

# Projet HelpDesk

Vous travaillez au sein d'une PME en tant que Responsable du support aux utilisateurs. Dans le cadre de votre travail, vous gérez aussi bien les incidents que les demandes d'assistance technique ou fonctionnelle sollicitées par les utilisateurs.

Les demandes d'assistance ou remontées d'incidents ne sont pour l'instant pas informatisées, et les utilisateurs doivent vous contacter directement, par mail ou par téléphone, pour vous communiquer ces informations.

Ce procédé est coûteux en temps, insatisfaisant pour les utilisateurs dont les demandes sont parfois oubliées. Il ne permet pas d'obtenir une traçabilité des actions d'exploitation menées.

Vous avez un temps envisagé la mise en place de solutions existantes (GLPI + OCS), mais ces solutions sont trop complètes, parfois trop complexes pour le SI de votre entreprise.

Vous avez donc décidé de réaliser une application Web permettant de gérer les demandes d'assistance, et dont les fonctionnalités sont adaptées à vos besoins.

## Fonctionnalités à mettre en place

2 des modules de l'application sont à mettre en place :

### -- Module report d'incidents

#### -- Création d'un ticket

- Acteur : utilisateur
- Données : ticket, statut, catégorie, utilisateur

Un utilisateur peut créer un ticket (report d'incident ou demande) dans une catégorie.

### Création d'un ticket :

The screenshot shows a form for creating a ticket. It has the following elements:

- Two dropdown menus: "Type" and "Catégorie", both with a checkmark icon on the right.
- A text input field labeled "Titre :".
- A large text area labeled "Description :".
- A text input field labeled "Utilisateur :".
- A text input field labeled "Date de création :".
- A text input field labeled "Statut :".
- Two buttons: "Valider" (green) and "Annuler" (teal).
- A tooltip box pointing to the "Utilisateur", "Date de création", and "Statut" fields, containing the text: "Ces 3 zones sont renseignées, mais en lecture seule".

### -- Echanges sur ticket

- Acteurs : Admin et/ou utilisateur
- Données : message, ticket, utilisateur

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

### -- Modification du statut

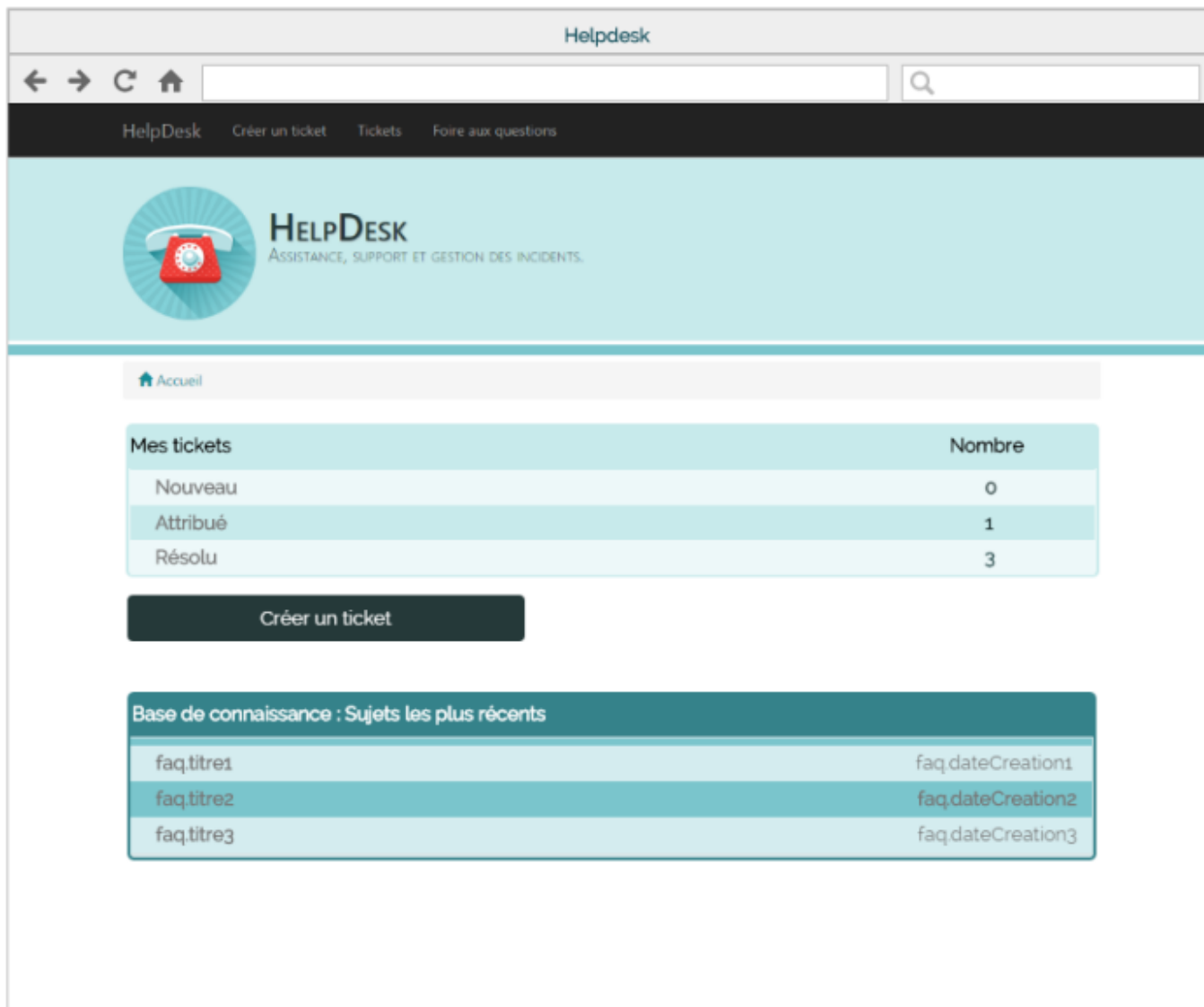
- Acteurs : Admin
- Données : ticket, statut

Les utilisateurs de type Admin peuvent modifier le statut d'un ticket existant

### -- Partie commune

- Acteurs : utilisateur
- Données : ticket, statut

Ecran d'accueil des utilisateurs :



## -- Module Base de connaissances

On appelle **article** un enregistrement de la table **FAQ**

### -- Création d'un article

- Acteur : Admin
- Données : faq, catégorie, utilisateur

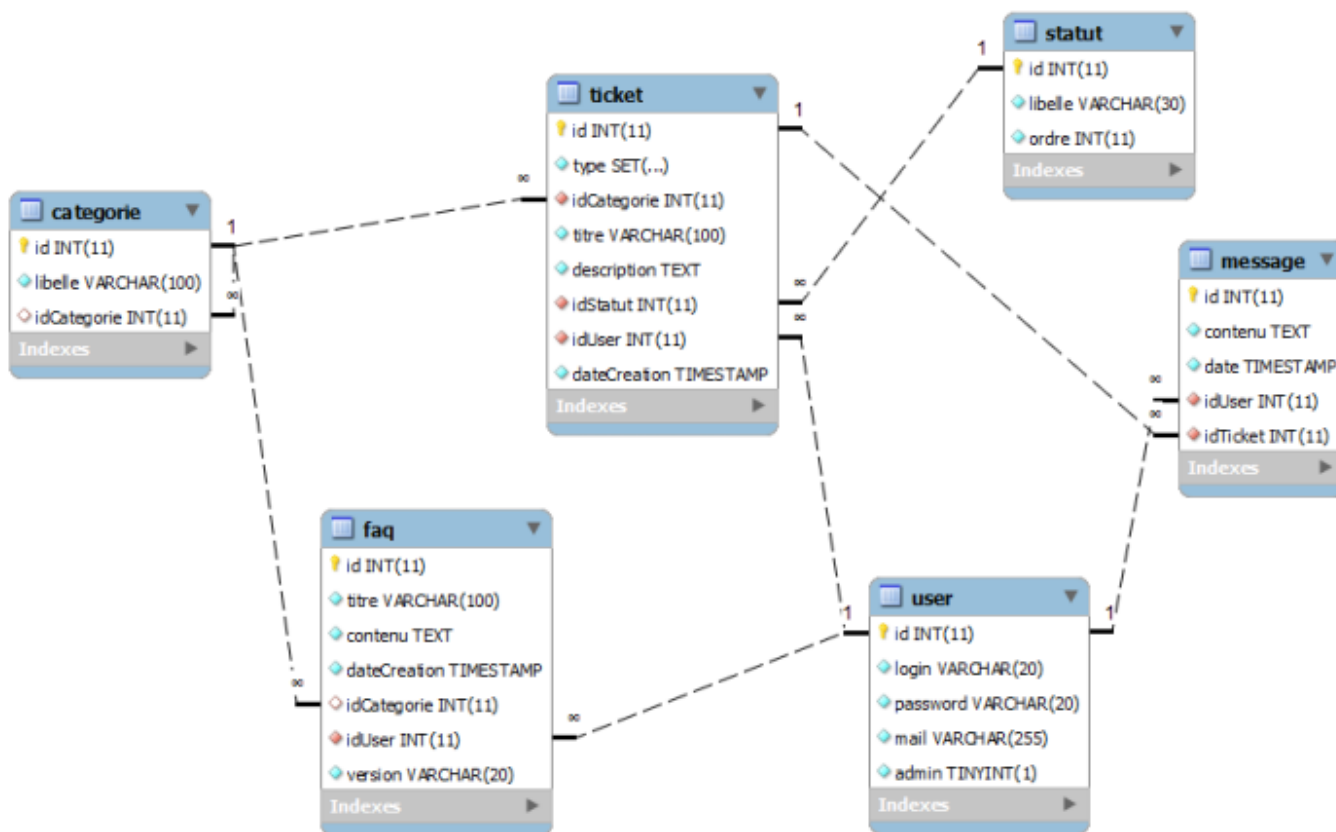
La base de connaissance est constituée par les utilisateurs de type Admin : il s'agit d'une FAQ.

### -- Consultation de la base

- Acteur : utilisateur
- Données : faq, catégorie

La base de connaissances est consultable par les utilisateurs : les articles sont classés par catégorie

## Base de données



## Contraintes techniques

- L'application sera développée en PHP, [Bootstrap](#) sera utilisé pour la partie présentation.
- Elle respectera au mieux la séparation des couches (objets Métiers), classes techniques et vues (interfaces web de saisie et d'affichage).
- Elle utilisera la base de données Mysql fournie en annexe. Cette base pourra évoluer en fonction des besoins du développement.
- L'utilisation de scripts côté client (javascript et ajax) pourra compléter les validations côté serveur. L'utilisation de frameworks est autorisée.

## Fichiers

- [Script de création de la base de données](#)
- [Grille d'évaluation projet](#)

## Modalités de remise du travail

- Date remise : rendre projet PHP + Lisezmoi.txt voir [Modalités de remise de votre travail](#)
- Date passage : soutenances + remise rapport (pdf)

# Compléments

## Bonnes pratiques

- Alimenter correctement la base de données en ajoutant des enregistrements valides et en nombre suffisant, mettant en valeur votre travail
- respecter la Normalisation HTML 5/Css 3
- Structurer les fichiers et dossiers de manière cohérente
- Nommer en respectant les normes et de manière significative (pages, fonctions, variables...)
- créer et utiliser des fonctions pour :
  - Accès à la base de données pour chargements ou mises à jour d'enregistrements
  - Affichages complexes (nécessitant php) dans les pages HTML
  - Fonctions utilitaires

From:

<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:

<http://slamwiki2.kobject.net/php-rt/projets/projet-2015?rev=1430957579>

Last update: **2019/08/31 14:26**

