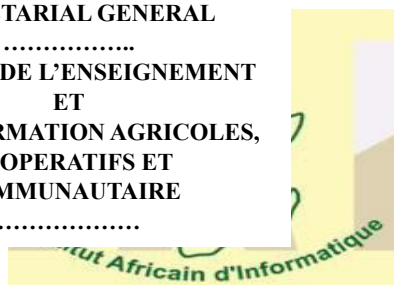


REPUBLIQUE DU CAMEROUN
Paix – Travail – Patrie
.....
MINISTRE DE L'AGRICULTURE
ET
DU DEVELOPPEMENT RURAL
.....
SECRETARIAL GENERAL
.....
DIVISION DE L'ENSEIGNEMENT
ET
DE LA FORMATION AGRICOLES,
COOPERATIFS ET
COMMUNAUTAIRE
.....



REPUBLIC OF CAMEROON
Peace – Work – Fatherland
.....
MINISTRY OF AGRICULTURE AND
RURAL DEVELOPMENT
.....
GENERAL SECRETARIAT
.....
AGRICULTURAL COOPERATIVE
AND
COMMUNITY EDUCATION AND
TRAINING DIVISION
.....
AGRICULTURAL TECHNICAL SCHOOL
OF

**THEME : APPLICATION DE
LOCALISATION LOCALISATION DE**

NOM ET PRENOM

TCHOUNANG WILFRIED
NAHIMI DOLLO
BAHANI THIERY
KONSO ANNE BEATRICE
TCHAFAC LOIC
SALIM SOULEYMANE

EXAMINATEUR :

M. FOMEKONG

INTRODUCTION

Une application mobile plus communément appeler « app » est un type de logiciel d'application conçus pour fonctionner sur les appareils mobiles sans file , tel un smartphone ou une tablette informatique . les application mobiles servent souvent a fournir aux utilisateurs des services similaire a ceux auxquels il accède sur un ordinateur. de ce fait , les app mobiles offres cet avantage d'être utilisable ou que l'on soit a condition d'être avec son appareil mobile . c'est dans cette logique que nous avons opté pour la création d'une application de l'localisation des transformateurs des différents régions du Cameroun .dans l'optique de mener a bien notre étude , notre travail a été base uniquement sur la phase technique consacre a la mise en exergue des processus a suivre pour concevoir et déployer notre plat forme .

DEVELOPEMENT

I. OBJECTIFS

1. OBJECTIF GLOBALE

L'objectif global de ce projet est de pouvoir localiser les transformateurs de manière spécifique et d'avoir toute l'information les concernant dans toutes les régions du Cameroun

2. OBJECTIF SPECIFIQUE

De manière spécifique, il sera question pour nous de concevoir une application mobile pouvant :

- Permettre au chef de station de localiser, ajouter, modifier, supprimer un transformateur tout en s'authentifiant
- Permettre au directeur général de non seulement gérer les chefs de station mais aussi de localiser, ajouter, modifier, supprimer un transformateur
- Permettre à l'entreprise Eneo d'assurer la gestion efficace et la sécurité de leurs équipements

II. EXPRESSION DES BESOINS

L'application à réaliser doit satisfaire les besoins fonctionnels qui constituent les fonctionnalités que l'application doit offrir et les besoins non fonctionnels qui perfectionnent l'application

1. Besoin fonctionnels

L'application à réaliser doit offrir un ensemble de fonctionnalités qui doivent être mises en relation avec un ensemble de besoins utilisateurs. Ces derniers définissent les services auxquels les utilisateurs s'attendent à voir fournis par cette application. La présente plateforme doit satisfaire les besoins fonctionnels suivants :

- La gestion de l'authentification : notre application doit authentifier les utilisateurs par un login pour accéder aux différentes fonctionnalités
- La gestion d'une agence de région donnée : notre application doit permettre à un chef d'agence donnée d'ajouter, consulter, modifier et supprimer un transformateur
- La gestion des chefs d'agences : notre application doit permettre au directeur générale de modifier un chef d'agence

2. Besoin non fonctionnel

Un besoin non-fonctionnel est une contrainte liée à l'environnement, l'implémentation et les exigences en matière de performances.

Dans le cadre de ce travail, l'application devra répondre à ces besoins :

- **La sécurité** : Tous les accès des utilisateurs doivent être protégés par un login et un mot de passe
- **L'intégrité** : Garantir l'intégrité et la cohérence des données ;
- **La performance** : Pouvoir supporter des grandes quantités de données et le temps de réponse des requêtes doit être court ;
- **La fiabilité** : Les données fournies par l'application doivent être fiables et l'application doit rendre des résultats corrects.
- **Prise en compte de l'expérience de l'utilisateur** : Le système doit respecter le niveau d'expérience de l'utilisateur, donc il doit être simple, compréhensible et facile à manipuler ;

III. PRÉSENTATION DES DIAGRAMMES

1. Diagramme de cas d'utilisation

a) Présentation

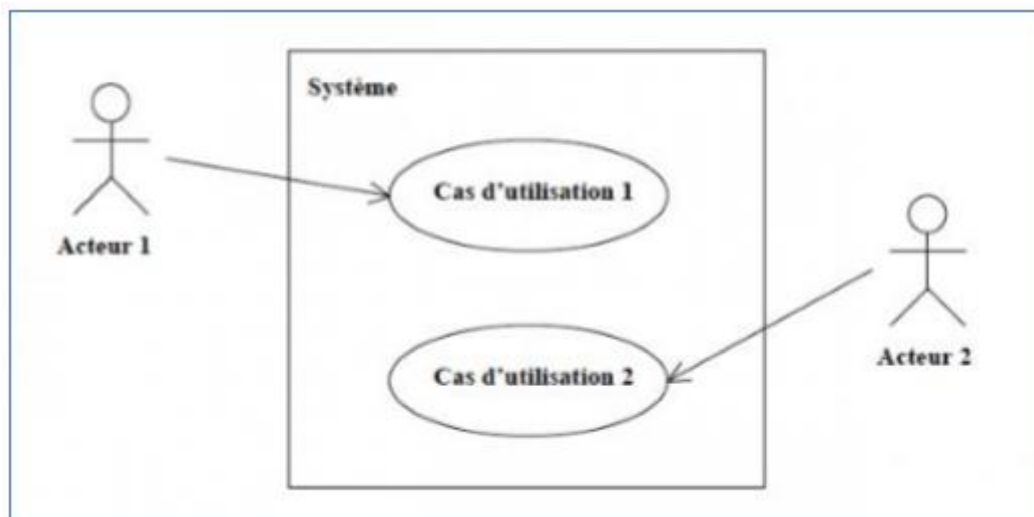
Les diagrammes de cas d'utilisation identifient les fonctionnalités fournies par le système (cas d'utilisation), les utilisateurs qui interagissent avec le système (acteurs), et les interactions entre ces derniers. Les objectifs principaux des diagrammes des cas d'utilisation sont :

- Fournir une vue de haut-niveau de ce que fait le système ;
 - Identifier les acteurs du système ;
 - Déterminer des secteurs nécessitant des interfaces hommes-machine

b) formalisme

Les éléments constitutifs d'un diagramme de cas d'utilisation sont : les acteurs, les cas d'utilisation, et les associations.

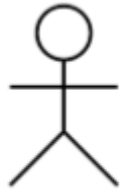
La figure suivante présente de façon graphique le formalisme d'un diagramme de cas d'utilisation :



Acteur : Un acteur est un type stéréotypé représentant une abstraction qui réside juste en dehors du système à modéliser.

En UML, l'acteur n'est pas nécessairement une personne physique, mais toute entité (une personne, un service, une société, un système informatique, bases de données, des équipements...) qui interagit avec le système.

Une même personne physique peut donc être représentée par plusieurs acteurs en fonction des rôles qu'elle joue. Formellement, un acteur est représenté en UML comme suit :

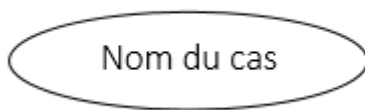


Nom de l'acteur

Cas d'utilisation : Un cas d'utilisation représente une fonctionnalité fournie par le système, typiquement décrite sous la forme Verbe + objet.

Il correspond à un objectif du système, motivé par un besoin d'un ou plusieurs acteurs. L'ensemble des cas d'utilisation (use cases) décrit les objectifs (le but) du système.

Les cas d'utilisation sont représentés par une ellipse contenant leur nom comme illustré ci-après :



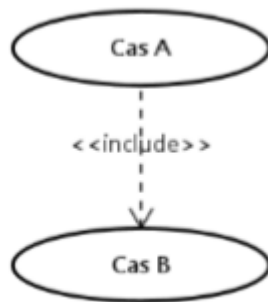
Association : Les associations sont utilisées pour lier des acteurs avec des cas d'utilisation. Elles indiquent qu'un acteur participe au cas d'utilisation sous une forme quelconque. Les associations sont représentées par une ligne reliant l'acteur et le cas d'utilisation.



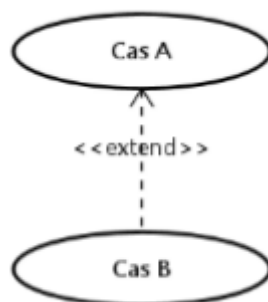
Relation : En UML, les acteurs sont liés à leurs différents cas d'utilisation par des diverses relations qui leur lient. Celles-ci présentent un lien d'association entre le dit

acteur et l'action qu'il mène dans le système. C'est ainsi que l'on distingue les relations suivantes :

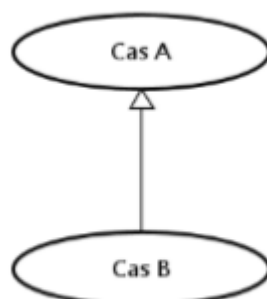
Inclusion : Un cas d'utilisation A inclus un cas d'utilisation B signifie que pour effectuer le cas d'utilisation A, il faut obligatoirement effectuer le cas d'utilisation B



Extension : On dit qu'un cas d'utilisation B étend un cas d'utilisation A si pendant que l'on effectue le cas A on a la possibilité d'effectuer B indépendamment de A



Generalisation : Un cas A est une généralisation d'un cas B si le cas B est une autre manière d'effectuer le cas A (spécialisation).



c) Identification des acteurs et cas d'utilisation

Le tableau suivant présente les différents acteurs de notre système ainsi que les cas d'utilisations auxquels ils ont accès.

Acteurs	Rôles
<ul style="list-style-type: none">• Chef d'agence	Gérer l'agence d'une région en ajoutant, consultant ou suppriment un transformateur
<ul style="list-style-type: none">• Directeur générales	Gérer les chefs d'agence. Il hérite également d'un chef d'agence

d) Diagramme de cas d'utilisation global

e) Diagramme de cas d'utilisation du chef d'agence

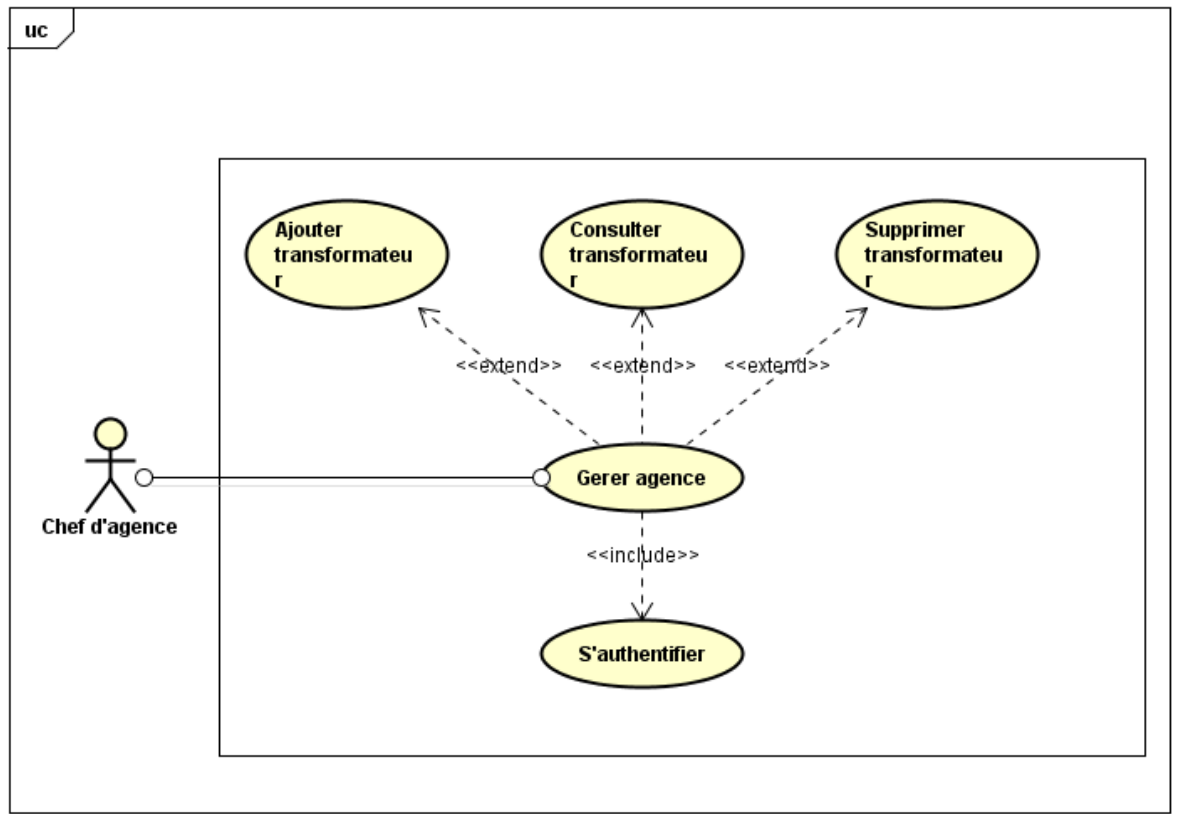
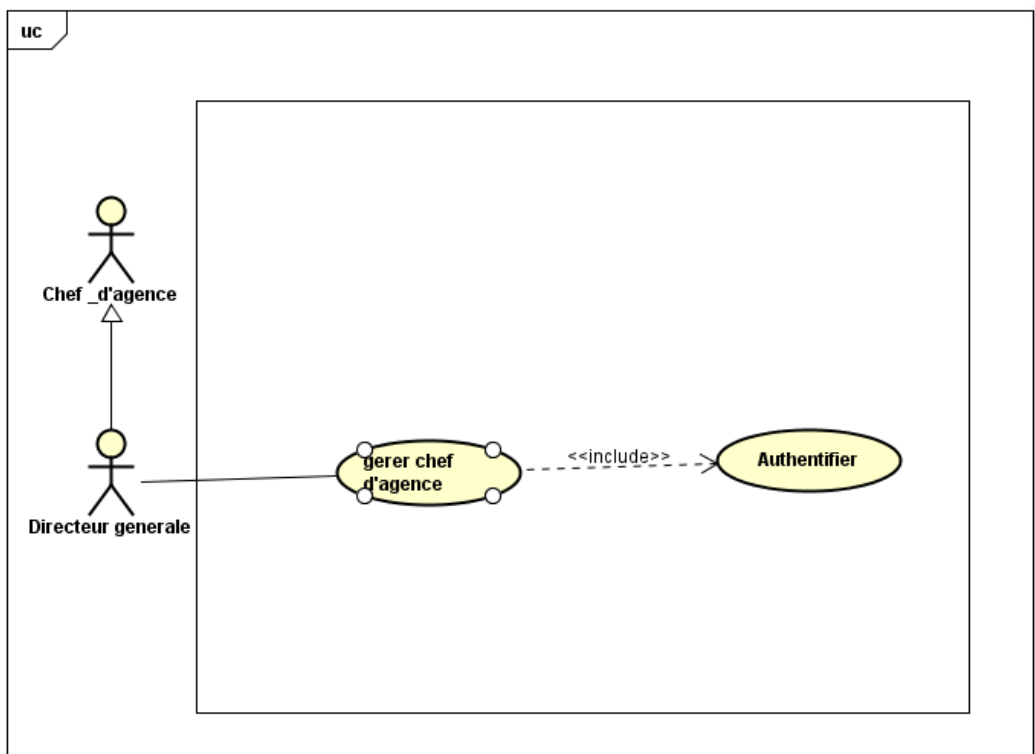


Diagramme de cas d'utilisation du Directeur generale



2. Diagramme de classe

a) Présentation

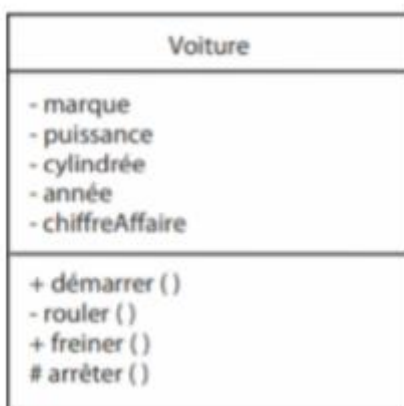
Le diagramme de classe exprime la structure statique du système en termes de classes et de relations entre ces classes. L'intérêt du diagramme de classe est de modéliser les entités du SI. Le diagramme de classe permet de représenter l'ensemble des informations finalisées qui sont gérées par système.

b) Formalisme

Les principaux éléments de ce diagramme sont : les classes, et leurs associations (association, généralisation, agrégation, composition, dépendance). Sa description se fonde sur les attributs, les opérations d'une classe, et les différents types d'associations entre classes.

Nous pouvons décrire ces éléments comme suit :

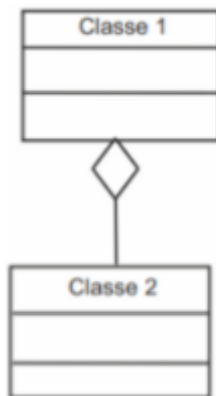
- **Classe** : Une classe décrit un groupe d'objets ayant les mêmes propriétés (attributs), un même comportement (opérations), et une sémantique commune (domaine de définition).
- **Attribut** : Un attribut est une propriété élémentaire d'une classe. Pour chaque objet d'une classe, l'attribut prend une valeur (sauf cas d'attributs multivalués).
- **Opération** : Une opération est une fonction applicable aux objets d'une classe. Une opération permet de décrire le comportement d'un objet. Une méthode est l'implémentation d'une opération.



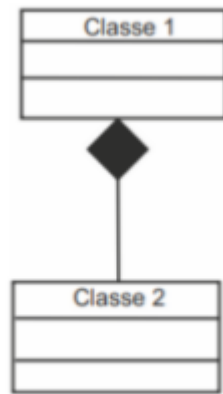
- **Association** : Une association entre classes représente les liens qui existent entre les instances de ces classes.



- **Agrégation** : L'agrégation est une association qui permet de représenter un lien de type « ensemble » comprenant des « éléments ». Il s'agit d'une relation entre une classe représentant le niveau « ensemble » et 1 à n classes de niveau « éléments »



- **Composition** : La composition est une relation d'agrégation dans laquelle il existe une contrainte de durée de vie entre la classe « composant » et la ou les classes « composé ». Autrement dit la suppression de la classe « composé » implique la suppression de la ou des classes « composant ».



c) Diagramme de classe de notre système

3. Diagrammes de séquence

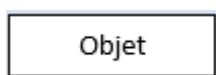
a) Présentation

Le diagramme de séquence nous permet de représenter les interactions entre les utilisateurs et la plateforme de gestion ainsi que la base de données tout en précisant la chronologie des échanges de message. Son but principal est de donner une description chronologique sur le déroulement des cas d'utilisation entre les acteurs et les objets ou objets entre eux.

b) Formalisme

Les éléments constitutifs d'un diagramme de séquence sont : l'objet, l'acteur, la ligne de vie, l'activation, le message, le cadre du diagramme

- **Objet** : Les objets sont les instances de classe et sont rangés horizontalement



- **Acteur** : Un acteur est une personne qui interagit ou communique avec le système et les objets



- **Ligne de vie** : La ligne de vie identifie l'existence de l'objet par rapport au temps

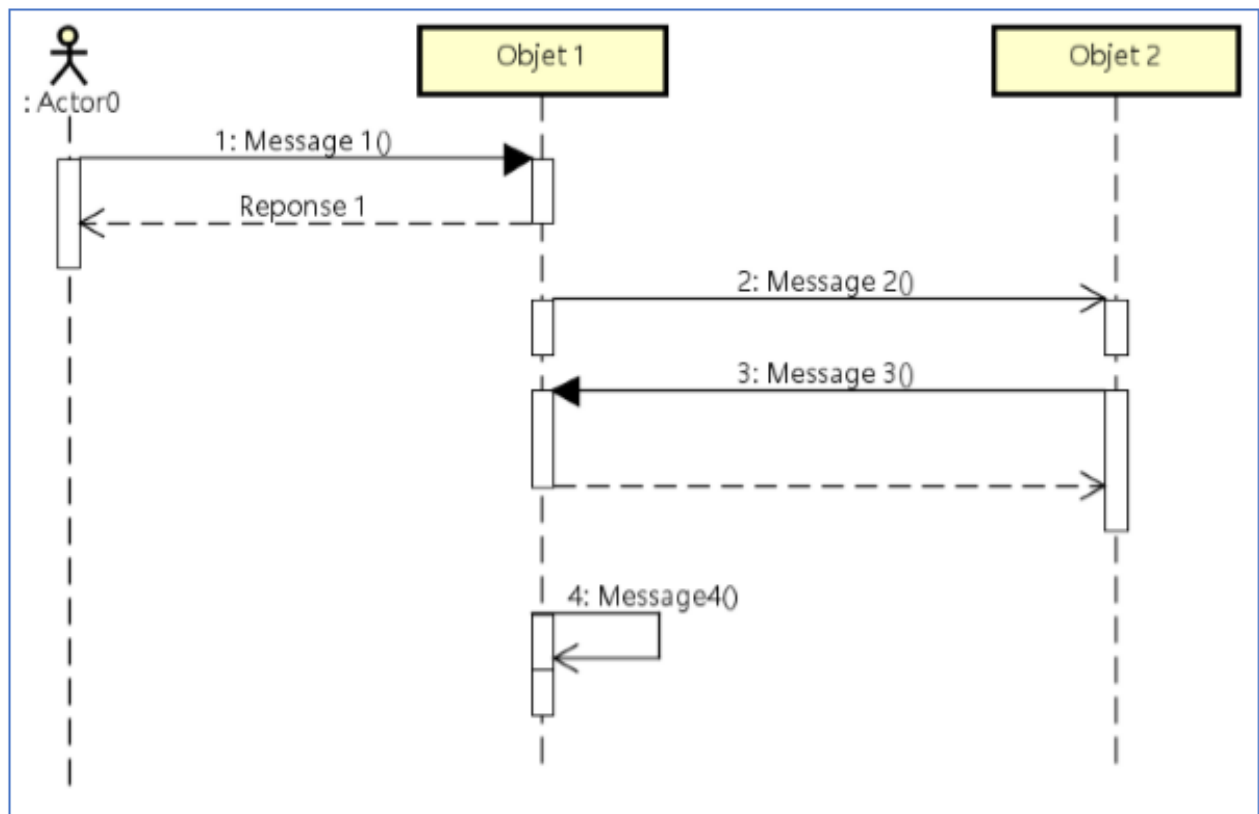


- **Activation** : L'activation représente l'ensemble des opérations exécutées par un objet

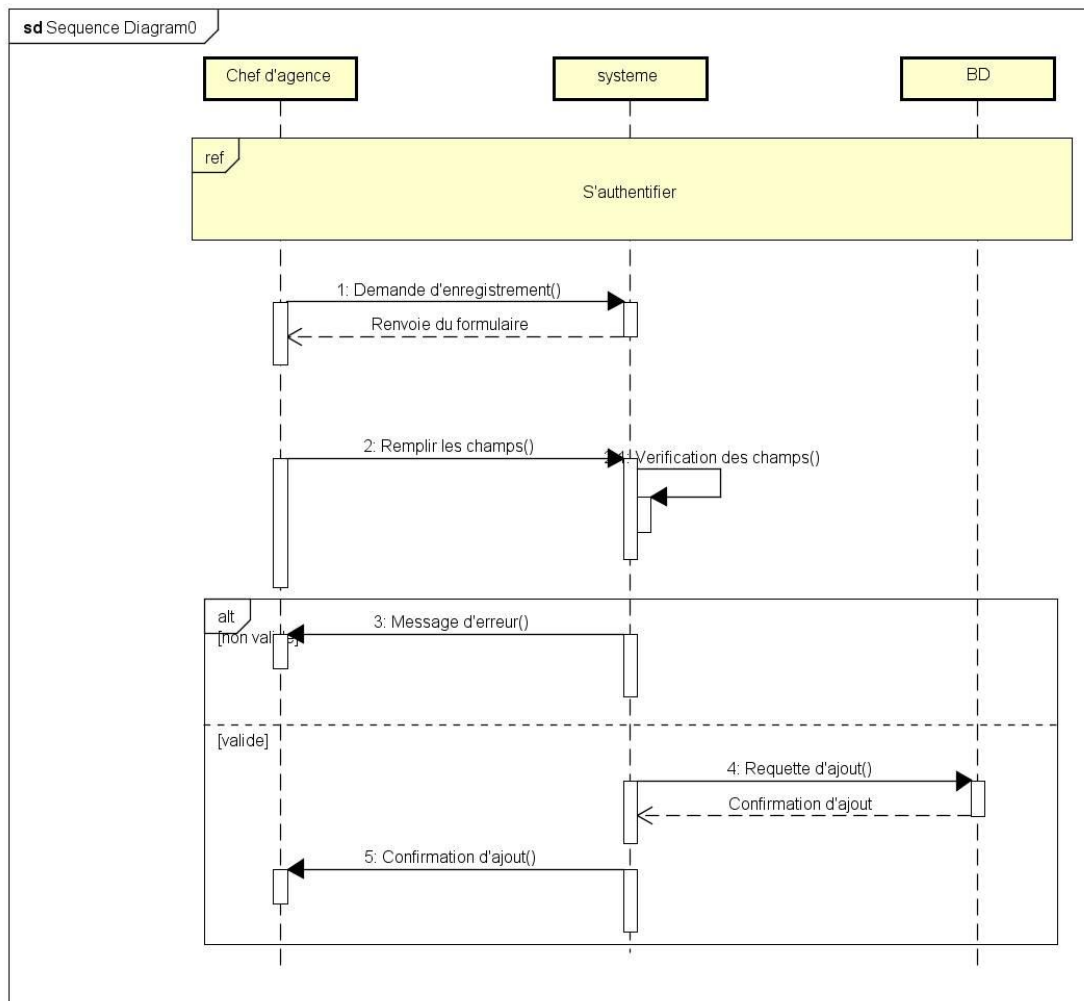


- **Message synchrone** : c'est une communication où l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions

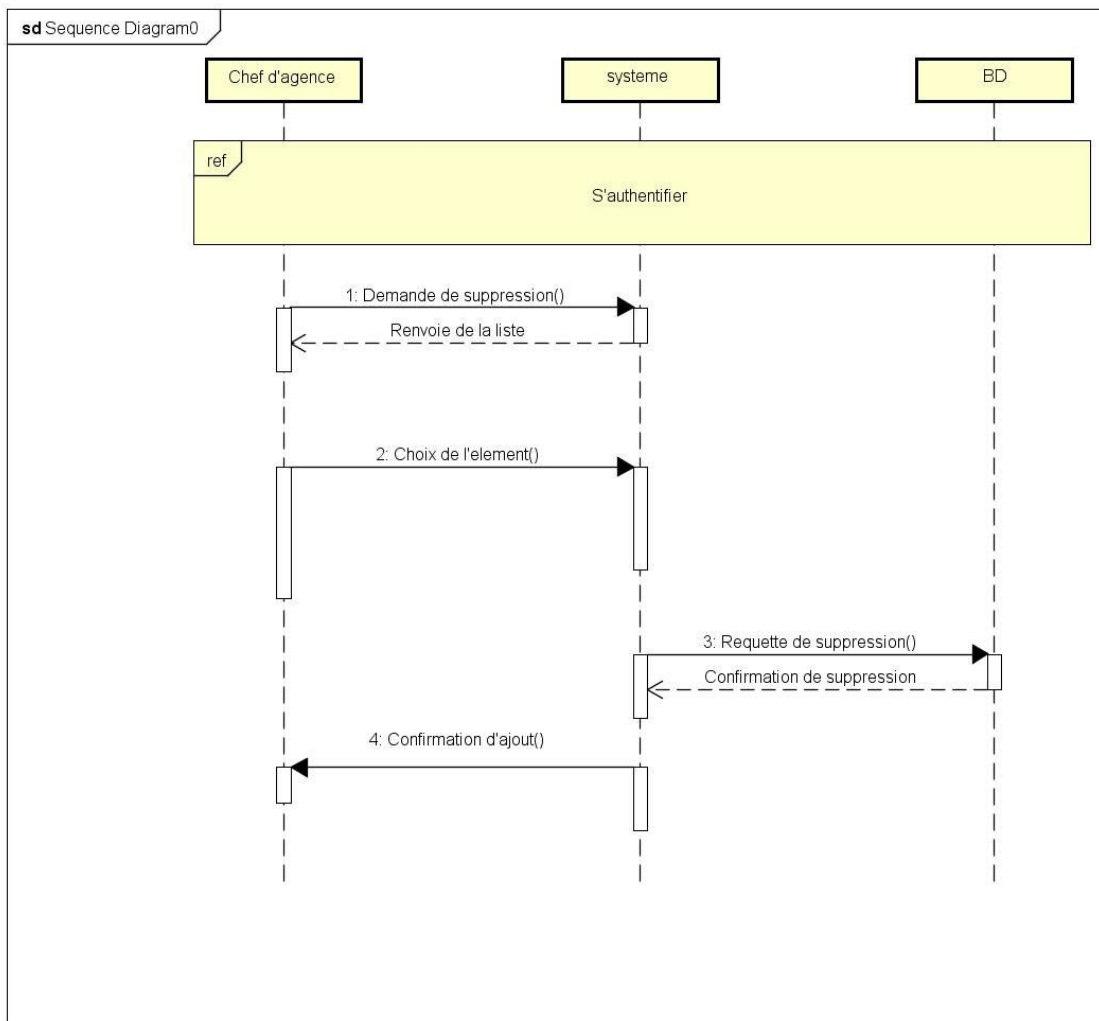
- **Message asynchrone** : Est une communication où l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations



c) Diagramme de sequence : « Ajout »



d) Diagramme de sequence : « Suppression »



4. Diagramme de composant et de déploiement

A. Diagramme des composants

a) Présentation

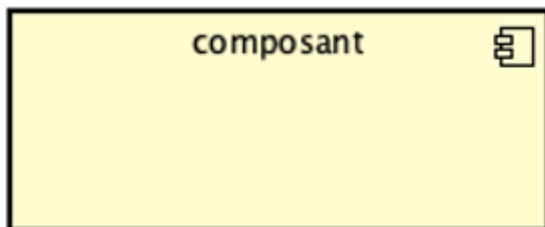
Le diagramme des composants est principalement employé pour décrire les dépendances entre les divers composants logiciels tels que la dépendance entre les fichiers exécutables et les fichiers source.

Représentation : Ils décrivent les composants et leurs dépendances dans l'environnement de réalisation.

a) Formalisme

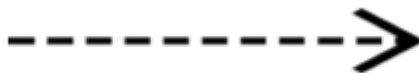
i. Composants

Un composant représente une entité logicielle d'un système. Un composant est représenté par une boîte rectangulaire, avec deux rectangles dépassant du côté gauche.



ii. Dépendance

Une dépendance est utilisée pour modéliser la relation entre deux composants. La notation utilisée pour cette relation de dépendance est une flèche pointillée, se dirigeant d'un composant donné au composant dont il dépend



B. Diagramme de déploiement

a) Présentation

Le diagramme de déploiement modélise les composants matériels utilisés pour implémenter un système et l'association entre ces composants. Des diagrammes de déploiement peuvent être mise en œuvre dès la phase de conception pour documenter l'architecture physique du système.

Représentation : Les éléments utilisés dans des diagrammes de déploiement sont des composants, comme dans les diagrammes des composants, et des nœuds, qui représentent les ressources physiques de traitement du système, et leurs association

b) Formalisme

iii. Nœud

Un nœud représente un ensemble d'éléments matériels du système. Cette entité est représentée par un cube tridimensionnel.

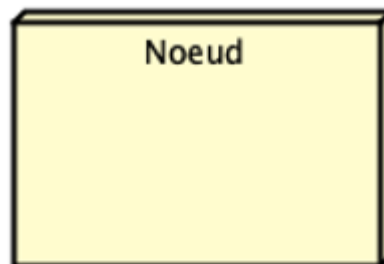


Diagramme de deploiement de notre système

pkg

