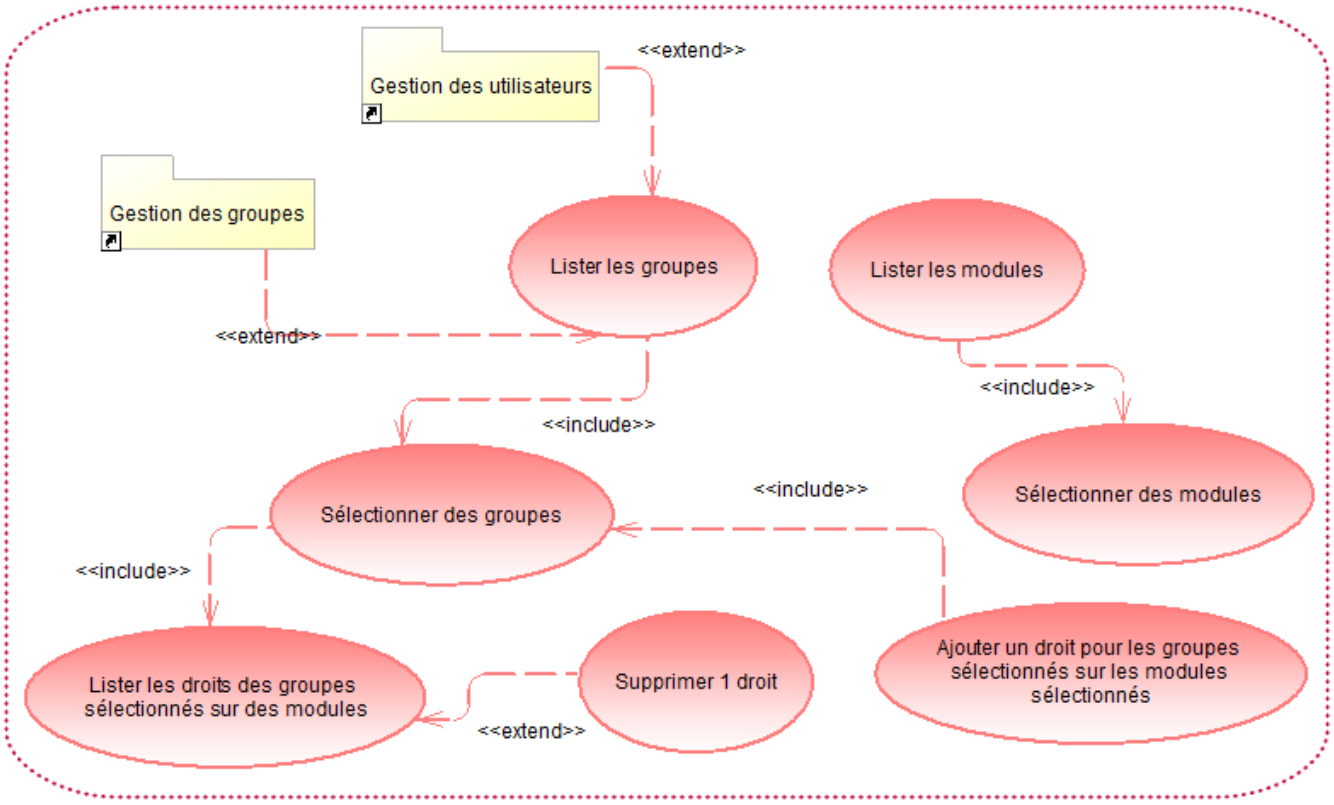
Documents d'analyse complémentaires

-- Descriptifs textuels des cas d'utilisation importants

-- Module gestion des droits

-- Diagramme partiel des uses cases

La **gestion des droits** est un cas d'utilisation dans lequel on trouvera cet ensemble de fonctionnalités :



-- Descriptif textuel

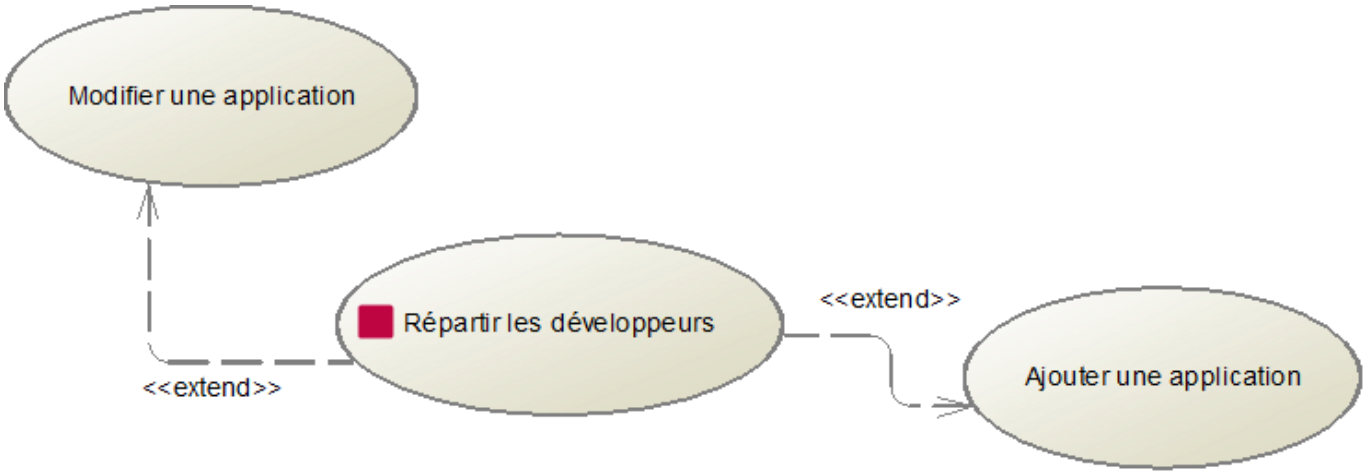
Titre	Descriptif textuel : Cas **Gestion des droits**
Contexte	Appli web **bugsReport**
Auteur	jc
Date	20 déc. 2013
Version	1.0.0.1

Cas d'utilisation	**Gestion des droits**
Acteur principal	Administrateur
Préconditions	• Existence de groupes et de modules, • l'utilisateur est authentifié et dispose des droits nécessaires à la réalisation du cas
Post-conditions	• Les droits sont modifiés
Objectifs	• l'administrateur doit pouvoir modifier les droits des groupes sur les modules de l'application

Scénario nominal	1. l'administrateur choisit le module de gestion des droits. 2. le système affiche l'interface de gestion des droits intégrant : ◦ la liste des groupes ◦ la liste des modules 3. l'administrateur sélectionne un ou plusieurs groupes. 4. le système affiche la liste des droits des groupes sélectionnés. 5. l'administrateur sélectionne un ou plusieurs modules. 6. l'administrateur ajoute des droits sur les modules sélectionnés aux groupes sélectionnés. 7. le système ajoute ces droits dans l'interface. 8. l'administrateur valide ses modifications. 9. le système met à jour la base de données.
Scenarii alternatifs	• En 6 : ◦ l'administrateur ajoute des droits déjà existants => 7. le système n'ajoute pas les droits déjà existants • A toutes les étapes : ◦ l'administrateur peut supprimer des droits existant dans la liste.

-- Module Gestion des applications

-- Diagramme partiel



-- Descriptif textuel

Titre	Descriptif textuel : Cas **Répartir les développeurs**
Contexte	Appli web **bugsReport**
Auteur	jc
Date	20 déc. 2013
Version	1.0.0.1

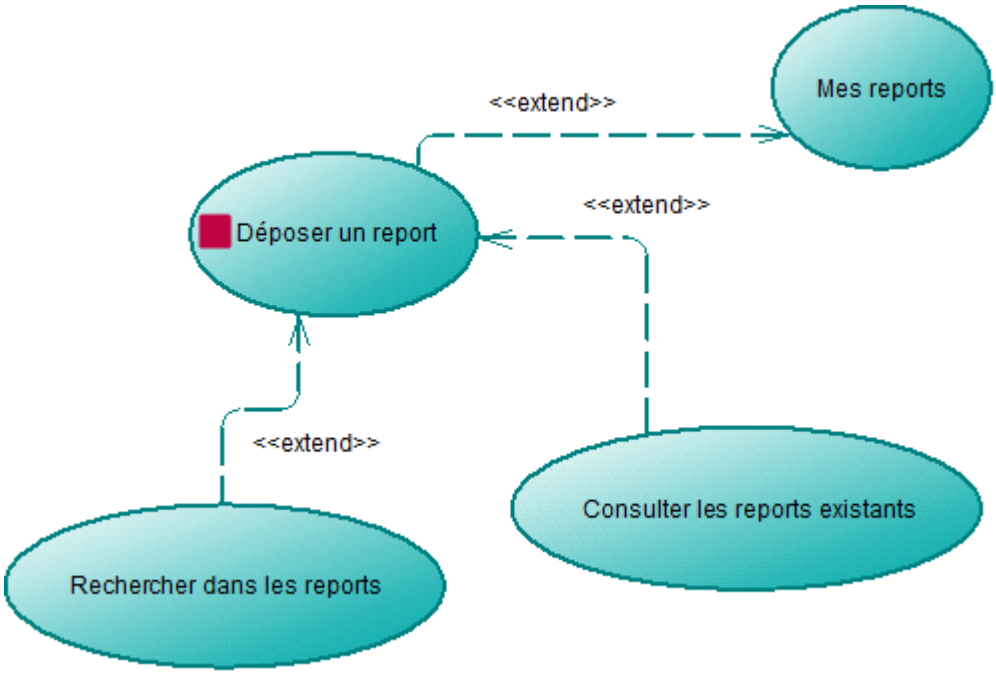
Cas d'utilisation	**Répartir les développeurs**
Acteur principal	Chef de projet
Préconditions	• L'utilisateur actuel est chef de projet, • il existe au moins une application • Il existe des développeurs
Objectifs	• Répartir ou affecter des développeurs au suivi des reports liés à une application
Scénario nominal	1. A partir de l'édition d'une application (ajout ou modification), le chef de projet fait le Choix de l'option **répartition des développeurs**. 2. Le système affiche la liste des développeurs disponibles, et la liste des développeurs déjà affectés à l'application. 3. Le chef de projet sélectionne les Développeurs à ajouter. 4. Il valide son choix d'affectation. 5. Le système affiche les développeurs sélectionnés dans la liste des développeurs de l'application, et retire ces mêmes développeurs de la liste de ceux qui sont disponibles. 6. Le chef de projet valide ses modifications. 7. Le système met à jour l'application dans la base de données.

Scenarii alternatifs	• En 3 : ◦ Le chef de projet peut retirer des développeurs de la liste des dév affectés à l'application, et valider son choix => Retour en 7 • En 4 et 6 : ◦ Le chef de projet abandonne ses modifications

-- Module Activités membre

-- Dépôt d'un report

Diagramme partiel



Descriptif textuel

Titre	Descriptif textuel : Cas **Déposer un report**
Contexte	Appli web **bugsReport**
Auteur	jc
Date	20 déc. 2013
Version	1.0.0.1

Cas d'utilisation	**Déposer un report**
Acteur principal	membre
Préconditions	• l'utilisateur membre est connecté à l'application, • il existe des applications et des cas d'utilisation.
Objectifs	• Permettre à un utilisateur membre de reporter un bug (dysfonctionnement) sur l'une des fonctionnalités (useCase) d'une application.

Scénario nominal	1. Le membre choisit l'option Report d'un nouveau bug. 2. Le système affiche un formulaire de saisie d'un report, présentant la liste des applications existantes. 3. Le membre sélectionne une application. 4. Le système affiche la liste de tous les cas d'utilisation (fonctionnalités) de l'application sélectionnée. 5. Le membre sélectionne un cas d'utilisation. 6. Le système affiche les zones de saisie des informations du report : ◦ libellé (titre ou sujet) ◦ descriptif (description générale et résumée du problème rencontré) ◦ userAction (la suite d'action réalisée par l'utilisateur et permettant de reproduire le bug) ◦ actualResults (les résultats obtenus par l'utilisateur) ◦ expectedResults (les résultats attendus par l'utilisateur en cas de fonctionnement "normal" 7. le membre procède au remplissage des champs. 8. le système affiche les reports similaires trouvés, au fur et à mesure de la saisie des informations par l'utilisateur 9. Le membre valide sa saisie. 10. le système enregistre le report dans la base de données (ajout auto de la date de création) et communique un message aux développeurs affectés à l'application concernée.
Scenarii alternatifs	• En 9 : ◦ Le membre trouve un report parmi la liste affichée des reports existants correspondant à son problème : il annule sa saisie et consulte le report proposé • En 10 : ◦ Le système avertit le membre des erreurs de remplissage (champs vides)

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/javaee/projet2013/analysecomp

Last update: **2019/08/31 14:21**