

# BugReport : plateforme de remontée d'incidents

## Documentation technique - Sommaire :

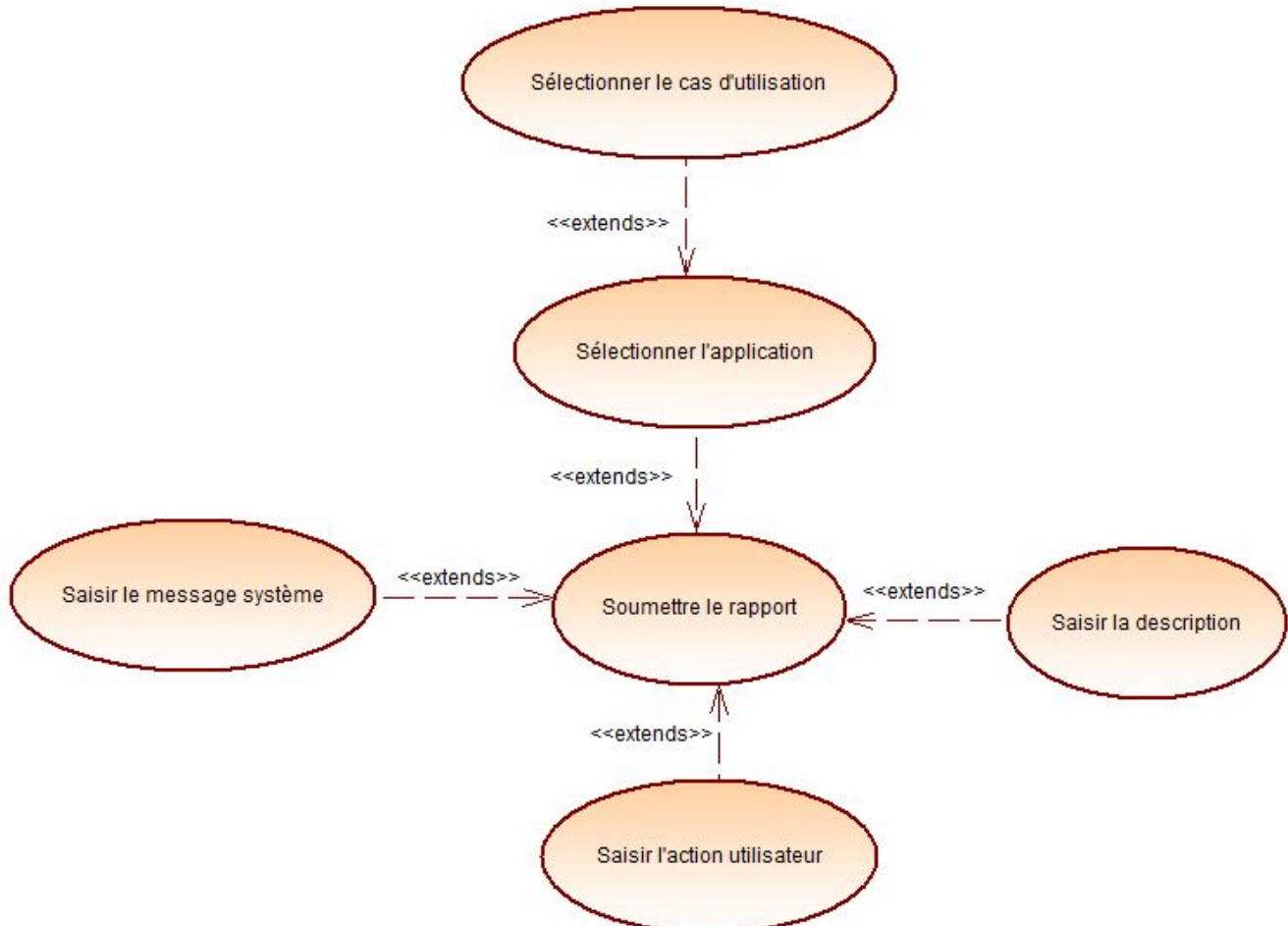
1. [Sources de l'application](#)
2. [Gestion des utilisateurs](#)
3. [Gestion des applications](#)
4. [Gestion des groupes](#)
5. Soumettre un report
6. [Gestion des Droits](#)
7. [Affectation d'utilisateurs](#)

## Documentation de support- Sommaire :

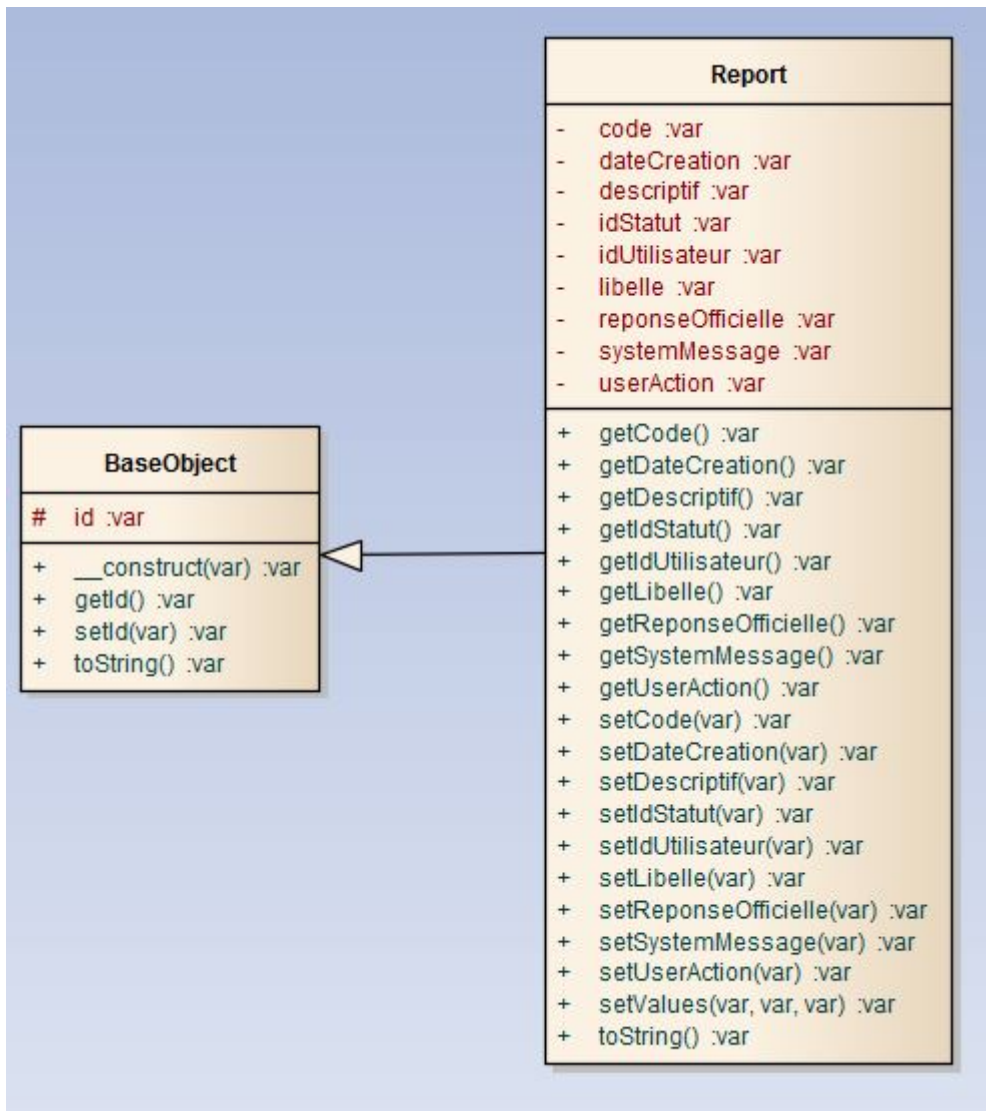
1. [Utilisateur](#)
2. [Technicien](#)
3. [Administrateur](#)

## Soumettre un rapport

### 1- Fonctionnalités



## 2- Classes métiers



## 3- Variables sessions (\$\_SESSION)

Nom	Type	Descriptif
\$_SESSION["user"]	String	login de l'utilisateur connecté

## 4- Vues

### 4.1 vReport

Soumettre un report

Application :	BugReport 2
Cas d'utilisation :	Envoyer un message relatif
Message système :	<input type="text"/>
Action utilisateur :	<input type="text"/>
Description :	<input type="text"/>

```

<form method="POST" action="" id="formReport" name="formReport">
<fieldset class="cadre">
<legend>Soumettre un report</legend>
  <table>
  <tr>
    <td>Application : </td>
    <td>
      <select id='application' name='application'>
        <?php echo $data['appli'];?>
      </select>
    </td>
  </tr>
  <tr>
    <td>Cas d'utilisation : </td>
    <td>
      <select id='selectUseCase' name='selectUseCase'>
      </select>
      <div id=autreUseCase>
      </div>
    </td>
  </tr>
  <tr>
    <td>Message système : </td>
    <td>
      <input type="text" name="systemMessage" required/>
    </td>
  </tr>
  <tr>
    <td>Action utilisateur : </td>
    <td>
      <input type="text" name="userAction" required/>
    </td>
  </tr>
  <tr>
    <td>Description : </td>
    <td>
      <textarea rows="5" cols="50" name="description"
style="resize:none;" required></textarea>
    </td>
  </tr>
</table>
      <div class="buttons">

```

```
                <div id="btSubmitReport" class="btn">Soumettre le report</div>
            </div>
        </form>
<?php echo $data['js'];?>
</fieldset>
```

## 5- Contrôleurs

### 5.1 cReport/index

Fonctionnalité	Descriptif
Soumettre un report	<b>Contrôleur de soumission de report</b>
	Appel du contrôleur <code>refresh</code>

```
public function index(){
    $this->refresh();
}
```

### 5.2 cReport/refresh

Fonctionnalité	Descriptif
Liste des applications	<b>Utilisé pour l'affichage initial et le rafraîchissement après une modification</b>
	Chargement des applications Mise en place logique client
	Passage de variables( <code>\$js</code> ) et affichage de la vue <code>vReport</code>

```
public function refresh(){
    $appli="<option selected='selected' value='null'
disabled='disabled'>Veuillez selectionner une application</option>";
    $applications=DAO::getAll("application");
    foreach($applications AS $apps){
        $appli.="<option id='".$apps->getId()."'
value='".$apps->getId().">".$apps->getLibelle()."</option>";
    }
    $js=JsUtils::postAndBindTo("#application", "click", "cReport/useCase",
'{}', "#selectUseCase",NULL,"value");
    $js.=JsUtils::postFormAndBindTo("#btSubmitReport", "click",
"cReport/submitReport","formReport","#module",true);
    echo JsUtils::doSomethingOn("#operation", "hide",400);
    echo JsUtils::doSomethingOn("#operation2", "hide",400);
    echo JsUtils::doSomethingOn("#operation3", "hide",400);
    $this->loadView("vReport",array("js"=>$js,"appli"=>$appli));
}
public function useCase($params){
    $use='';
    foreach ($params AS $idApplication)
    {
        $condition="idApplication=".$idApplication;
        $useCase=DAO::getAll("usecase", $condition)    ;
        foreach ($useCase AS $usecase){
```

```

        $use.="<option
value=' ".$usecase->getCode()." '>".$usecase->getLibelle()."</option>";
    }
}
echo $use;

}
    
```

### 5.3 cReport/useCase

Fonctionnalité	Descriptif
Liste des cas d'utilisations	<b>Utilisé pour l'affichage de la liste des cas d'utilisations dans la liste déroulante</b>
	Pour chaque cas d'utilisations : chargement des cas d'utilisations correspondant à l'application sélectionnée affichage des cas dans la liste déroulante
	Passage de variables(\$js) et affichage de la vue <u>vReport</u>

```

public function useCase($params){
    $use='';
    foreach ($params AS $idApplication)
    {
        $condition="idApplication=".$idApplication;
        $useCase=DAO::getAll("usecase", $condition)    ;
        foreach ($useCase AS $usecase){
            $use.="<option
value=' ".$usecase->getCode()." '>".$usecase->getLibelle()."</option>";
        }
    }
    echo $use;

}
    
```

### 5.4 cReport/submitReport

Fonctionnalité	Descriptif
Enregistre un report de bug	<b>Utilisé pour l'enregistrement d'un report de bug, dans la base de données</b>
	On récupère les champs du formulaire envoyés en POST
	On applique les fonctions de suppressions de doublons, majuscules...
	On crée le report
	On compare le report créé avec les reports existants Si le report n'existe pas déjà dans la base, on l'y ajoute On affiche le message qui valide ou non si l'opération s'est déroulée correctement

```

public function submitReport(){
    if($_SERVER['REQUEST_METHOD']=='POST'){
        $searchEngine=new SearchEngine();
        $idApplication=$_POST['application'];
        $idUseCase=$_POST['selectUseCase'];
        $systemMessage=$_POST['systemMessage'];
        $userAction=$_POST['userAction'];
        $description=$_POST['description'];
    }
}
    
```

```
if(isset($_POST['autreUseCase'])){
    $autreUseCase=$_POST['autreUseCase'];
}
$systemMessage=$searchEngine->cleanExpression($systemMessage);
$userAction=$searchEngine->cleanExpression($userAction);
$description=$searchEngine->cleanExpression($description);
$user=$_SESSION['user'];
foreach ($user AS $info){
    $idUtilisateur=$info->getId();
}
$report=new Report();
$report->setIdUtilisateur($idUtilisateur);
$report->setIdStatut(1);
$report->setCode($idUseCase);
$report->setSystemMessage($systemMessage);
$report->setDescriptif($description);
$report->setUserAction($userAction);
$report->setDateCreation(date("Y-m-d"));
$reportTest=DAO::getAll("report");
$retour=FALSE;
foreach ($reportTest AS $value){
    if($searchEngine->compare($report,$value)==TRUE){
        $retour=TRUE;
    }
}
if(!$retour){
    if(DAO::insert($report)){
        echo "<div class='send'>Votre report a ete envoye a une equipe
de maintenance | Nous essayons de vous repondre dans les meilleurs delais</div>";
    }else{
        echo "<div class='sendError'>Erreur lors de l'envoi du report |
Merci de bien vouloir recommencer</div>";
    }
}else{
    echo "<div class='sendCurrent'>Report identique deja existant | Le
probleme sera resolu dans les plus bref delais</div>";
}
}
}
```

From:  
<http://slamwiki2.kobject.net/> - **Broken SlamWiki 2.0**

Permanent link:  
<http://slamwiki2.kobject.net/etudiants/2014/bertrand/bugreport4?rev=1387561381>

Last update: **2019/08/31 14:32**

