

WHISKR

FICHE PRODUIT



#SAE302

#Security

#Network

<https://github.com/Ykari68/SAE302/>

Cette fiche produit a pour but de faire un point sur ce qui a été fait ou non vis-à-vis du produit whiskr.

Table des matières

Table des matières.....	2
I. Fonctionnalités.....	3
1/ Client sous la forme d'une interface graphique	3
2/ Authentification en lien avec le serveur.....	3
3/ Authentification sur le serveur pour réaliser des commandes	4
II. Côté serveur	5
1/ Seul le serveur a accès à la base de données (MySQL)	5
2/ Serveur sous forme texte	6
3/ Le serveur se connecte sur une base de données	6
4/ Commandes serveur	6
III. Côté client	7
1/ Graphique de forme originale.....	7
2/ Une page d'inscription avec l'identité de la personne (unique)	8
3/ Les clients peuvent demander à intégrer des salons :	8
4/ Communication privée avec statut de connexion d'un client	9
III. Conclusion.....	9

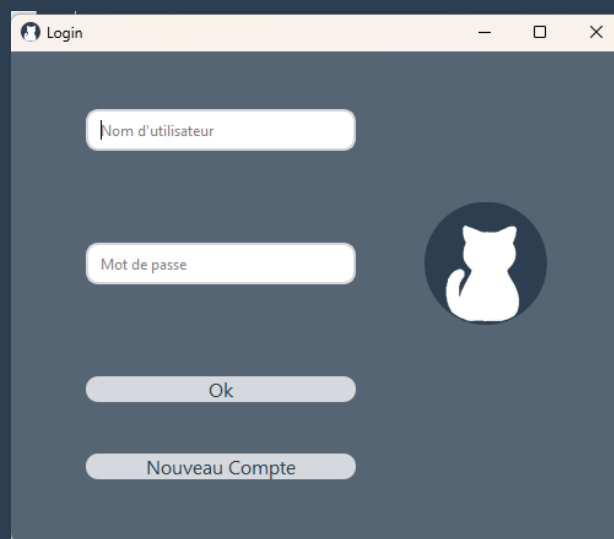


I. Fonctionnalités

1/ Client sous la forme d'une interface graphique

Le client pour votre entreprise Whiskr a été conçu avec une interface graphique fonctionnel et intuitive. Elle porte les couleurs de votre entreprise ainsi que son logo.

Voici un aperçu de l'interface graphique :



L'entrée du mot de passe est cachée et est crypté dans la base du serveur pour des raisons de sécurité.

L'utilisateur peut facilement s'y retrouver, avec des boutons de couleur blanche et facilement lisible, une interface simple et intuitive.

2/ Authentification en lien avec le serveur

Un système d'authentification est mis en place afin d'assurer une entrée sécurisée au chat du serveur.

Ce système est fonctionnel et oblige aux utilisateurs de s'authentifier avant de pouvoir utiliser le client.



3/ Authentification sur le serveur pour réaliser des commandes

Lorsque le serveur tourne, une console est lancée et demande un identifiant et un mot de passe.

Si les identifiants et mot de passe rentrés sont bon, alors l'administrateur pourra exécuter un certain nombre de commande.

```
root@debian:~# python3 SAE302/serveur/serveur.py
En attente de connexion...
Connecté à la base de données MySQL
Nom d'utilisateur : toto
Mot de passe : |
```

```
Authentication réussie.
Serveur> utilisateurs
Utilisateurs connectés:
```

4/ Le serveur enregistre les identifiants dans une base de données pour l'authentification

Une base de données est présente et une table est réservée pour la sauvegarde des identifiants et mot de passe d'utilisateur. Cette table est sollicitée par le serveur pour authentifier les clients.

```
+-----+
| Tables_in_serveur |
+-----+
| admin              |
| bannis             |
| historique         |
| utilisateurs      |
+-----+
```

```
mysql> select * from utilisateurs;
+----+-----+-----+
| id  | nom_utilisateur | mot_de_passe |
+----+-----+-----+
| 1   | john_doe       | 64751b0b2552c1f4654e975cbf56317a11fd74847222fb044cbc3217f68670c4 |
| 4   | toto          | 31f7a65e315586ac198bd798b6629ce4903d0899476d5741a9f32e2e521b6a66 |
+----+-----+-----+
```

Les mots de passe sont cryptés pour des raisons de sécurité.

5/ Le serveur enregistre les conversations émises par le client

Dans la base de données, il a été implémenté une table pour enregistrer les conversations émises par les clients et leurs dates. Cette table est sollicitée pour que le client puisse retrouver ses messages dans son application, mais également sous forme de commande administrateur.

Table MySQL :

```
mysql> select * from historique;
```

id	sender	message	timestamp
1	tata	test	2023-12-02 21:33:16
2	tete	test	2023-12-02 21:33:18
3	Serveur	tata a quitté la discussion.	2023-12-02 21:33:22
4	Serveur	tata a été kick.	2023-12-02 21:33:45

Commande serveur :

```
Serveur> historique
2023-12-02 21:33:16 - tata: test
2023-12-02 21:33:18 - tete: test
2023-12-02 21:33:22 - Serveur: tata a quitté la discussion.
2023-12-02 21:33:45 - Serveur: tata a été kick.
```

Historique client :

```
tata: test
tete: test
Serveur: tata a quitté la discussion.
Serveur: tata a été kick.
```

II. Côté serveur

1/ Seul le serveur a accès à la base de données (MySQL)

Dans le code, tous ce qui est lié à la base de données, c'est le serveur qui gère. Si le client a besoin de données, comme l'historique, il demande au serveur sous forme de message codé.

```
conn_db = mysql.connector.connect(
    host='127.0.0.1',
    user='root',
    password='toto',
    database='serveur'
```

Connexion à la base de données dans le code du serveur.

Si vous utilisez le client Whiskr pour la première fois, vous aurez certainement besoin d'un nouveau

2/ Serveur sous forme texte

Le serveur n'a pas été fait en graphique.

En effet, créer un serveur graphique a été jugé comme non nécessaire, étant donné que le serveur se doit d'être peu coûteux en ressource, et doit pouvoir tourner dans un serveur Linux non graphique, si l'entreprise utilise ce type de serveur.

Le serveur a donc été conçu sous forme textuel afin d'économiser des ressources et du temps.

3/ Le serveur se connecte sur une base de données

Comme expliqué dans le « 1/ », le serveur se connecte à une base de données afin de gérer différents éléments du produit, tel que les utilisateurs et leur mot de passe, l'historique des conversations ou alors la liste des utilisateurs bannis.

```
+-----+
| Tables_in_serveur |
+-----+
| admin              |
| bannis             |
| historique         |
| utilisateurs      |
+-----+
```

4/ Commandes serveur

Bien qu'il n'ait été convenu seulement trois commandes serveur (Kill, Kick et Ban), des commandes supplémentaires ont été ajoutées. Ces commandes sont utiles et indispensables pour un administrateur.

Voici toutes les commandes et leur effet.

Commande	<u>Effet</u>
kill	Fermeture du serveur et de ses clients

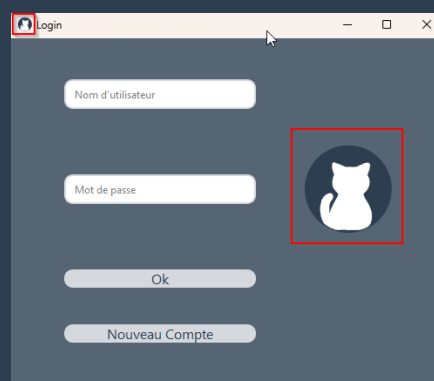
kick (Nom d'utilisateur)	Expulsion temporaire d'un utilisateur
ban (Nom d'utilisateur)	Expulsion définitive d'un utilisateur
register (Nom d'utilisateur) (mot de passe)	Création d'un nouvel utilisateur
regadmin (Nom d'utilisateur) (mot de passe)	Création d'un compte administrateur (console)
historique	Afficher l'historique du chat
utilisateurs	Afficher la liste des utilisateurs connectés

III. Côté client

1/ Graphique de forme originale

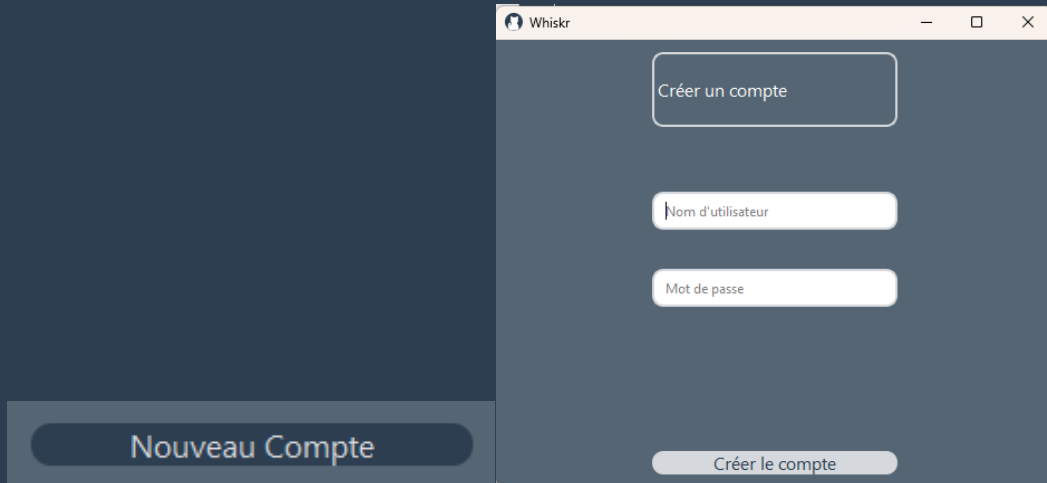
L'interface graphique a été pensée pour être unique et au couleur de votre entreprise Whiskr.

Elle contient le fond bleu marine, avec son blanc grisé. Le logo de Whiskr est présent sur l'application, notamment à la page de Login, en guise de bienvenue.



2/ Une page d'inscription avec l'identité de la personne (unique)

Une page d'inscription a été créée afin de permettre aux nouveaux utilisateurs de se créer un compte. Elle reste simple, un identifiant et un mot de passe est demandé.

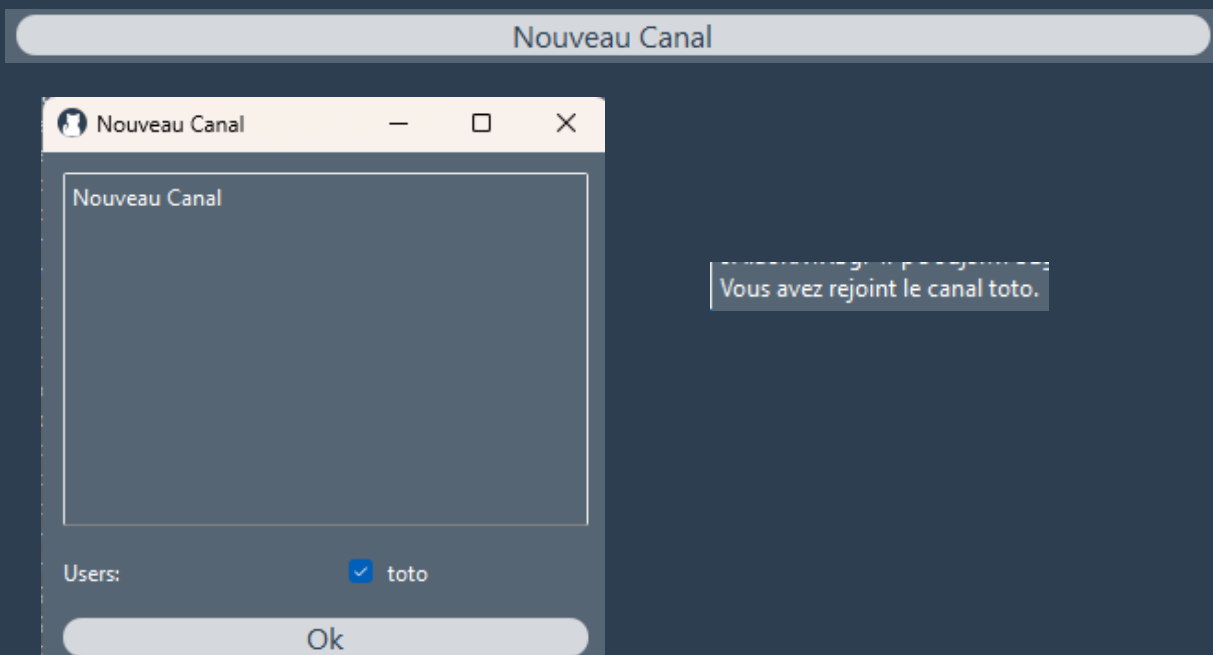


The image shows a registration form for Whiskr. It is a dark-themed window titled 'Whiskr'. At the top, there is a button labeled 'Créer un compte'. Below it are two input fields: 'Nom d'utilisateur' and 'Mot de passe'. At the bottom right, there is a button labeled 'Créer le compte'. To the left of the form, there is a button labeled 'Nouveau Compte'.

3/ Les clients peuvent demander à intégrer des salons :

Pour l'instant, la possibilité de rejoindre un salon n'est pas encore disponible, mis à part le salon Général, qui est le salon par défaut.

Cependant, il est possible de créer un canal personnalisé temporairement, afin de discuter avec un groupe d'utilisateur restreint.



The image shows a dialog box titled 'Nouveau Canal'. It has a large empty text area for a message. Below the text area, there is a section labeled 'Users:' with a list of users. The user 'toto' is selected with a blue checkmark. At the bottom, there is an 'Ok' button. To the right of the dialog box, there is a message box that says 'Vous avez rejoint le canal toto.'

4/ Communication privée avec statut de connexion d'un client

Cette fonctionnalité est également encours de production. Il n'est pas possible d'envoyer des messages privés à un utilisateur déconnecter. On ne peut pas non plus ajouter des utilisateurs en amis, ou voir son statut de connexion.

Si le client est connecté, on peut le parler en privé en créant un canal entre vous et lui.

III. Conclusion

En conclusion, le produit est prêt à être utilisé pour votre entreprise de manière sécurisé et confidentielle.

Cependant, il est encore en cours de développement et n'est pas encore complet. Bien que l'essentiel soit présent, il reste encore des fonctionnalités à ajouter, en termes d'ergonomie, de sécurité et d'efficacité.

Des travaux début 2024 sont à prévoir avec vous, en espérant pouvoir travailler à vos côtés afin de finaliser le développement de votre application.

Merci de m'avoir sollicité pour développer votre solution, c'est un honneur de travailler avec Whiskr.



#SAE302

#Security

#Network

<https://github.com/Ykari68/SAE302/>