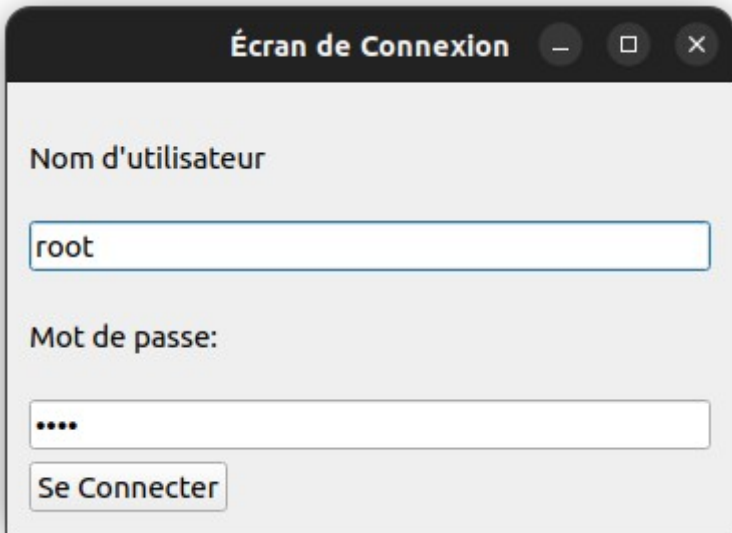


Suivi progression projet au 22/03

La première phase de développement consiste en la configuration de l'interface graphique. On la reliera plus tard à la BDD, lorsque tous les tabwidgets seront configurés et filtrés correctement. Pour cela, j'utilise le logiciel Designer lié à PyQt5. Il permet de take and drop les différents boutons et champs de texte en les plaçant librement sur l'interface. Il nous reste ensuite à coder les fonctions qui font les liens entre différentes actions, comme par exemple lier le bouton home à l'écran principal d'accueil, ou bien un bouton ajouter qui fera appel à une fonction à coder ultérieurement qui ajoutera un client dans la BDD.

Au lancement de l'application, on atterrit d'abord sur une fenêtre de connexion qui est nécessaire à toute utilisation du logiciel.

En effet, les données sensibles ayant besoin d'être protégées de toute intrusion externe, un compte principal, "admin", devra toujours conserver l'accès afin de permettre la création de nouveaux utilisateurs.

A screenshot of a Qt-style window titled "Écran de Connexion". The window has a dark title bar with standard minimize, maximize, and close buttons. The main area is light gray and contains two text input fields. The first field is labeled "Nom d'utilisateur" and contains the text "root". The second field is labeled "Mot de passe:" and contains four dots. Below the password field is a button labeled "Se Connecter".

Pour le moment, la fonction de connexion vérifie des chaînes de caractères stockées localement uniquement, mais elle fera plus tard appel à la base de données.

```

55 mycursor = mydb.cursor()
56
57 id_connexion = self.entrée_utilisateur.text()
58 mdp_connexion = self.entrée_mdp.text()
59
60 1 usage  adamowski13
61 def login(self): #en local
62     # Récupère les informations saisies par l'utilisateur
63     nom_utilisateur = self.entrée_utilisateur.text()
64     mot_de_passe = self.entrée_mdp.text()
65
66     # Vérifie les informations de connexion
67     if nom_utilisateur == 'root' and mot_de_passe == '1308':
68         QMessageBox.information(self, 'Connexion Réussie', 'Bienvenue, {}'.format(nom_utilisateur))
69         self.change_content() # Appel de la fonction pour changer le contenu
70     else:
71         QMessageBox.warning(self, 'Connexion Échouée', 'Nom d\'utilisateur ou mot de passe incorrect')
72
73 1 usage  adamowski13
74 def change_content(self):
75     self.setWindowTitle('Écran d\'Accueil')
76     # Supprime tous les widgets de la fenêtre de connexion
77     for i in reversed(range(self.layout().count())):
78         self.layout().itemAt(i).widget().setParent(None)
79
80     # Importe la nouvelle classe et ajoute les nouveaux widgets
81     accueil_window = Accueil()
82     self.layout().addWidget(accueil_window)
83
84 if __name__ == '__main__':
85     # Initialise l'application PyQt
86     app = QApplication(sys.argv)
87
88 Connexion().init_ui()

```

Ensuite, lors d'une connexion réussie, l'application renvoie vers un écran principal en fermant la première fenêtre (encore incomplète) qui propose plusieurs onglets (accueil-médicament-client-médecin-vente-commande-paramètres)

l'onglet vente va par exemple permettre de consulter les ventes ou d'en créer une nouvelle en prenant en compte l'utilisateur.

