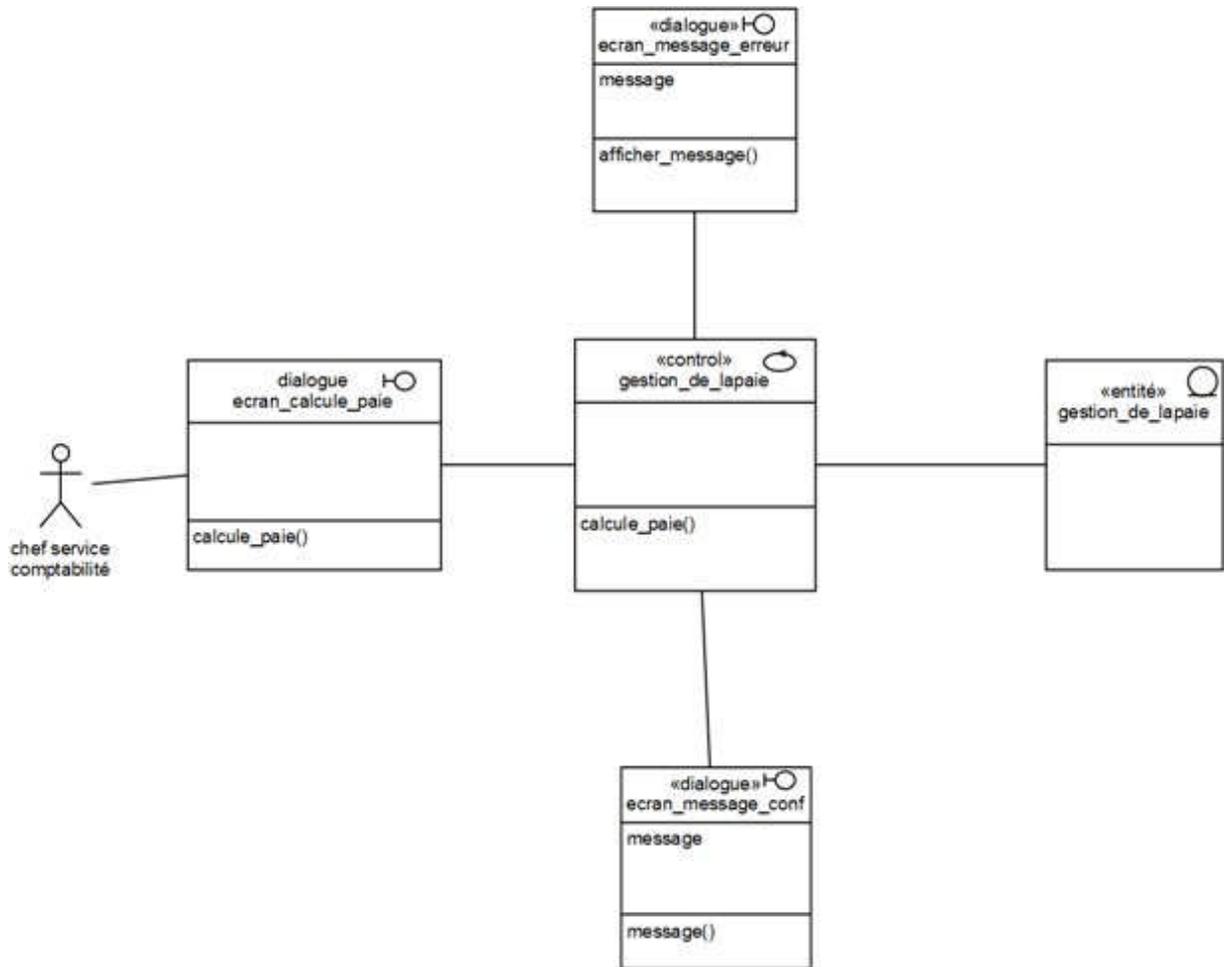
2.17. Initialiser la grille



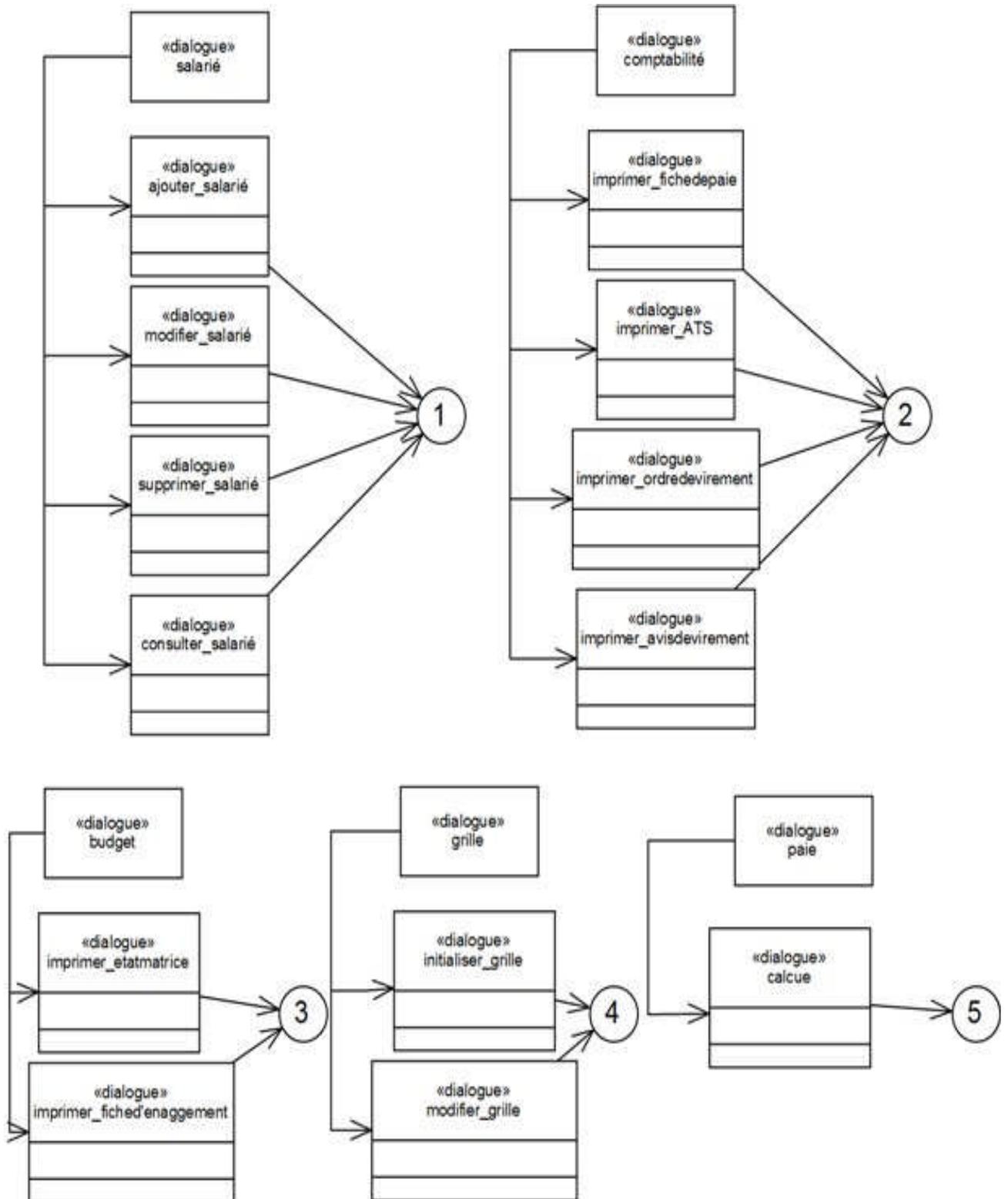
2.18. Modifier grille

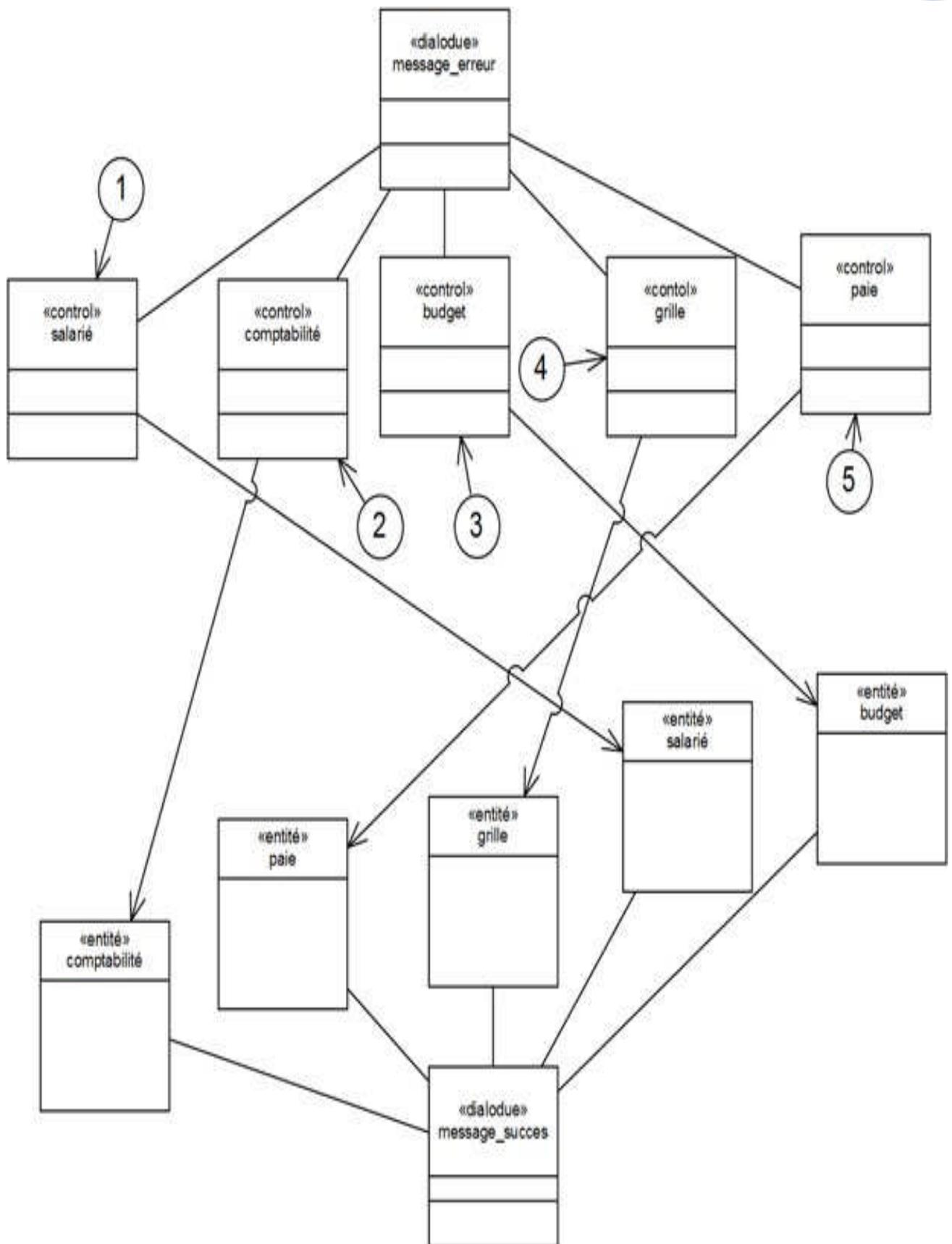


2.19. Calcul de la paie



### 3. Diagramme de class





## *Chapitre 05 : implémentation*

### *Introduction*

*1. représentation de notre base de données*

*2. le langage de programmation  
DELPHI 7*

*3. la base de données ACCES 2010*

*4. l'interface graphique*

*Conclusion*

- Introduction

Après la réalisation de l'étape d'analyse, il faut passer à la dernière phase de la démarche utilisé à savoir la phase d'implémentation. Dans cette partie de projet nous présentons d'abord les outils utilisés pour le développement de l'application. Ensuite, nous allons préparer les données nécessaires pour l'application, en convertissant les classes entités en une base de données. Enfin, nous décrivons quelques interfaces du logiciel réalisé utilisant l'environnement de développement Delphi 7 avec Accès comme un SGBD.

1. représentation de notre base de donnés

Relation	Propriété	Signification	Type	Taille
Comptabilité	<u>nom_ser</u>	Nom du service	texte	500
	somme_salaire	Somme du salaire	Monnaie	500
	salaire_ancien	ancien salaire	Monnaie	500
	salaire_nouveau	nouveau salaire	Monnaie	500
	nbr_sal	Nombre du salarié	Monnaie	500
Budget	<u>nom_ser</u>	Nom du service	Texte	500
	date_d_eng	Date début engagement	Date	500
	date_f_eng	Date fin engagement	Date	500
	durre_d'eng	Durée d'engagement	Date	500
Salarié	<u>num_sal</u>	Numéro du salarié	numérique	500
	nom_s	Nom du salarié	texte	500

	pre_s	Prénom du salarié	texte	500
	adresse_s	Adresse du salarié	texte	500
	date_N	Date naissance du salarié	date	500
	num_s_s	Numéro de sécurité sociale du salarié	texte	500
	Grade	Grade du salarié	Texte	500
	Sexe	Sexe du salarié	Texte	500
	date_rct	Date recrutement du salarié	Date	500
	situation_f	Situation familiale du salarié	Texte	500
	mode_paiement	Mode de paiement du salarié	Texte	500
Paie	<u>Mois</u>	Mois de payement	Texte	500
	annee	Année du payement	numérique	500
	nbr_jour	Nombre de jour travaillé	Numérique	500
	salaire_sal	Salaire salarié	monnaie	500
Grade	<u>Intitulé</u>	Intitulé du grade	Texte	500
	Nom	Nom du garde	Texte	500
	Garde	Grade	Texte	500
Grille	<u>num case grl</u>	Numéro de la case dans la grille	Numérique	500
	categorie	Catégorie du salarié dans	Numérique	500

		la grille		
	echelon	Echelon du salarié dans la grille	Numérique	500
	groupe	Groupe du salarié dans la grille	Numérique	500
	val_ind	Valeur indiciaire du salarié dans la grille	Numérique	500

## 2. le langage de programmation DELPHI 7

### 2.1. Définition

Delphi est un système de développement créé par Borland. Il permet de créer des applications fenêtrées directement exécutables, il utilise le langage Pascal Orienté Objet.

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.

### 2.2. Les avantages de Delphi 7

Ce langage est bien structuré, d'une difficulté moyenne, il donne des applications rapides. Ce langage offre beaucoup de possibilités, les exécutables sont très rapides. Il est utilisé par les professionnels. Il existe en version 1 pour Windows 3.x, en version 2 et 3 pour Windows 95 et NT.

La quasi-totalité des composants Windows (boutons, boîtes de dialogue, menus, barres d'outils...) permettant de créer facilement divers types d'applications et de librairies.

### 3. la base de données ACCES 2010

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.

#### 3.1. avantage accès 2010

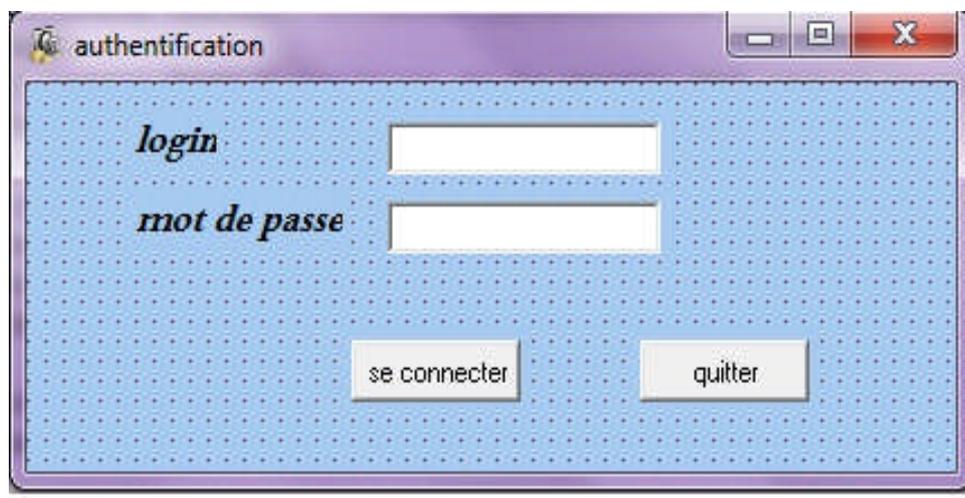
- Obtenir de meilleurs résultats plus vite grâce à l'interface utilisateur office fluent,
- Démarrer en un clin d'œil à l'aide de solutions prédéfinies ;
- Créer plusieurs états avec différentes vues des mêmes informations ;
- Créer des tables rapidement sans se soucier de la complexité des bases de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

## 4. l'interface graphique

Dans cette étape nous allons représenter quelque fenêtre de notre application

### 4.1. Fenêtre authentification

Pour entrer a notre application il faut s'authentifier on entre un login et un mot de passe.



## 4.2. Fenêtre gestion de la paie

Définit la page d'accueil de notre projet.



### 4.3. Fenêtre ajouté salarié

Dans cette page on fait l'ajout du salarié à notre base de données.

ajouter

**Numéro du salarié**

**Date de recrutement**

**Nom**

**Situation familiale**

**Prénom**

**Grade**

**Date de naissance**

**Mode de paiement**

**Adresse**

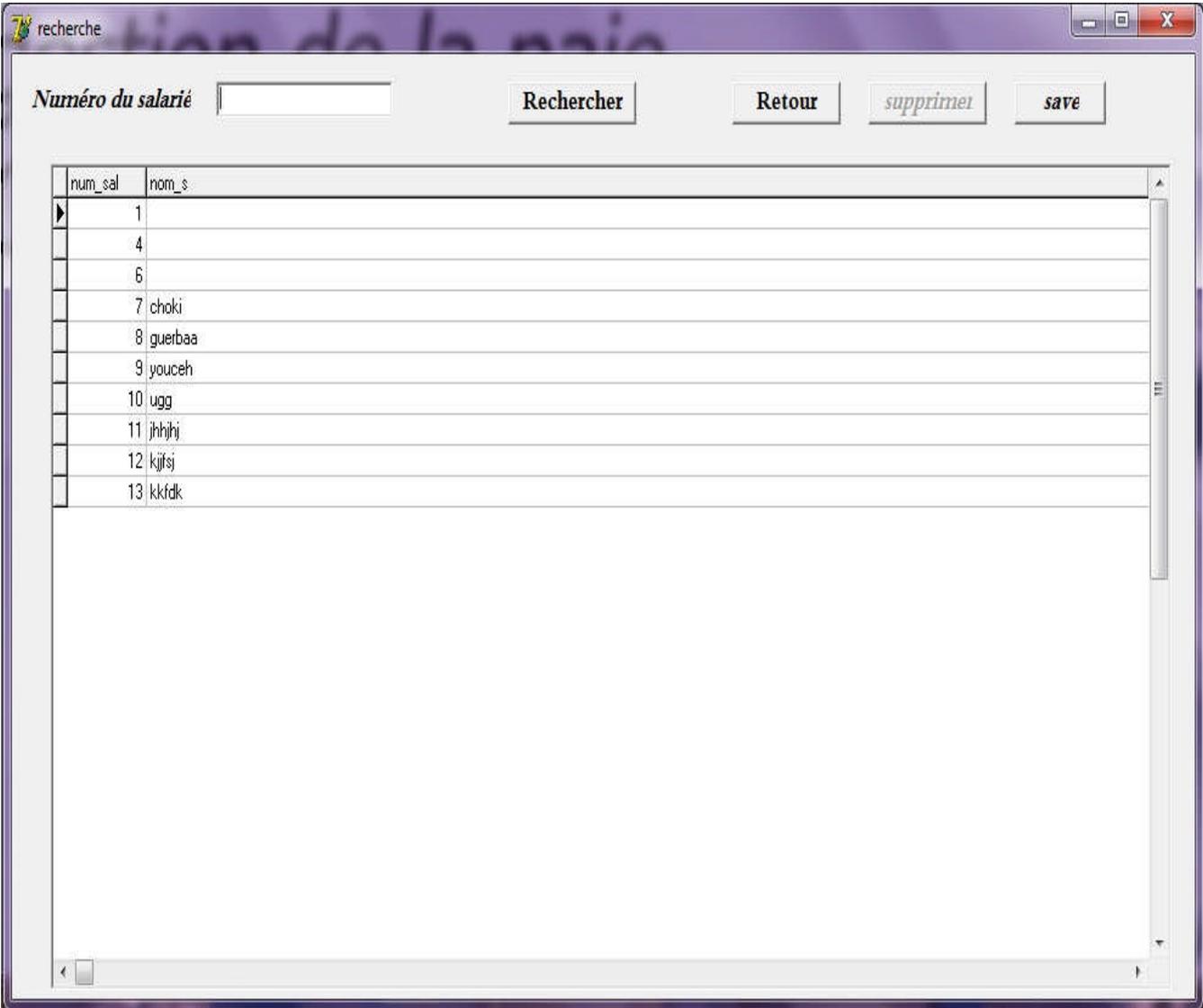
**Numéro sécurité sociale**

**Sexe**

num_sal	nom_s
1	
2	
3	
5	
4	
6	

#### 4.4. Fenêtre modifié salarié

Dans cette page on fait la modification du salarié dans notre base de données.



recherche

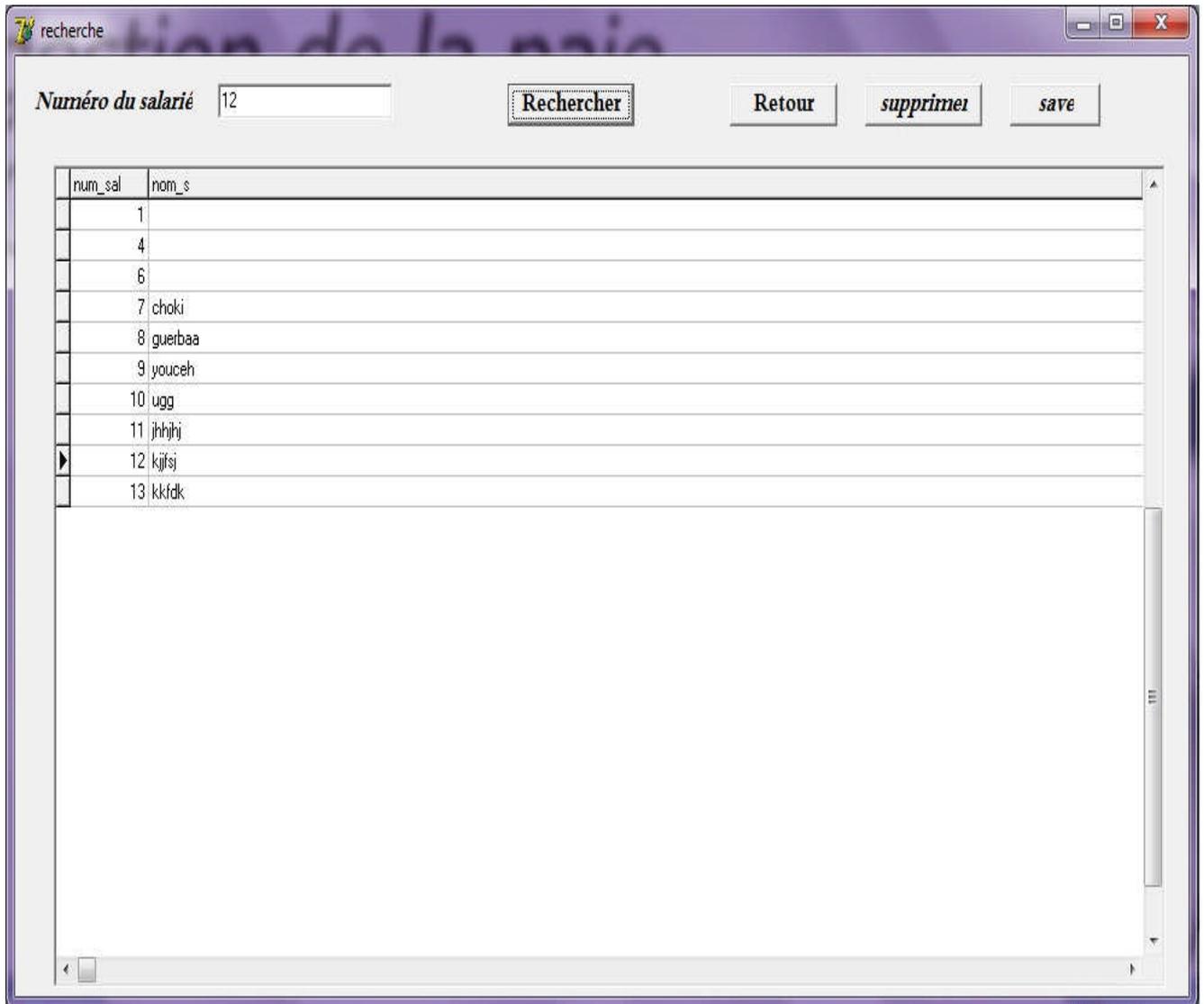
Numéro du salarié

Rechercher Retour *supprimer* save

num_sal	nom_s
1	
4	
6	
7	choki
8	guerbaa
9	youceh
10	ugg
11	jhhhj
12	kjfsj
13	kkfdk

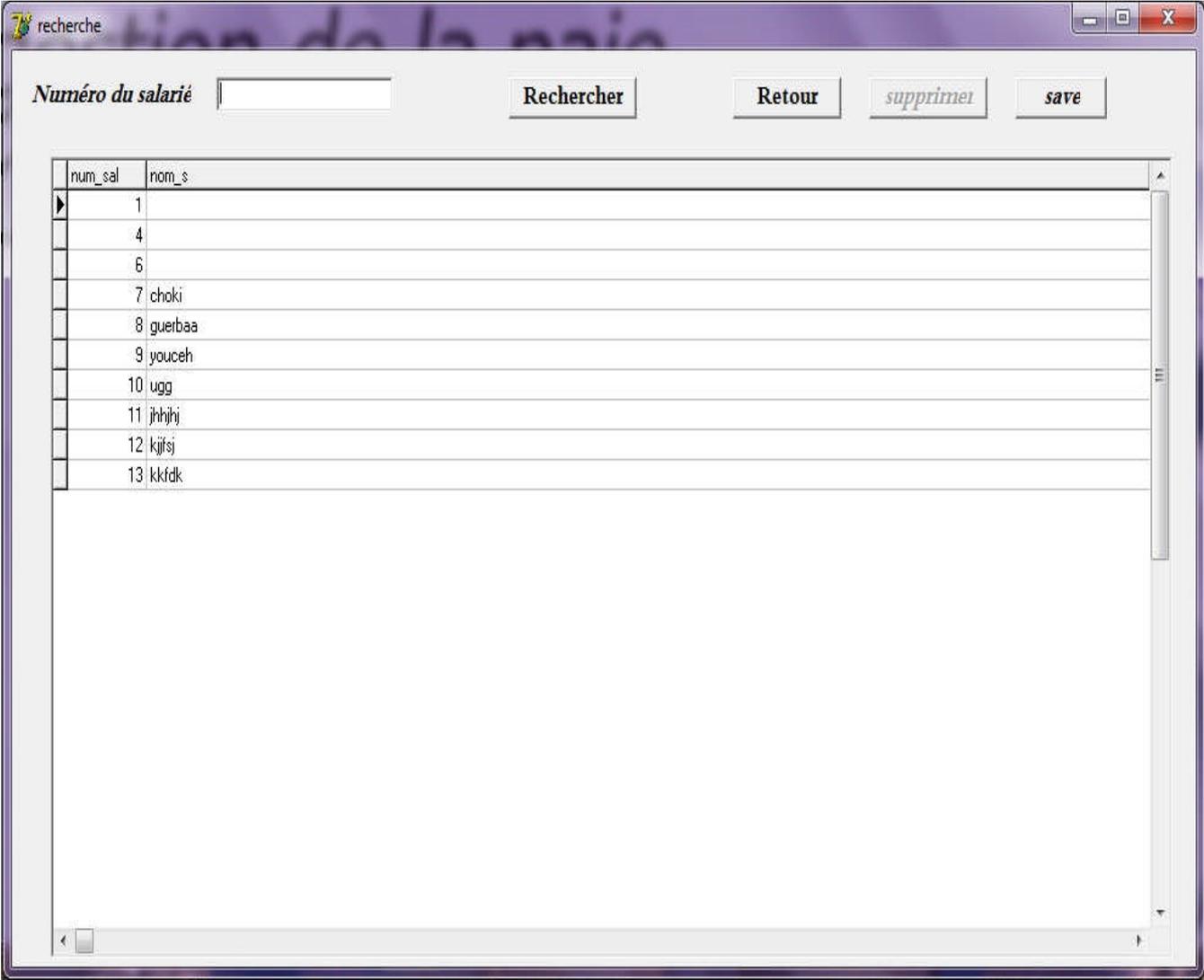
#### 4.5. Fenêtre supprimé salarié

Dans cette page on fait la suppression du salarié dans notre base de données.



#### 4.6. Fenêtre consulté salarié

Dans cette page on fait la consultation des salariés trouvés dans notre base de données.

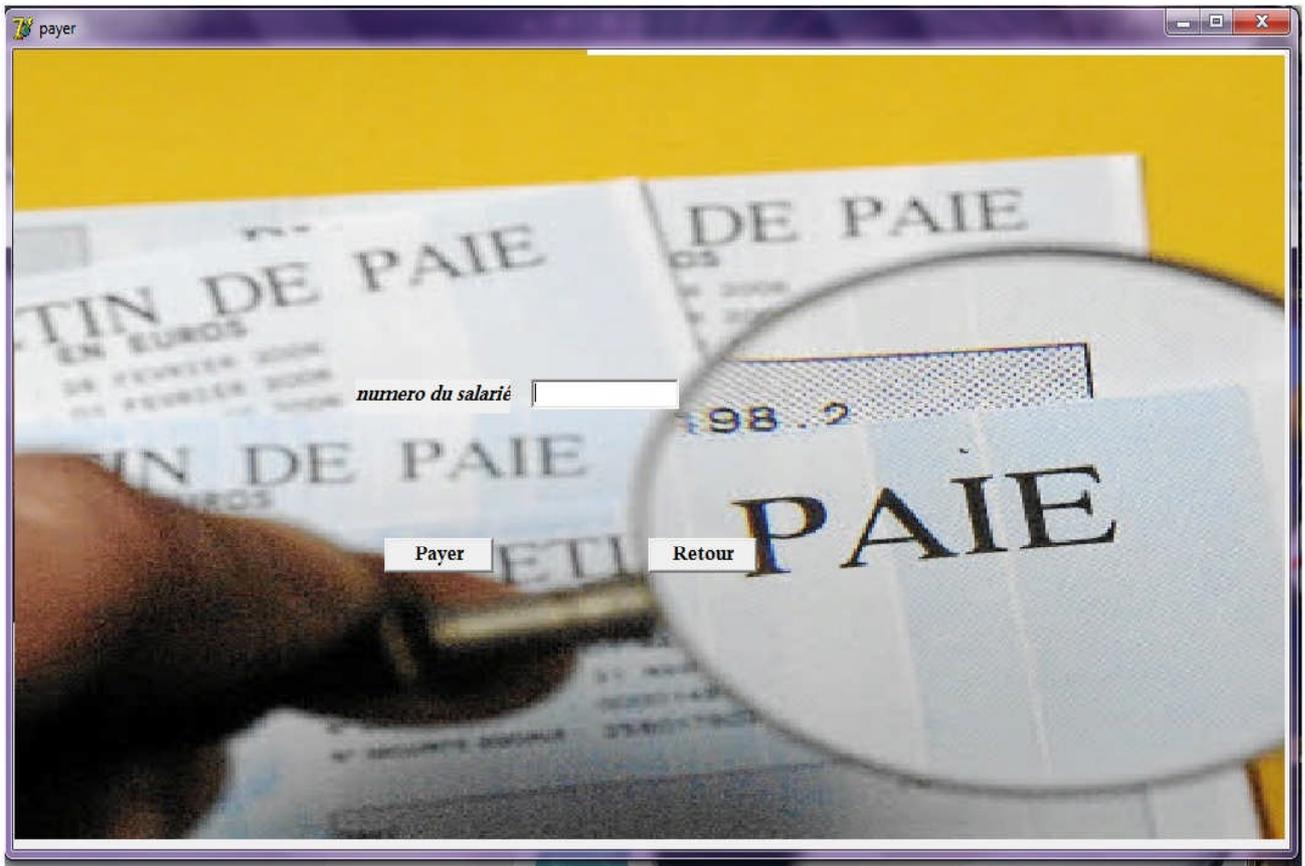


The screenshot shows a window titled "recherche" with a search interface. At the top, there is a text input field labeled "Numéro du salarié" and four buttons: "Rechercher", "Retour", "supprimer", and "save". Below the input field is a table with two columns: "num\_sal" and "nom\_s". The table contains the following data:

num_sal	nom_s
1	
4	
6	
7	choki
8	guerbaa
9	youceh
10	ugg
11	jhhhj
12	kjjsj
13	kkfdk

#### 4.7. Fenêtre payé salarié

Dans cette page on fait le paiement des salariés dans notre base de données.



#### 4.8. Fenêtre imprimer fiche de paie

Dans cette page on fait l'impression de fiche de paie du salarié.

The screenshot shows a window titled "imprimer fiche de paie". At the top, there is a label "numero du salarié" followed by an input field, a "valider" button, and a "retour" button. Below this, a dashed box contains the header information: "République Algérienne Démocratique et populaire", "ministre de l'enseignement supérieur et de recherche scientifique", and "centre universitaire de mila". Underneath, the text "En-tête de page" is visible. The main title of the slip is "FICHE DE PAIE", followed by "Titre" and three input fields: "[mois de]", "[mois]", and "[année]". A large rectangular box contains the following fields: "[Nom]", "[nom\_s]", "[Prenom]", "[pre\_s]", "[Grade]", "[grade]", "[Situation familiale]", "[situation\_f]", "[Categorie]", "[categorie]", "[Echelon]", "[echelon]", "[Indice]", "[val\_ind]", and "[Mode de paiement]", "[mode\_paiement]". On the left side of the window, there are three small icons: a printer, a document, and a help icon.

#### 4.9. Fenêtre imprimer avis de virement

Dans cette page on fait l'impression de l'avis de virement du salarié.

The screenshot shows a window titled "imprimer avis de virement". At the top, there is a label "numero du salarié" followed by an input field, a "valider" button, and a "retour" button. In the center, a dashed box contains a form for the transfer notice. The form includes a "Titre:" field with the text "AVIS DE VIREMENT", a "Détail:" section with "NET A PAYER" and an input field, and fields for "[nom]", "[prenom]", "Compte N°", and "CLE". At the bottom, there is a table with three columns: "BRUT", "INDICE", and "SEC SOC", each with an empty input field below it.

BRUT	INDICE	SEC SOC

#### 4.10. Imprimer ordre de virement

Dans cette page on fait l'impression d'ordre de virement du salarié.

numero du salarié

**ORDRE DE VIREMENT**  
En-tête de

NET A PAYER  
 DA

COMpte N °  CLE

Détail

<input type="text"/> visa de c/c	<input type="text"/> imbre a date
----------------------------------	-----------------------------------

- **Conclusion**

Dans ce chapitre, nous avons décrit brièvement la phase d'implémentation ainsi que l'application que nous avons réalisée. En premier lieu nous avons présenté les interfaces de l'application avec toutes les options qu'elles contiennent, pour faciliter son utilisation.

## Référence bibliographie

1	Mémoire de fin d'étude gestion de la paie université Mentouri Constantine année 2008
2	Uml2 en action de l'analyse des besoins a la conception
3	Uml 2, Laurent AUBERT, édition 2007-2008
4	<a href="http://fr.wikipedia.org">http://fr.wikipedia.org</a>
5	<<Conception et implémentation d'un outil de délibération>> Benmakhlouf amal, laieb Karima, centre universitaire Mila 2011-2012
6	<a href="http://www.commenrcamarche.com/">http://www.commenrcamarche.com/</a>

