2017

Bruno Avinint

BTS SIO 2016 option SLAM

IRUP



GSB Backoffice : client lourd

**Dossier GSB Client lourd :**

**Sommaire**

2 identifiants de connexion

2 technologies utilisées

3 Introduction

3 Fenêtre de login

**4 Accès visiteur**

4 Fenêtre profil utilisateur standard

4 Fenêtre changer mot de passe

**5 Accès administrateur**

5 Fenêtre gérer utilisateurs

6 Fenêtre afficher utilisateur

6 Fenêtre création

7 Fenêtre modification

7 Fenêtre importer utilisateurs

8 Fenêtre demande de mot de passe oublié

9 Schéma MCD base de données

**Dossier GSB – client lourd :**

**développement back-office**

**par Bruno Avinint**

**Identifiants de connexion :**

**Windows :**

*bavinint / jklr43*

**Application GSB :**

Admin : *bruno / 1234*

Visiteur : *kim / 1234*

**Technologies utilisées**

**L’application GSB backoffice a été développée en deux versions, une application web et une version client lourd, présentée dans ce document.**

Cette application web a été développée sur les technologies suivantes :

* Serveur MySQL Debian (même serveur que le serveur utilisé pour la version web)
* Langage: Ruby
* Librairies : Qt / QtRuby bindings / ActiveRecord, Securerandom, bcrypt et autres gems
* Outils : GIT / PhpMyAdmin / MySQLWorkbench / Sublime text / Notepad++ / Photoshop

**Introduction :**

Cette application client lourd a été réalisé avec le langage de programmation Ruby et la librairie Qt, célèbre librairie permettant la conception de composants graphiques, des accès en base de données, de la programmation réseaux etc. L’objectif ici était d’avoir une bibliothèque graphique permettant d’utiliser Ruby en programmation client. Les autres fonctionnalités de la librairie ne sont pas utilisées. A la place, Activerecord, ORM utilisé dans Ruby on Rails, est utilisé pour l’accès aux bases de données ainsi que plusieurs plugins (gems) de Ruby tels que Securerandom pour générer des tokens uniques, ou bien encore bcrypt, gem de cryptage, pour une compatibilité du cryptage des mots de passe entre les deux versions de l’application et la base de données partagée.

La version client lourd a été en effet conçue pour travailler avec une base de données partagée avec l’application web afin que les données soient les mêmes d’une version de l’application à l’autre.

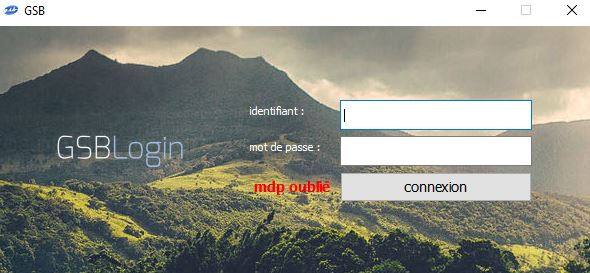
L’équipe a choisi de créer une application différente et indépendante pour chaque fonctionnalité. Ainsi l’application backoffice est différente de l’application frais ou l’application incidents. Chaque application requiert de se loguer et réagit d’une manière différente en fonction des droits d’accès de l’utilisateur.

Pour l’application back-office, seuls les administrateurs auront accès à la totalité de fonctionnalités, notamment la gestion des utilisateurs.

Un utilisateur lambda pourra : consulter son profil, modifier son mot de passe, et éventuellement réinitialiser son mot de passe s’il l’a perdu.

La totalité des fonctionnalités de la version web sont implémentées, à l’exception de l’upload de fichiers images des utilisateurs et l’affichage de leur portrait.

**fenêtre de Login :**



Marche à suivre proposée :

1. Se connecter dans un premier temps comme visiteur et constater les fonctionnalités accessibles :
2. Consulter son profil
3. Changer de mot de passe
4. Se connecter comme administrateur :

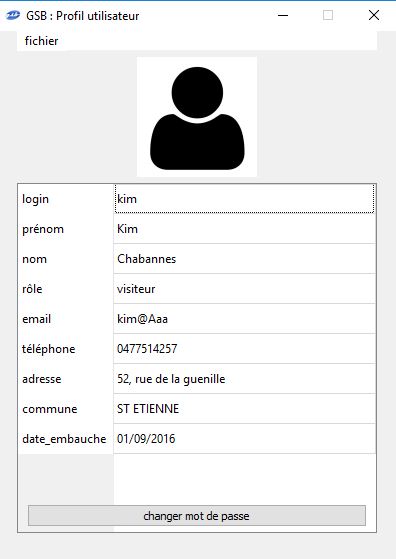
Constater les fonctionnalités décrites dans ce document

1. Se déconnecter et constater la fonctionnalité « mot de passe perdu »

Accès visiteur :

*(Se connecter les identifiants visiteur : login : kim / mdp : 1234)*

**Fenêtre profil standard :**



Une fois connecté, l’utilisateur arrive directement sur son profil. Les membres de l’équipe ont créé des applications autonomes (A part en ce qui concerne la base de données)

J’ai tout de même créé un accès non-administrateur dans mon application. Cela m’a paru plus intéressant qu’un simple message d’erreur affichant la nécessité d’un droit d’accès admin même si je reconnais que l’intérêt fonctionnel est limité, mais cela permet tout de même :

- A l’utilisateur :

o de consulter son profil

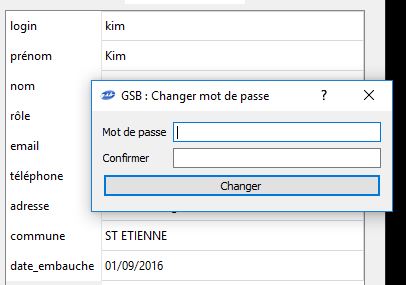
o de changer son mot de passe

- A l’entreprise :

o de permette à l’employé de consulter son profil (Loi informatique et libertés)

De plus, cette fonctionnalité, présente dans la version web, n’aurait pas été fournie par les autres applications.

**Fenêtre « changer mot de passe » :**

Il a été décidé que seul l’administrateur pourrait changer les données d’un utilisateur. Une exception a été faite pour les mots de passe. En effet, si les administrateurs doivent gérer les utilisateurs à l’échelle du pays, changer les mots de passe régulièrement ou spontanément en cas de problème pourrait être contraignant. Dans cette version, la fenêtre « changer de mot de passe » est une fenêtre modale.

Accès administrateur **:**

*(Se déconnecter et se reconnecter avec les identifiants admin : login : bruno/ mdp : 1234)*

**Fenêtre index (gestion des utilisateurs) :**



Affiche la liste de tous les utilisateurs avec un bouton « modifier » qui redirige vers une fenêtre dans laquelle un formulaire permet de modifier les données de l’utilisateur (voir plus loin).

Cliquer sur une colonne du tableau permet de sélectionner un utilisateur, ainsi que d’afficher un volet droit où les données de l’utilisateur sont affichées. En fait ce volet est une fenêtre autonome qu’on peut déplacer de manière autonome mais déplacer la fenêtre principale replacera le volet à côté de celle-ci. Fermer la fenêtre principale fermera également la fenêtre secondaire.

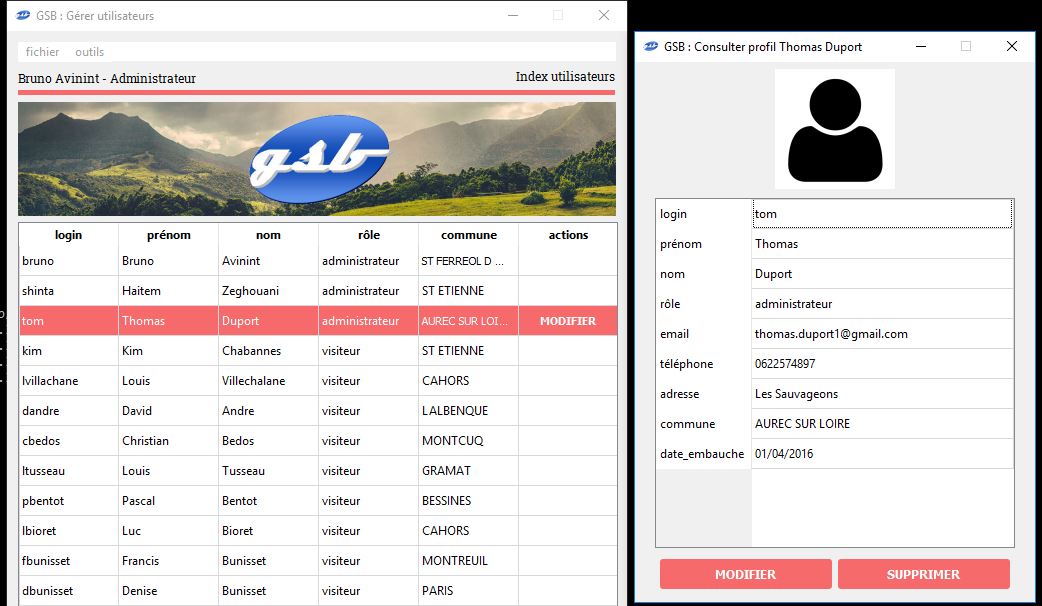
Un clic droit dans le tableau ferme la fenêtre secondaire.

La liste utilise un système de pagination permettant de ne pas afficher la totalité des données à la fois. Utile pour une entreprise comme GSB qui compte des centaines d’employés. Les accès base de données sont assez lents, vous le constaterez. La pagination permet de ne pas empirer le problème au fur et à mesure que la base de données est remplit. Cela ménage la performance de l’application quand la base de données commence à gérer plusieurs milliers d’utilisateurs.

Au bas de la fenêtre, un bouton « créer utilisateur » ouvrira le formulaire de création d’utilisateur en fermant la fenêtre principale.  Tandis que le bouton importer liste utilisateurs ouvre la fenêtre d’import.

Ce bouton ne ferme pas la fenêtre principale

**Fenêtre utilisateur (afficher utilisateur) :**

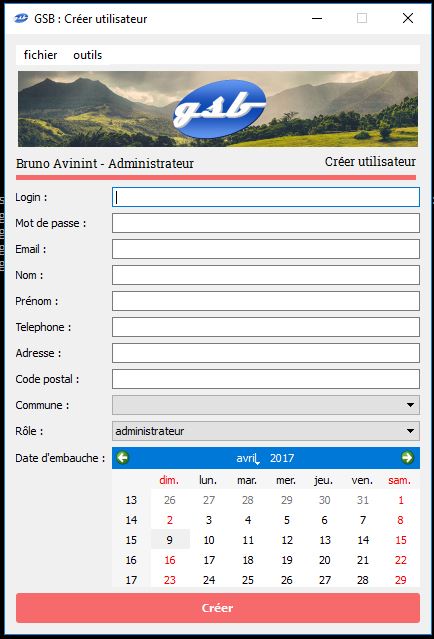


Permet de visualiser les données et propose également deux boutons qui permettent soit de modifier les données soit de supprimer l’utilisateur de la base. C’est la seule fenêtre où l’on peut supprimer un utilisateur.

La suppression fait apparaitre une boite de dialogue qui demande la confirmation de l’action.

L’application ne gère pas le « soft delete », toute suppression est donc définitive.

**Fenêtre création d’utilisateur :**

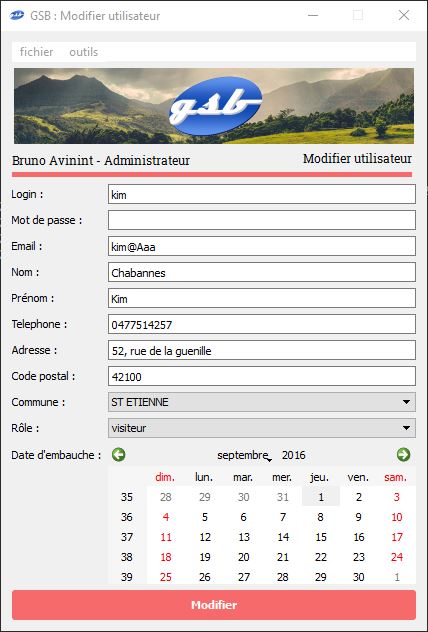


En plus de renseigner les identifiants, les informations essentielles, l’administrateur peut attribuer un rôle à l’utilisateur, ce qui conditionne ses droits d’accès. La création d’un compte provoque l’envoi d’un email à l’utilisateur concerné avec ses identifiants de connexion, pour peu que l’email soit valide.

Comme dans l’application web, la commune est renseignée en deux temps : On renseigne d’abord le code postal qui affiche une liste déroulante des communes correspondant à ce code postal, il faut alors sélectionner la commune souhaitée s’il y en a plusieurs dans la liste.

A l’heure actuelle, l’application ne gère pas les portraits des utilisateurs.

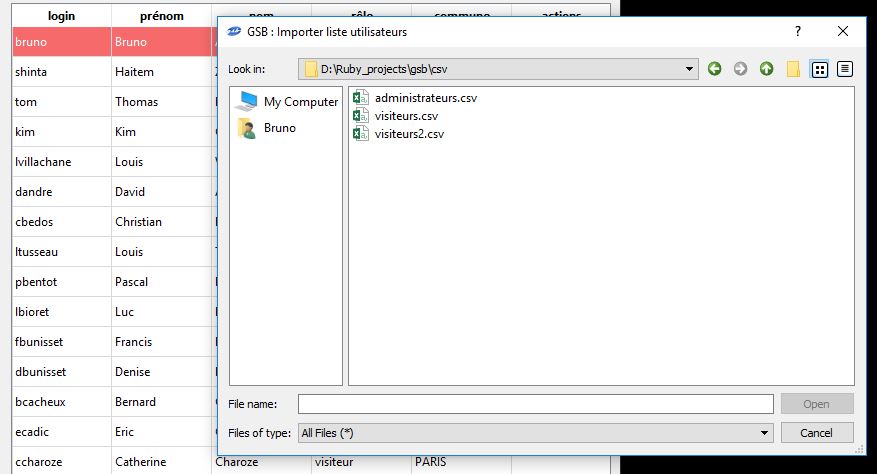
**Fenêtre modification :**



Le formulaire est généré sur le même modèle que le formulaire de création.

La seule différence est que le mot de passe n’est pas un champ obligatoire en modification.

**Fenêtre importer utilisateurs :**



Cette fenêtre affiche un formulaire permettant de récupérer le contenu d’un fichier csv généré sous Excel et d’importer les données converties en liste d’utilisateurs et de les sauvegarder dans la base de données de l’application.

**Fenêtre demande de mot de passe oublié :**

*(Se déconnecter)*

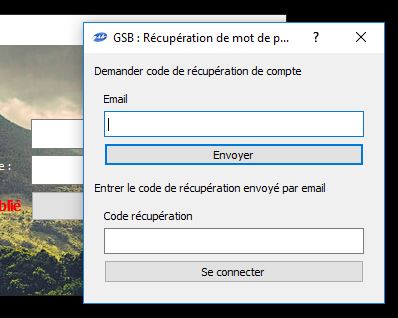
En plus du parcours ordinaire d’utilisateur connecté, l’utilisateur non connecté peut récupérer un mot de passe perdu. Pour cela il peut renseigner son adresse email ou son login et s’il est reconnu, un email lui sera envoyé, avec un lien de mise à jour utilisant un token.

Pour tester cette fonctionnalité, se connecter sur Gmail au compte :

[kim42.gsb@gmail.com](mailto:kim42.gsb@gmail.com) / mdp : azerty42

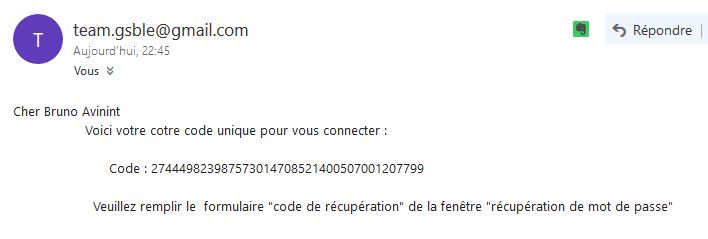
Se déconnecter en cliquant sur l’action déconnexion en haut à gauche dans le menu « Fichier » l’utilisateur sera redirigé vers la fenêtre de connexion et ensuite cliquer sur « mdp oublié » en rouge, à gauche du bouton « se connecter ». Cela appellera une fenêtre de dialogue dans lequel il doit taper son email et cliquer sur

« envoyer »



Vérifier qu’un email a été envoyé dans la boite mail ouverte.

Copier le token reçu par mail et renseigner le champ « code récupération » en bas de la fenêtre de dialogue et cliquer sur « se connecter »



Cela ouvre la fenêtre profil standard, dans laquelle l’utilisateur a la possibilité de changer son mot de passe en ouvrant la fenêtre de dialogue « changer de mot de passe » . S’il ferme la fenêtre de profil sans changer de mot de passe, il devra demander un code de récupération à nouveau car celui-ci n’est plus valable.

**Schéma MCD de la base de données :**

