

AP-5		Projet
J2EE		17 décembre 2013

# Projet BugsReport

## Application de remontée d'incidents

---

- Le projet sera rendu sous forme d'archive web compressée (war), la semaine de la rentrée (06 janvier).

### A Objectifs

- **BugsReport** est un projet d'application web permettant de gérer et de suivre les remontées de bugs, constatés par des utilisateurs sur des applications développées par une SSII. Vous devez prendre en charge ce développement, en constituant des équipes de 4 personnes.

### B Contraintes techniques

- L'application sera développée en Java (Java EE), avec le framework KObject.
- Elle respectera au mieux le design pattern MVC2, en séparant modèle (classes Métier), vues (interfaces web de saisie et d'affichage) et contrôleur (vérifiant l'intégrité des données et mettant en place la logique applicative).
- Elle utilisera la base de données Mysql fournie en annexe. Cette base pourra évoluer en fonction des besoins du développement.

### C Règles de gestion du SI mis en place :

#### 1 Applications et useCases

Une application est un outil développé disposant de fonctionnalités (usecases), fonctionnalités sur lesquelles les utilisateurs pourront remonter les bugs (report). Une application dispose :

- d'un libellé
- d'une description
- d'un numéro de version
- d'une date de création (mise à disposition du public)
- de fonctionnalités (use cases ou cas d'utilisation)

Chaque cas d'utilisation a un code et un libellé.

#### 2 Reports

Un report est déposé par un utilisateur membre.

Il est associé au cas d'utilisation d'une application.

Il dispose :

- d'un libellé (court)
- d'une description
- d'une date de création (date auto)

Il permet de spécifier l'action que l'utilisateur a effectuée, et le message système qu'il a reçu.

Le report dispose d'un statut, correspondant à son cycle de vie (Nouveau, Affecté, Réponse, Résolu...).

Le report a une réponse officielle, donnée par la SSII (info sur bug existant, ou possible résolution).

AP-5		Projet
J2EE		17 décembre 2013

### **3 Affectation des reports**

Les reports d'une application sont affectés à des développeurs, à une date donnée, par le(s) chef(s) de projet.

### **4 Echanges sur report**

Des messages sont échangés relativement à un report, par les utilisateurs. Les messages sont décrits par un objet, un contenu et une date.

### **5 Gestion des utilisateurs**

Chaque utilisateur appartient à un groupe, et est décrit par les infos suivantes :

- login
- password
- email

### **6 Gestion des droits dans l'application**

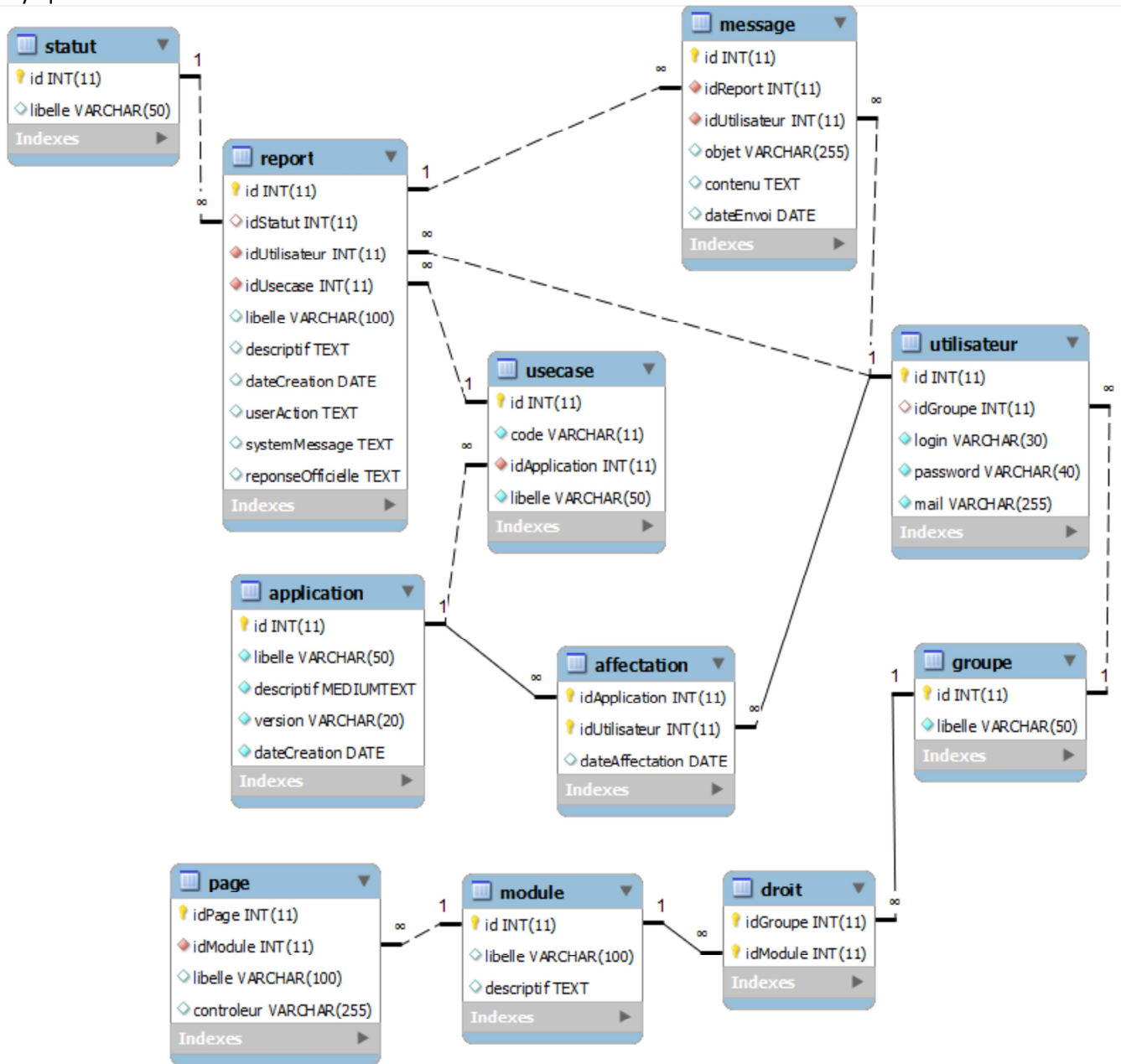
L'application est constituée de modules, chaque module correspondant à un ensemble de pages (URL en \*.do).

Les modules sont accessibles pour des groupes d'utilisateurs.

On obtient le modèle physique suivant :

# Base bugReport

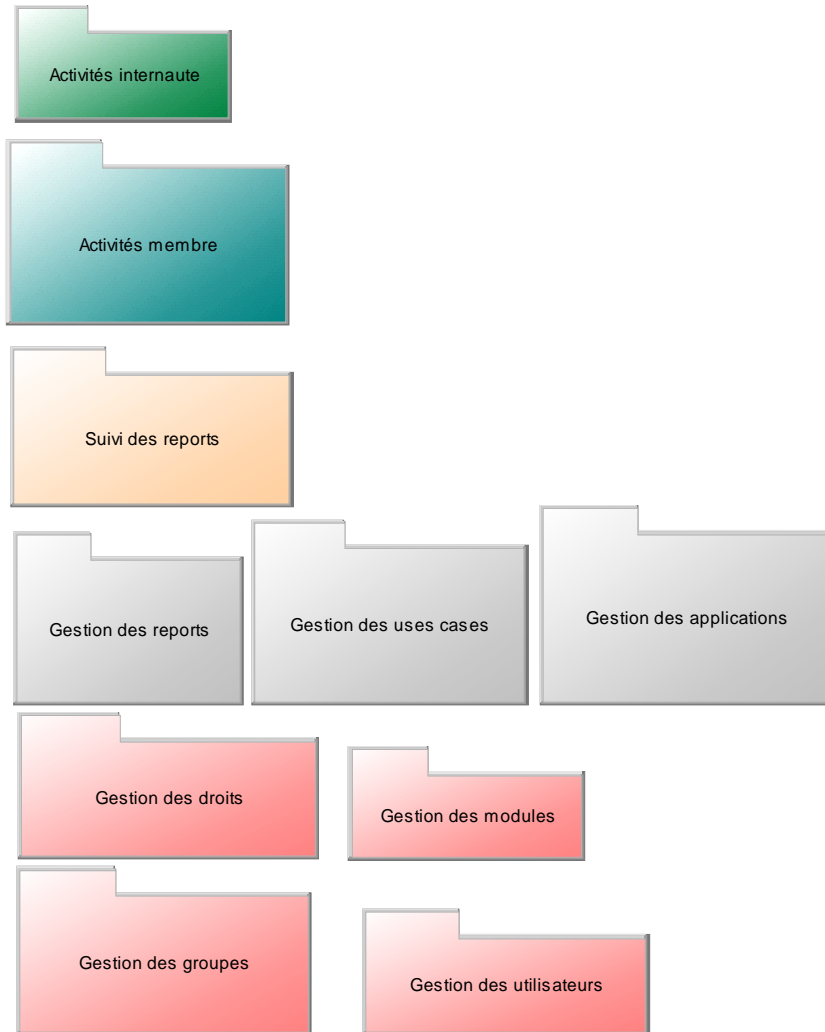
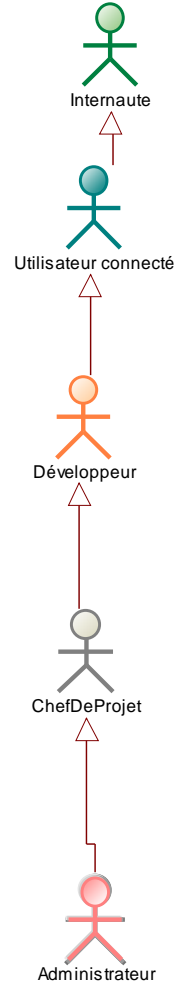
Mysql schema



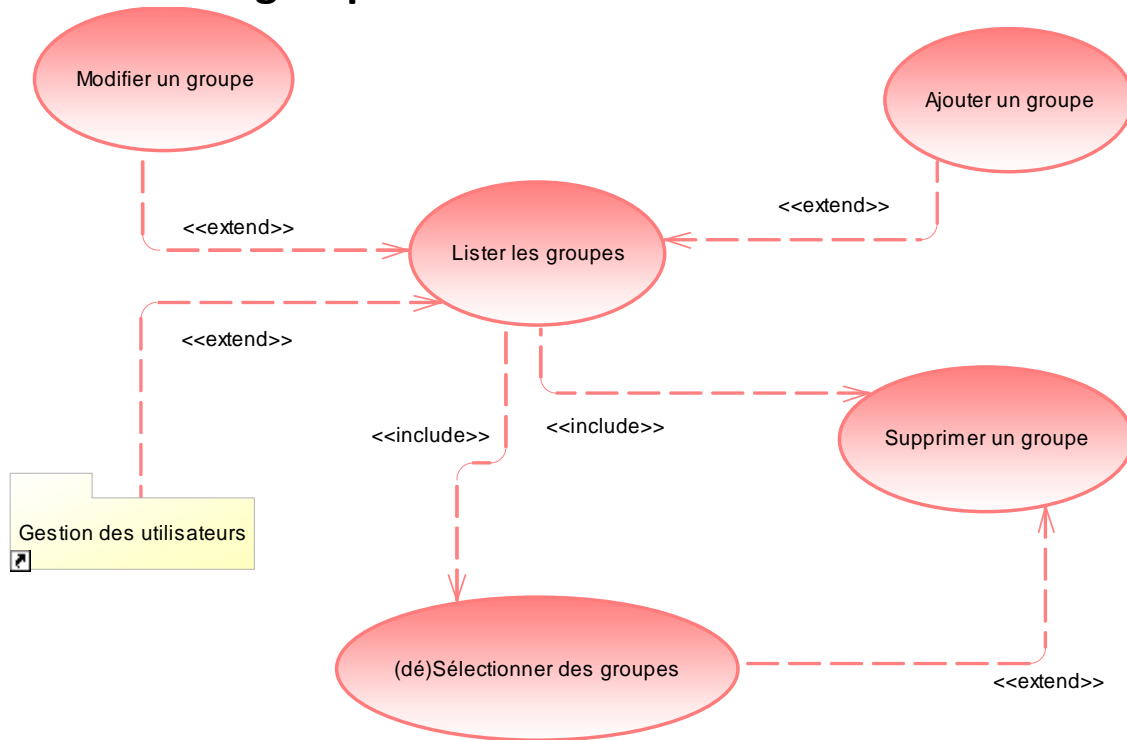
## D Fonctionnalités à implémenter :

# BugsReport

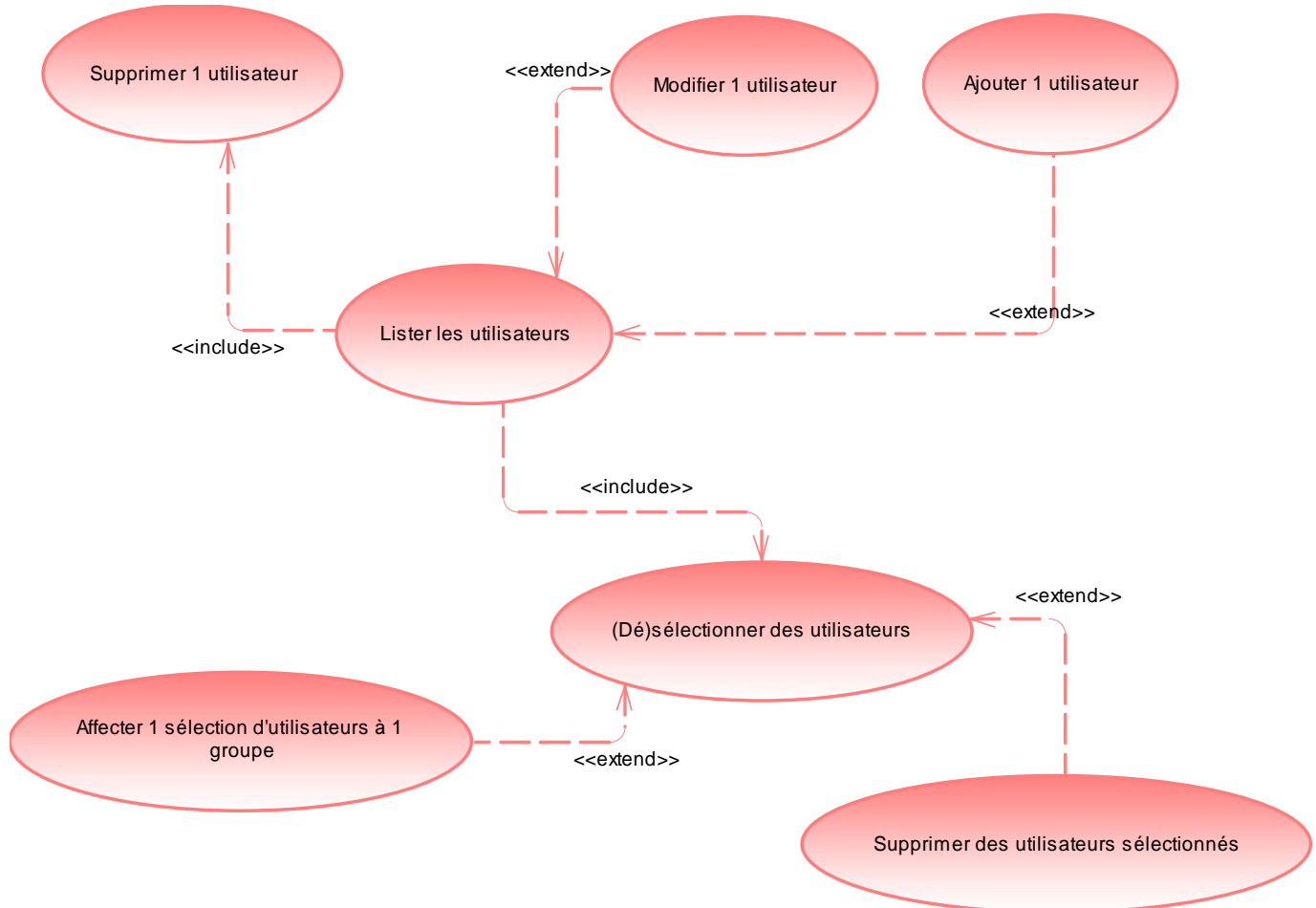
Acteurs et modules



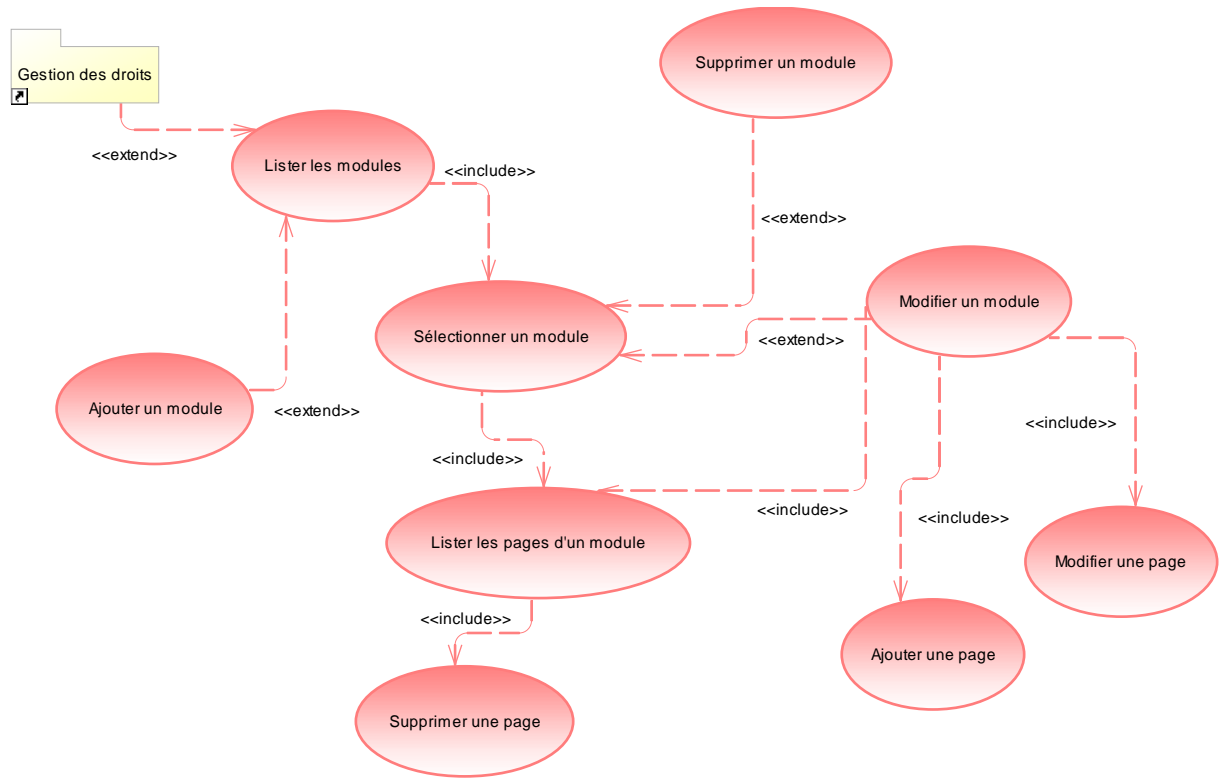
## Gestion des groupes



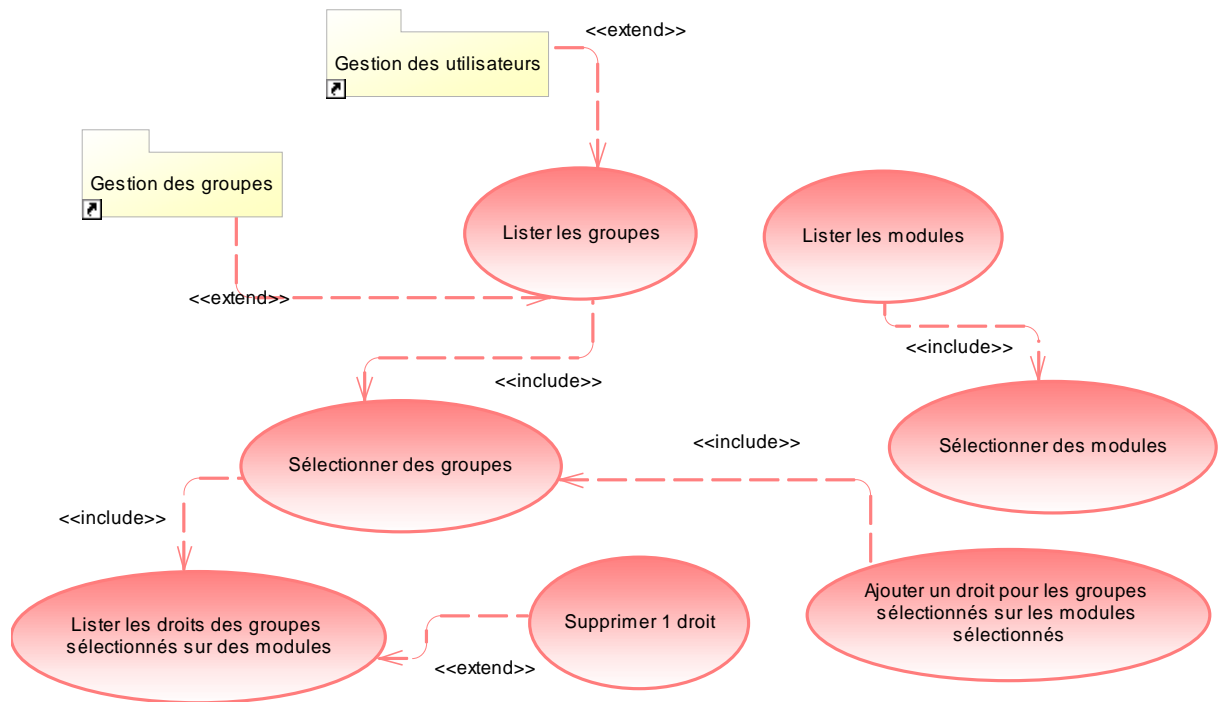
## Gestion des utilisateurs



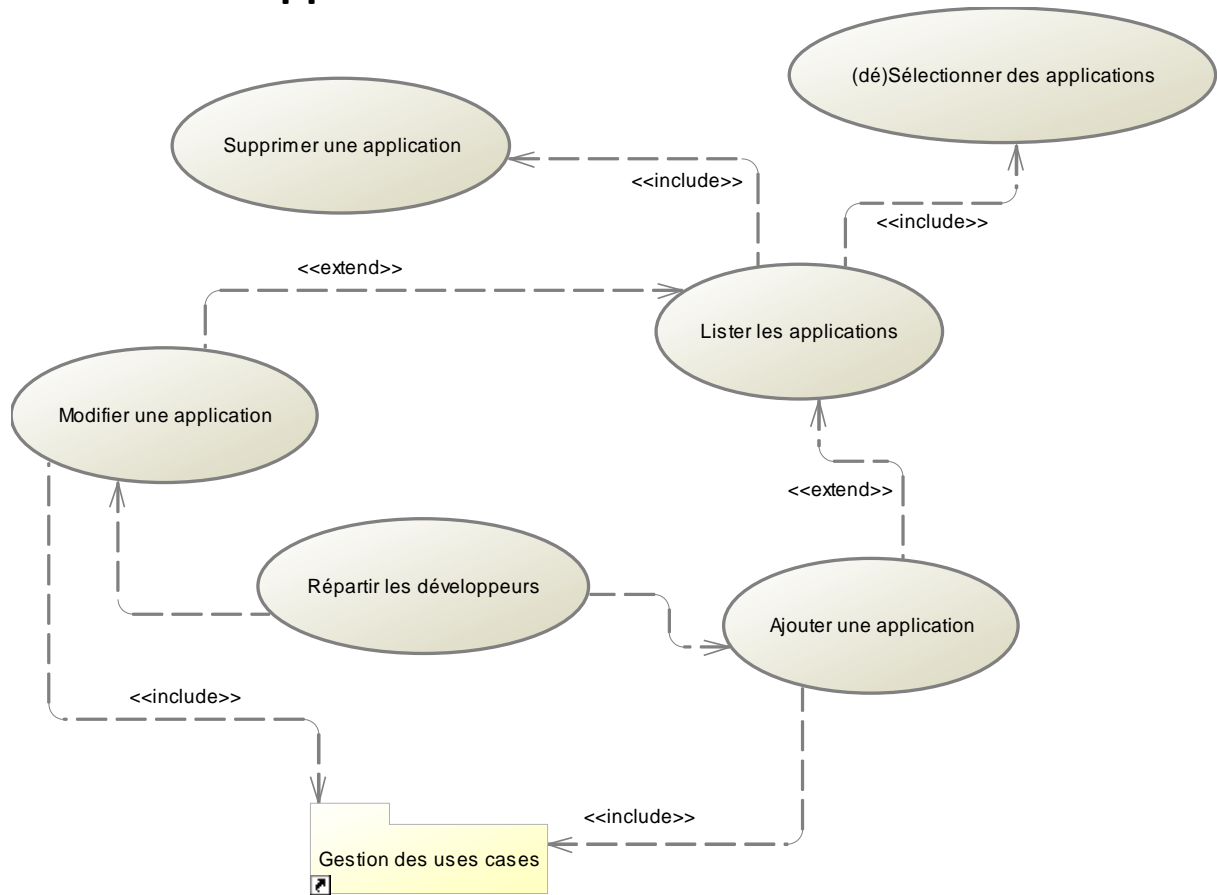
# Gestion des modules



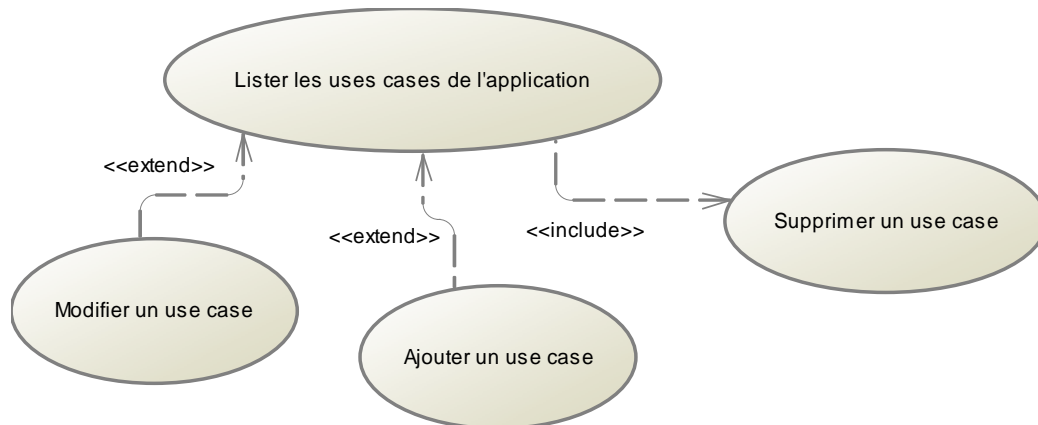
# Gestion des droits



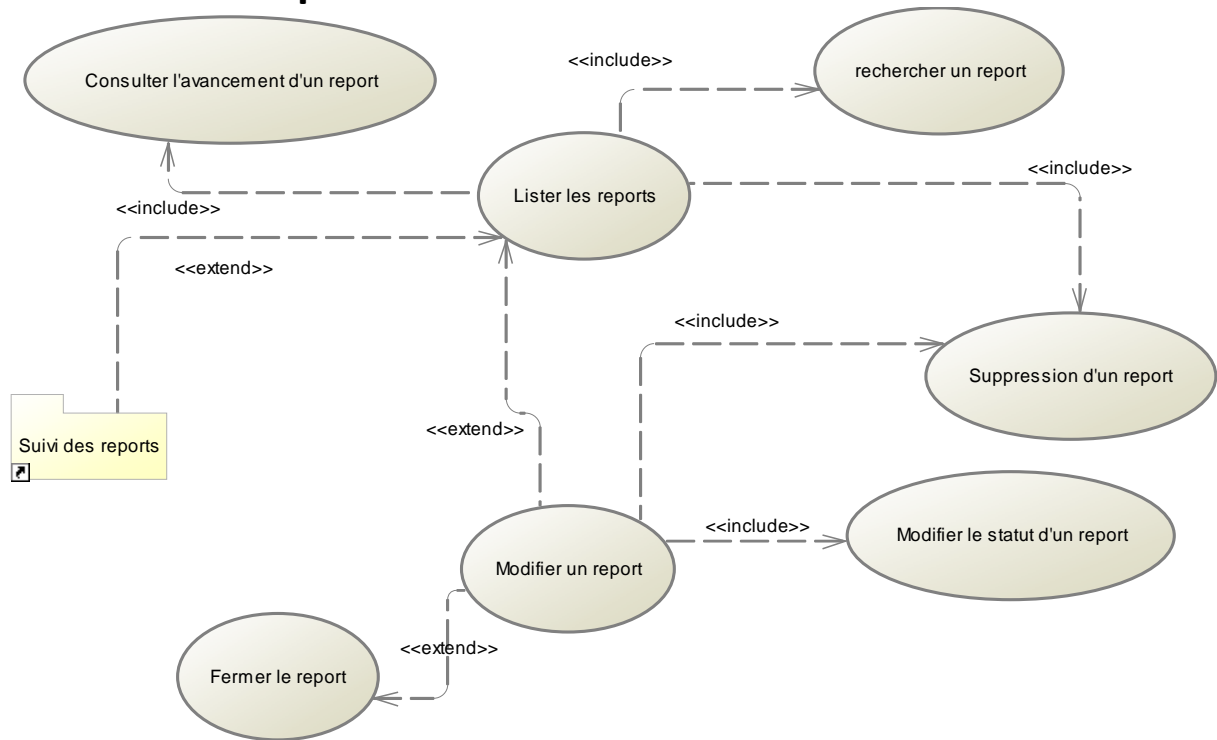
## Gestion des applications



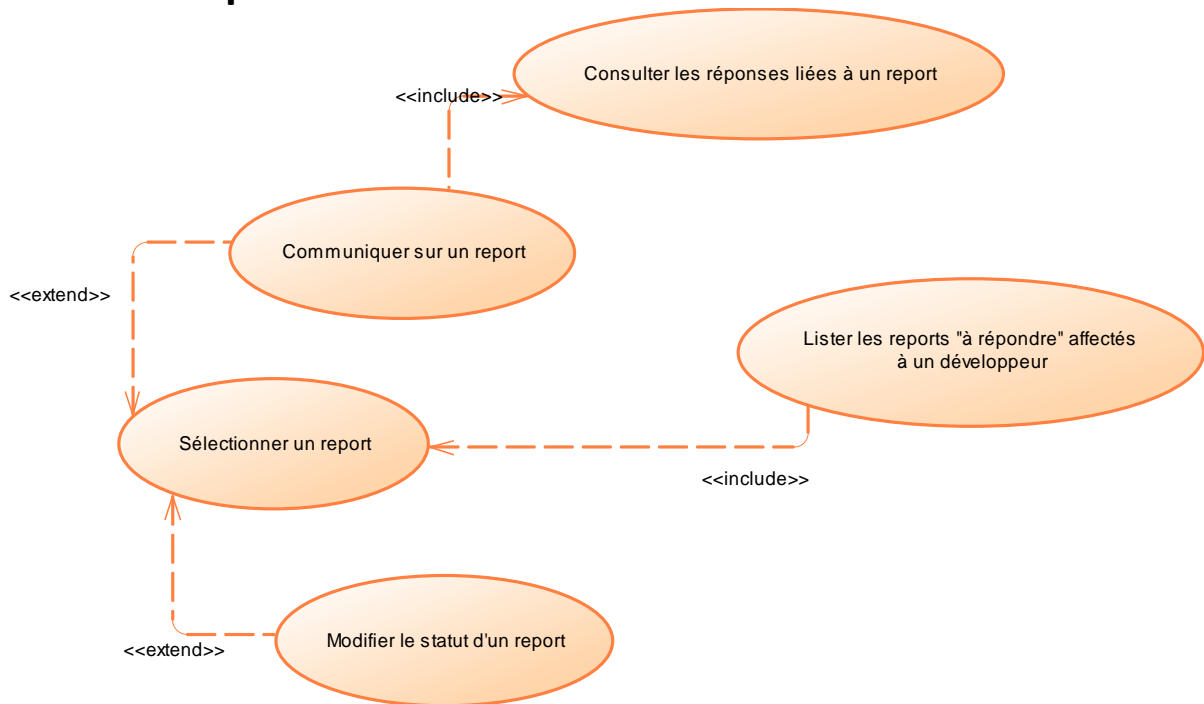
## Gestion des uses cases



## Gestion des reports

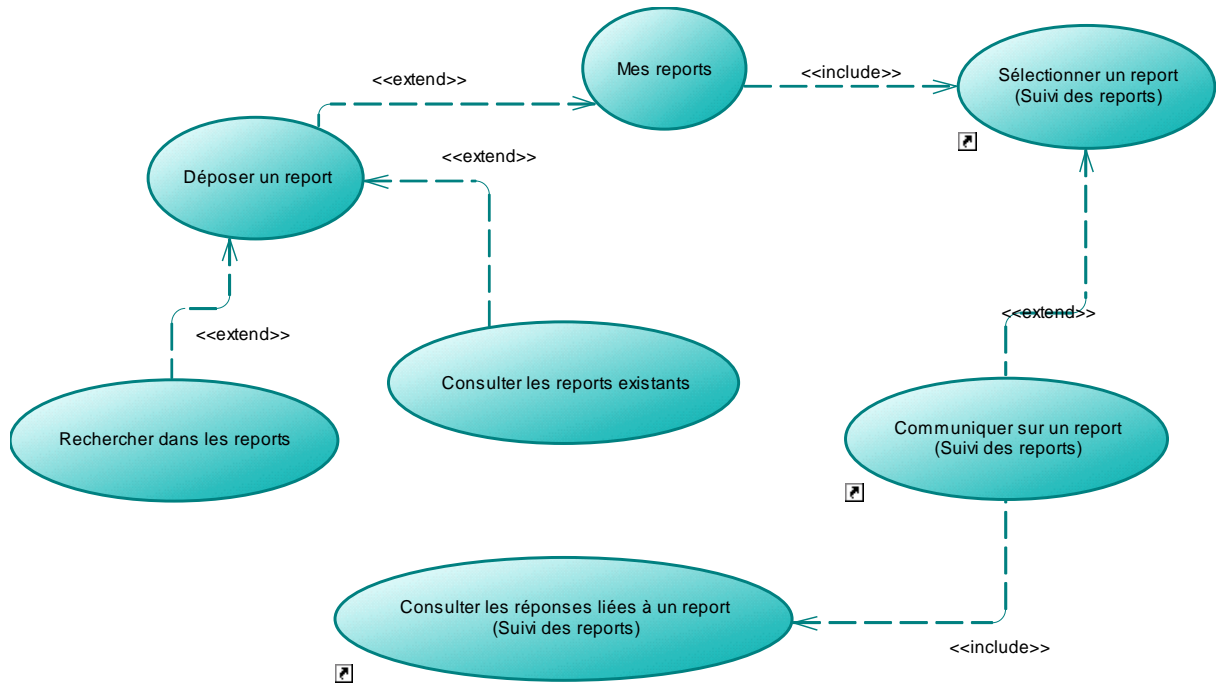


## Suivi des reports

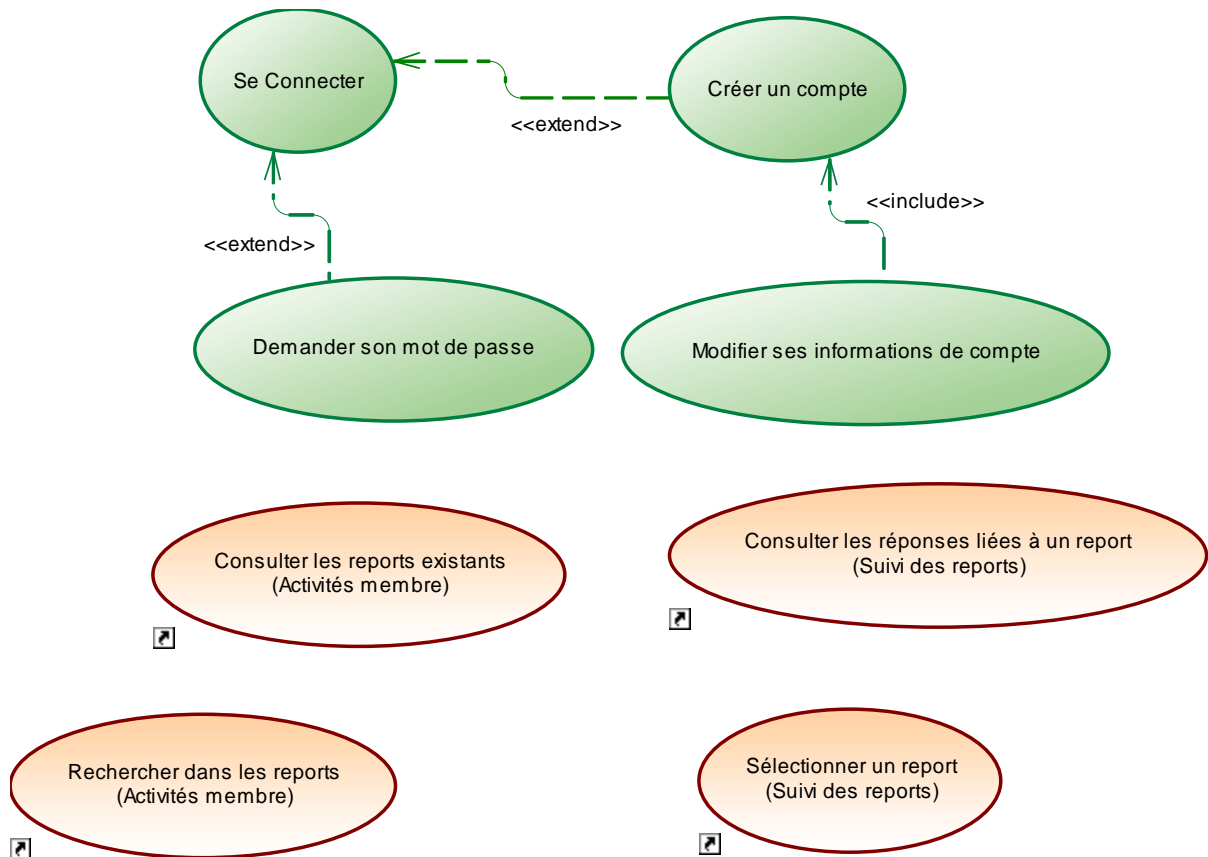




## Activités membre



## Activités internaute



AP-5		Projet
J2EE		17 décembre 2013

## E Travail à rendre

Le projet est à réaliser par groupe de 4 (**obligatoire**), il sera rendu sous la forme d'une archive WAR nommée **CodeDuGroupe.war**, où CodeDuGroupe est le code entré sur l'interface de constitution des équipes (<http://bugsreport.kobject.net>).

L'archive devra comprendre :

1. L'application web
2. une documentation technique composée :
  - a. de la javadoc complète des classes, objets, variables, méthodes et fichiers créés
  - b. d'un plan détaillé de l'application précisant les interactions :
    - Mappings d'url, contrôleurs
    - Logique applicative détaillée (ajax-includes) + Displays
    - Formulaires, contrôles, validation

Ces documents techniques doivent permettre une éventuelle reprise (pour ajout de fonctionnalités par exemple) par une autre équipe de développement.

Ils seront accessibles par un lien sur la page d'accueil, dans une rubrique « A propos de... »

## F Critères de notation

### ▪ La documentation :

- Qualité informative
- Qualité rédactionnelle et ergonomie.

### ▪ Le code :

Doit être facilement maintenable et évolutif :

- Nommage correct (classes, variables, fichiers, membres...) respectant des normes
- respect du principe de délégation
- respect des principes de la POO
- Séparation des couches (MVC2)
- Lisibilité et documentation du code

### ▪ Les interfaces :

On veillera particulièrement à leur ergonomie :

- à la simplicité de prise en main (sans documentation),
- au respect des normes de présentation des interfaces web,
- à l'accessibilité des fonctionnalités,
- à la normalisation W3C (+ intégration de HTML 5).

### ▪ L'application :

- Devra implémenter de façon complète et accessible les fonctionnalités en respectant les contraintes données.

### ▪ Le travail en équipe :

- Chaque membre de l'équipe, quel que soit son rôle, doit veiller au bon avancement du projet.

### ▪ Le respect des consignes données.