



BugReport : Application de remontée d'incidents

Documentation Technique

SOMMAIRE :

Partie I : Spécification	3
1. Présentation :.....	3
2. Caractéristiques techniques :.....	3
3. Base de donnée :	4
4. Framework :	5
Partie II : Règles & Convention	5
1. General.....	5
2. Fonction :	5
Partie III : Structure de l'application	6

Partie I : Spécification

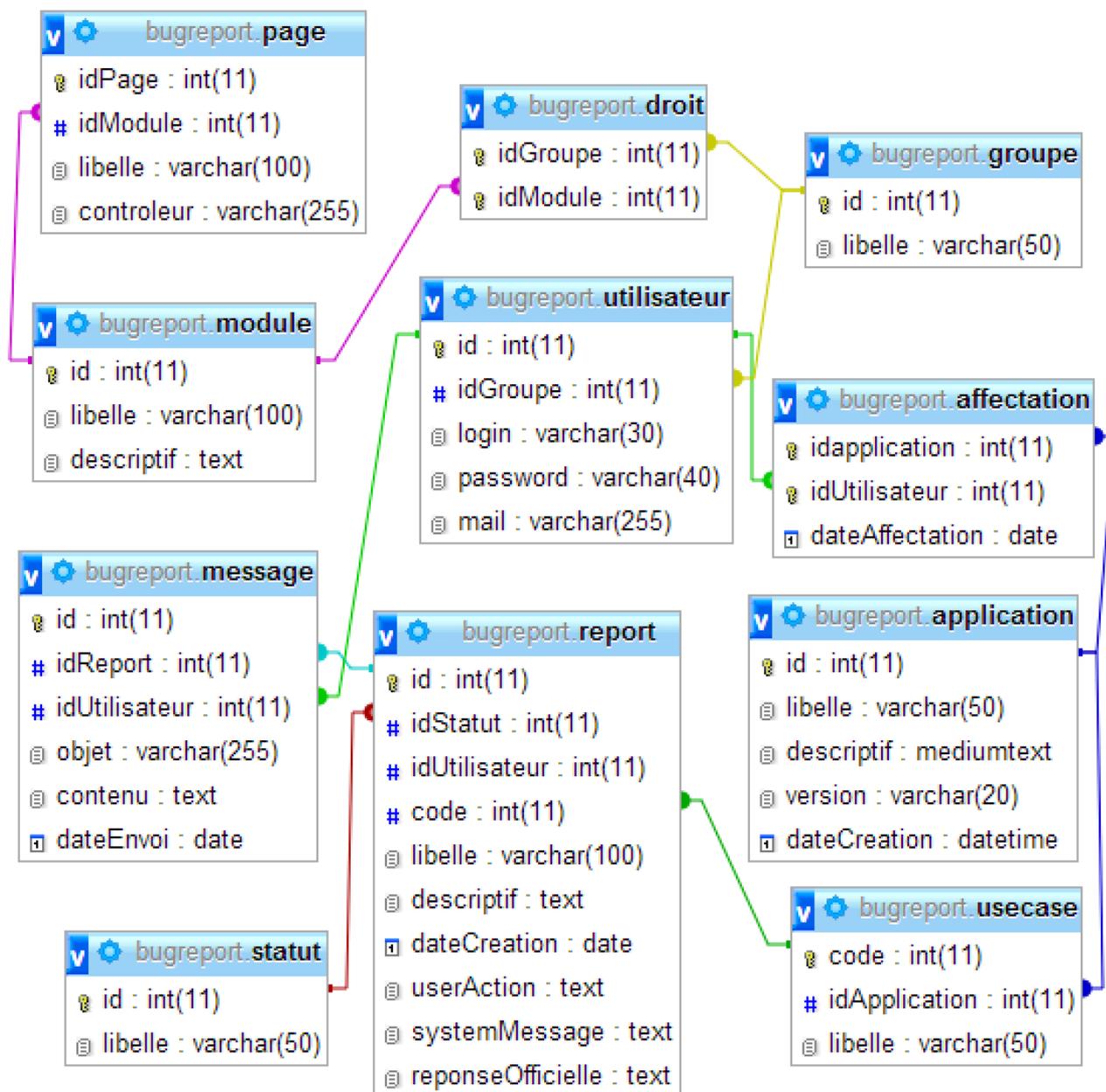
1. Présentation :

L'application BugReport est une application web de remontée d'incidents. Celle-ci permet de rendre compte d'un problème à résoudre sur une application en particulier. Elle est accessible depuis internet, la résolution du problème peut être vu par tout le monde, ce qui permet une non duplication des problèmes postés.

2. Caractéristiques techniques :

- Langage de programmation : Php
(<http://www.php.net/manual/fr/>)
- Type de développement : Programmation Orientée Objet (POO)
- Utilisation de l'ORM (Object Relational Mapper): Doctrine
(<http://www.doctrine-project.org/>)
- Système de Gestion de Base de Données : MySQL (version 5.5.27)
- Serveur Web : Apache (version 2.4.3)
- Navigateurs web : Google Chrome (version 34.0.1847.131 m)
Opéra (version 21.0.1432.67)
- Plateforme de développement : XAMPP (version 3.1.0)
(<http://www.apachefriends.org/>)
- Environnement de Développement Intégré : Eclipse (version Luna)
(<http://www.eclipse.org/>)

3. Base de donnée :



C'est le SGBDR MySQL qui a été utilisé pour l'architecture de données de cette application BugReport en Php. Grâce à l'ORM Doctrine, qui gère la persistance des objets en base de données relationnelle, il est possible de générer les classes métiers de l'application directement à partir de la console avec le driver.

4. Framework :

- Pour mener à bien le projet BugReport (outil de gestion de projet), nous avons utilisé l'ORM (Object-Relationnal Mapping) Doctrine.
- Utilisation du système de partage collaboratif Google Code afin de synchroniser nos productions.
- Utilisation du ORM Doctrine afin de développer de façon plus cohérente en mode MVC (Les modèles d'un côté, les vues et les contrôleurs).

Partie II : Règles & Convention

1. General

Pour mener à bien ce projet, nous avons respecté les règles de nommage CamelCase.

2. Fonction :

Pour ce qui concerne ma partie, l'application permet de :

- Ajouter/Supprimer :
 - Une ou plusieurs useCase(s)
- Ajouter :
 - Une ou plusieurs Application
- Modifier :
 - La version de l'application
- Se connecter
- S'enregistrer en tant que nouvel utilisateur

Partie III : Structure de l'application

- BugReport
 - application (**Contient les fichiers sources de l'application et les configurations**)
 - Config
 - Controllers
 - CtrlHaut.php (**Controlleur d'accueil**)
 - CtrlInscription.php
 - CtrlGestionApplication.php
 - Models (**Contient tous les modèles de l'application**)
 - Mappings
 - Proxies
 - Application.php
 - Usecase.php
 - Utilisateur.php
 - Views(**Contient toute les vues**)
 - vInscription.php
 - vmajUseCase.php
 - Doctrine-cli.php
 - Assets (**Contient tous les ressources du projet**)
 - Css
 - Img
 - system
 - user_guide
 - index.php
 - PHP Include Path
 - PHP Language Library