

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique
Université A.Mira Bejaia
Faculté Des Sciences Exactes
Département D'informatique



Niveau : Master 1
Option : Génie logiciel
Module : Méthode de conception

Rapport sous le thème

Modélisation d'une application informatique avec UML à l'aide d'un AGL

Réalisé par :
MAOUCHI Mohamed djamil
ZADIR Azeddine

Dirigé par :
A.Achroufene
A.Ait Abdelouhab

20 mai 2019

Table des matières

Introduction	1
1 Spécification des besoins	2
1.1 Identification des acteurs	2
1.2 Modélisation du contexte	2
1.3 Identifications des cas d'utilisation	3
1.4 Diagramme de cas d'utilisation	3
2 Analyse des besoins	5
2.1 Description des cas d'utilisation	5
Conclusion	7

Introduction

Objet du rapport

L'objet de ce document est la modélisation d'une application d'informatisation d'un entrepôt de stockage avec UML et l'aide d'un AGL. Nous allons tout d'abord spécifier les besoins en détailles cette phase a pour but de décrire précisément :

- L'ensemble des fonctionnalités de l'application.
- Les objets manipulés, leurs buts et leurs principes de fonctionnement.

Dans la 2ème partie du rapport nous allons nous focaliser sur l'analyse des besoins, cette phase nous permettra de bien comprendre le contexte et de déterminer les besoins et les contraintes.

La 3ème partie consiste en "La conception" du system. Nous établissons les diagrammes de communication qui sont utilisés pour décrire entre les objets de notre system, enfin le diagramme de classes de conception.

La 4ème partie concerne la génération du code java à partir du diagramme de classe de conception. Nous présentons l'AGL qui nous permettra de générer du code de façon automatique.

Domaine du system

Ce rapport est applicable pendant la phase de développement du system « Informatisation d'un entrepôt de stockage. La modélisation de ce system sera conforme aux éléments présents dans ce rapport.

1 Spécification des besoins

1.1 Identification des acteurs

On entend par acteur, un humain, une machine ou un système qui ne fait pas partie de la solution à réaliser mais qui participe à son fonctionnement général par une interaction. Dans notre cas, nous aurons typiquement des acteurs humains qui sont des employés et du superviseur (Voir tableau 2.1).

Acteur	Rôle
Employé	L'employé aura accès aux fonctionnalités du logiciel après authentification
Superviseur	En plus d'avoir aura accès aux fonctionnalités du logiciel il pourra gérer la bonne application des consignes
Administrateur	Gérer les utilisateurs
Système de gestion	Recevoir des informations à l'issue de la réalisation du cas d'utilisation des autre acteurs

TABLE 1 – Les acteurs du système

1.2 Modélisation du contexte

La figure 2.1 représente les interactions entre le système et les acteurs qui y sont impliqués :

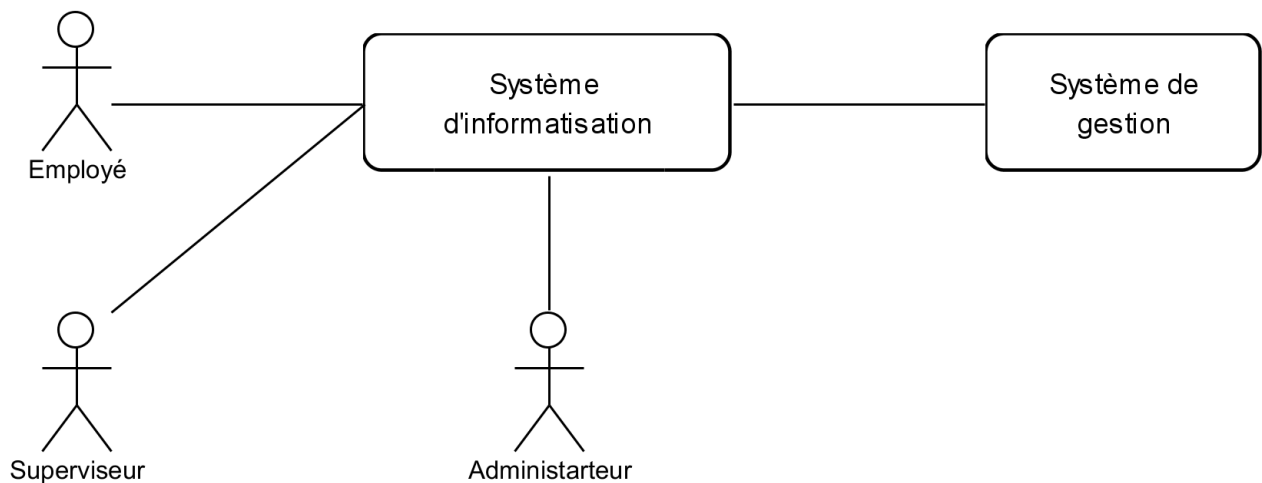


FIGURE 1 – Diagramme de contexte du système à réaliser.

1.3 Identifications des cas d'utilisation

Le Cas d'utilisation est une description des interactions qui permettront à l'acteur d'atteindre son objectif en utilisant le système [UML]. Le tableau 2.2 résume les cas d'utilisations du système à réaliser :

N	Cas d'utilisation	Acteur
1	S'authentifier	Employé/Superviseur
2	Saisie des caractéristiques d'articles lors du chargement	
3	Saisie des caractéristiques d'articles lors du déchargement	
4	Gérer les employés	Superviseur

TABLE 2 – Cas d'utilisation du système à réaliser

1.4 Diagramme de cas d'utilisation

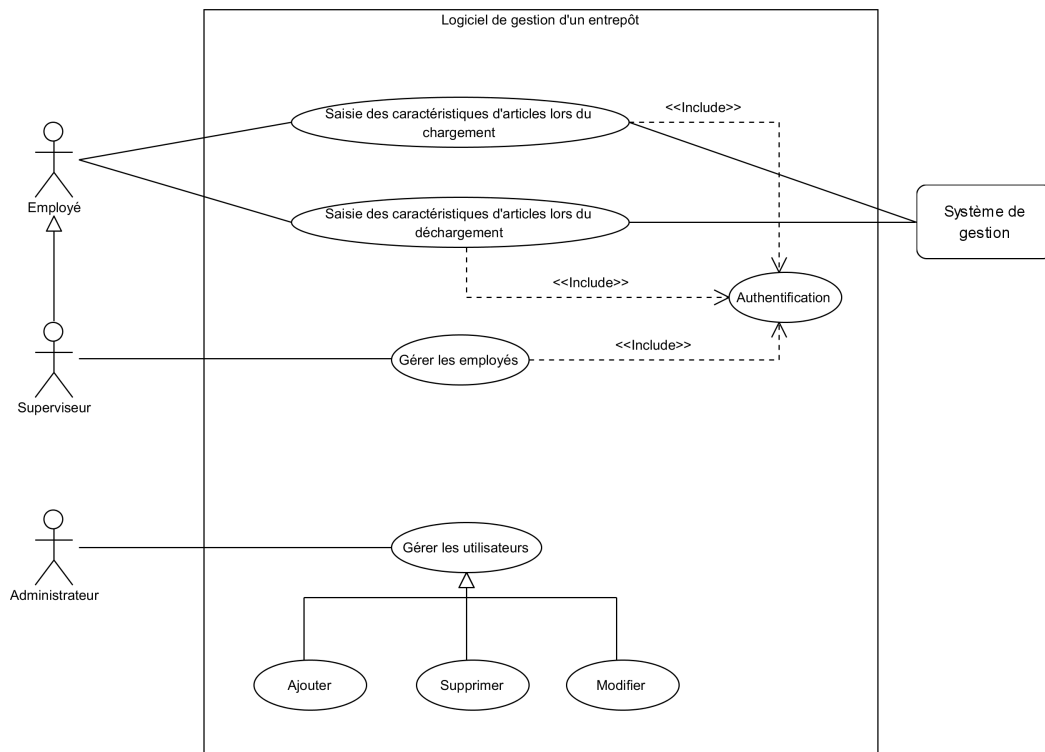


FIGURE 2 – Diagramme de contexte du système à réaliser.

2 Analyse des besoins

2.1 Description des cas d'utilisation

Les diagrammes réalisés jusqu'à maintenant (diagramme de contexte, diagramme de cas d'utilisation) nous ont permis de découvrir petit à petit les fonctionnalités que l'on devrait avoir dans le futur logiciel.

Nous allons désormais parler de l'interaction entre les acteurs et le système : il s'agit de décrire la chronologie des actions qui devront être réalisées par les acteurs et par le système lui-même. Cette description va nous permettre :

- clarifier le déroulement de la fonctionnalité.
- décrire la chronologie des actions qui devront être réalisées.
- d'identifier les parties redondantes pour en déduire des cas d'utilisation plus précises qui seront utilisées par inclusion, extension ou généralisation/spécialisation. Et oui, dans ce cas nous réaliserons des itérations sur les diagrammes de cas d'utilisation.
- d'indiquer d'éventuelles contraintes déjà connues et dont les développeurs vont devoir tenir compte lors de la réalisation du logiciel. Ces contraintes peuvent être de nature diverse.

Cas d'utilisation «S'authentifier» :

Cas d'utilisation : <S'authentifier>

Acteurs : <Employé,Superviseur>

Objectif : -Il permet à l'acteur de s'authentifier.

[Pré-condition :] -Les identifiants de l'acteur doivent exister dans la base de donnée.

[Post-condition :] -Acteur Identifié

Scénario nominal : 1. L'acteur ouvre le logiciel.

2. Le système affiche la fenêtre d'authentification

3. L'acteur saisit les identifiants

4. Le système vérifie l'existence des données

5. Le système identifie l'acteur.

Scénario alternatif : *A. Erreur d'authentification* : identifiants non valides.

Cet enchaînement démarre au point 4.

5. Le système affiche un message d'erreur.

Le scénario reprend au point 2.

[Contraintes non fonctionnelles :] <Confidentialité>

Cas d'utilisation «Saisie des caractéristiques d'articles lors du déchargement» :

Cas d'utilisation : <Saisie des caractéristiques d'articles lors du déchargement>

Acteurs : <Employé,Superviseur>

Objectif : - Il permet à l'acteur de générer une liste où figure un emplacement pour chaque article saisi.

[Pré-condition :] - L'article doit exister dans la base de donnée.

[Post-condition :] -Génère une liste où figure un emplacement pour chaque article saisi.

Scénario nominal : 1. L'acteur saisit les caractéristique de l'article.

2. Le système vérifie l'existence de l'article.

3. Le système recherche un emplacement dans un stock.

4. Le système affiche la liste où figure un emplacement pour chaque article.

Scénario alternatif : *A. Erreur article non-existant* : caractéristique invalide.

Cet enchaînement démarre au point 2.

5. Le système affiche un message d'erreur.

Le scénario reprend au point 1.

[Contraintes non fonctionnelles :] <Temps de réponse>

Conclusion