

## Documents d'analyse

---

### Application BugReport

## Contenu

---

[Diagramme de cas d'utilisation](#)

[Règles de gestion](#)

[Modèle Conceptuel de Données](#)

[Modèle Logique de Données](#)

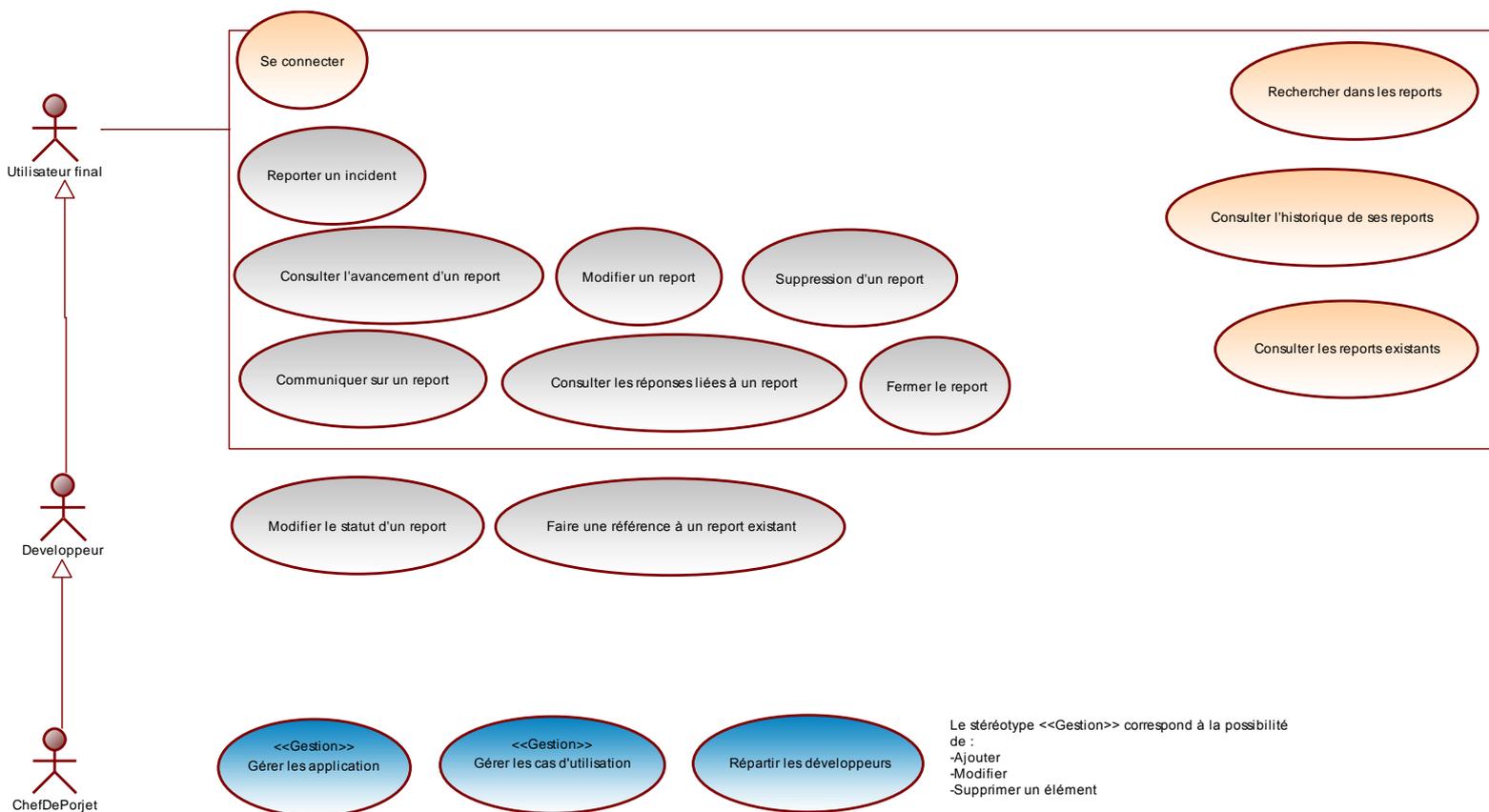
Version : 1.0.0.1

## Présentation du contexte :

Il s'agit de développer une application web de remontée d'incidents. Cette application, que l'on nommera BugReport, permettra en premier lieu de saisir des reports concernant des bugs survenus à des applications. Outre ces fonctionnalités essentielles, d'autres fonctionnalités, relatives notamment à l'administration de l'application, devront être implémentées.

## Diagramme de cas d'utilisation :

Les fonctionnalités implémentées, dans l'application BugReport, peuvent être représentées par un diagramme de cas d'utilisation.



### Précisions sur le diagramme de cas d'utilisation :

- 3 types d'acteurs sont concernés par l'application : les utilisateurs finals, les développeurs, et les chefs de projet
- les développeurs héritent des utilisateurs finals : ils ont donc tous les droits des utilisateurs finals, ainsi que d'autres supplémentaires
- les chefs de projet héritent des développeurs : ils disposent donc d'un accès à toutes les fonctionnalités implémentées dans l'application

**Version : 1.0.0.1**

- les utilisateurs peuvent se connecter à l'application, saisir un report et rechercher un report, entre autres
- les développeurs ont, en plus, la possibilité de gérer les reports
- les chefs de projet peuvent aussi gérer les applications, les cas d'utilisation, et les affectations de développeurs à des applications

---

Version : 1.0.0.1

## **Règles de gestion :**

Les règles de gestion à respecter dans l'implémentation des différentes fonctionnalités de BugReport sont les suivantes :

### **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 (usecases)

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

### **Reports**

---

1 Report est déposé par un utilisateur.

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 effectué, 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 (info sur bug existant, ou possible résolution).

### **Affectation des reports**

---

Les reports sont affectés à des développeurs, à une date donnée.

---

Version : 1.0.0.1

## Echanges sur report

---

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

## Gestion des utilisateurs

---

Chaque utilisateur appartient à un groupe.

Ils sont décrits par les infos suivantes :

- nom
- prénom
- login
- password
- email

## Gestion des droits dans l'application

---

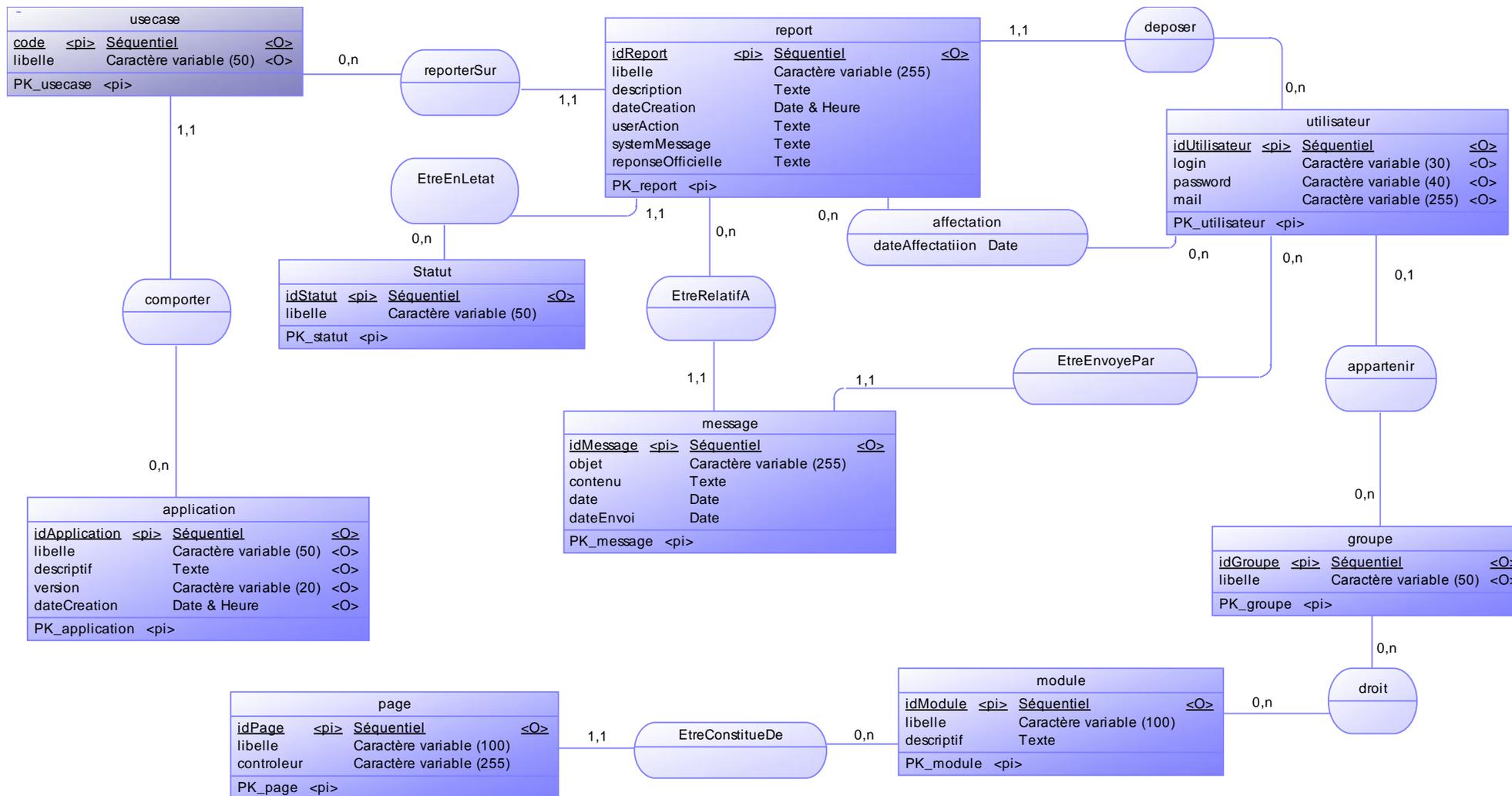
L'application sera constituée de modules, chaque module correspondant à un ensemble de pages (URL).

Les modules sont accessibles pour des groupes d'utilisateurs.

Version : 1.0.0.1

### Modèle Conceptuel de Données :

L'étude de ces différentes règles de gestion a permis d'élaborer le Modèle Conceptuel de Données (MCD) suivant :



Version : 1.0.0.1

### Modèle Logique de Données :

Le MCD a, ensuite, été traduit en Modèle Logique de Données (MLD) :

