

GSB AppliVisiteur

Contexte :

Le laboratoire Galaxy Swiss Bourdin (GSB) est issu de la fusion entre le géant américain Galaxy et le conglomérat européen Swiss Bourdin, lui-même déjà union de trois petits laboratoires.

Dans l'objectif d'uniformiser la gestion des frais des visiteurs médicaux des différents laboratoires, une application web sera mise en place.

// AJOUTER LE FICHIER DE CONTEXTE

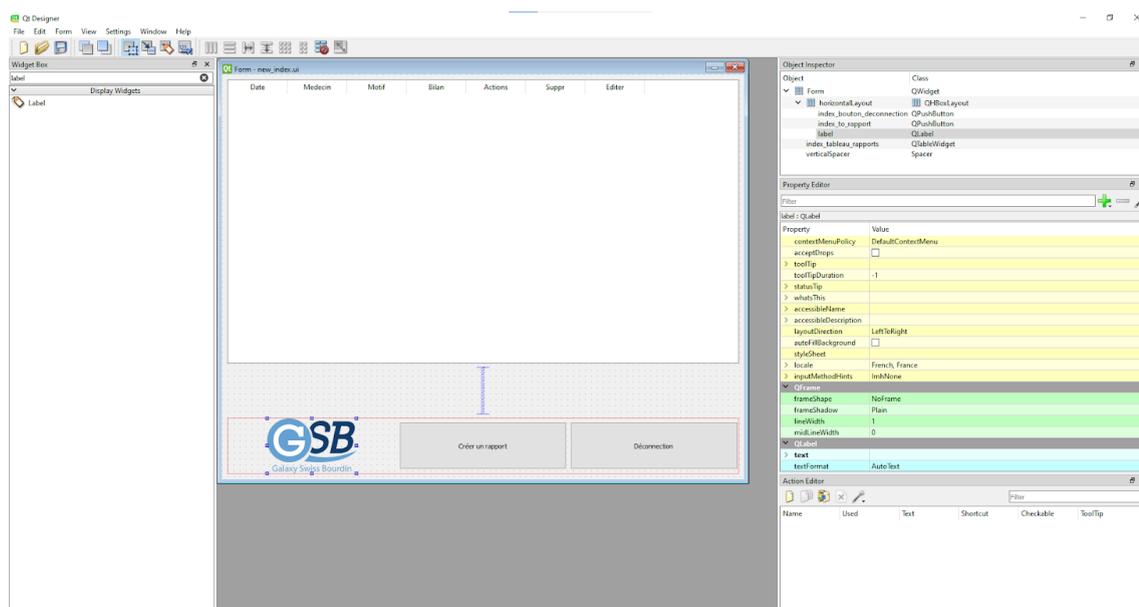
Langages utilisés :

- Python : Framework: PyQt6

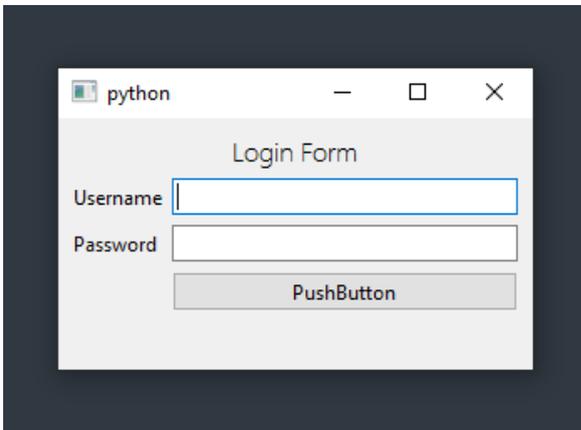
Développement :

Pour la mise en place du front pour l'application, nous utilisons le langage de programmation Python, avec la Bibliothèque PyQt et Requests. Ces bibliothèques Python sont faciles d'utilisation et permettent de faire l'interface graphique de notre software en liant la partie front avec l'API grâce à l'utilisation de la bibliothèque Requests.

Pour utiliser PyQt, nous avons choisi la version 6 avec Python 3.11. Afin de nous faire gagner du temps dans la création du client lourd de GSB, nous utilisons QT Designer, un logiciel permettant de faire une ou plusieurs fenêtres avec le module PyQt à sur une interface à l'aide de widgets et modules.

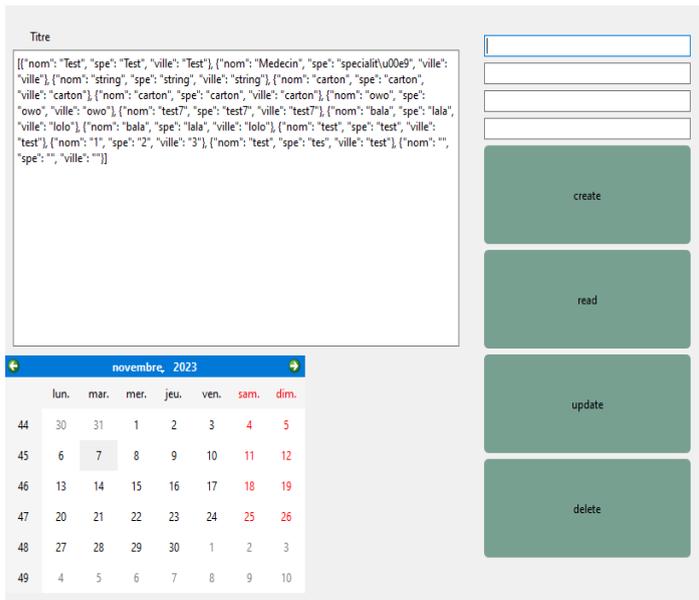


Afin de prendre en main le module PyQt nous avons commencé par créer une interface avec un CRUD (Create Read Update Delete) et un formulaire de connexion pour accéder à l'application.



Création de la page d'authentification dans un premier temps. Une fois la fenêtre créée et les widgets en place, nous sommes occupés du fonctionnement de la page de connexion au client lourd.

```
12 class Window(QWidget):
13     def __init__(self):
14         loadUi("login.ui", self)
15         button1.clicked.connect(self.login)
16
17     def gotoscreen2(self):
18         """
19         Change la fenêtre à afficher
20
21         Affiche la fenêtre suivante
22         """
23         widget.setCurrentIndex(widget.currentIndex()+1)
24
25     def login(self):
26         """
27         Envoie une demande d'authentification à l'API
28
29         - Récupère les informations renseigné dans les champs du formulaire
30         - Envoie une requête à l'API avec les informations renseignées en paramètre
31         - Si le code Http retour est "200" alors : appelle de la méthode permettant le changement de fenêtre
32         - Sinon renvoie un message d'erreurS
33         """
34         login = self.input1.text()
35         mdp = self.input2.text()
36         logedin = False
37
38         x = requests.post(f'http://127.0.0.1:8000/login',
39                         data={
40                             "grant_type": "",
41                             "username": login,
42                             "password": mdp,
43                             "scope": "",
44                             "client_id": "",
45                             "client_secret": ""
46                         })
47
48         if x.status_code==200:
49             logedin = True
50             self.gotoscreen2()
51         else:
52             print('Username or Password are wrong !')
```



Création de la fenêtre avec un CRUD.
 Un widget permet de d'afficher ce que contient la base de donnée, 3 champs permettent à l'utilisateur d'entrer des informations et les boutons en dessous permettent d'effectuer une certaine action du CRUD sur la DB avec les informations renseignées.

```

55 class Screen2(QMainWindow):
56     def __init__(self, login): ...
65     def set_list(self): ...
71     """
72     Renvoie la liste des médecins de la base de donnée
73     """
74     def gotoCpRendu(self): ...
76     """
77     Change la fenêtre à afficher
78     """
79     def createDr(self): ...
89     """
90     Envoie les informations renseigné afin d'ajouter un médecin dans la base de donné
91     """
92     def updateDr(self): ...
103     """
104     Modifie les informations d'un médecin désigné dans la base de donné
105     """
106     def deleteDr(self): ...
110     """
111     Supprime un médecin de la base de donnée
112     """
  
```

```

55 class Screen2(QMainWindow):
56     def __init__(self, login):
57         super().__init__()
58         super(Screen2, self).__init__()
59         loadUi("main.ui", self)
60         self.set_list()
61         self.index_button_create.clicked.connect(self.createDr)
62         self.index_button_update.clicked.connect(self.updateDr)
63         self.index_button_delete.clicked.connect(self.deleteDr)
64         self.index_button_read.clicked.connect(self.gotoCpRendu)
65     def set_list(self):
66         x = requests.get('http://127.0.0.1:8000/medecins')
67         jason = x.json()
68         fake = json.dumps(jason)
69         self.List.clear()
70         self.List.setText(fake)
71     def gotoCpRendu(self):
72         widget.setCurrentIndex(widget.currentIndex()+1)
73     def createDr(self):
74         nom = self.dr_nom.text()
75         spe = self.dr_spe.text()
76         ville = self.dr_ville.text()
77         x = requests.post(f'http://127.0.0.1:8000/create_medecin', json={
78             "nom":nom,
79             "spe":spe,
80             "ville":ville
81         })
82         self.set_list()
83     def updateDr(self):
84         id_dr = self.dr_id.text()
85         nom = self.dr_nom.text()
86         spe = self.dr_spe.text()
87         ville = self.dr_ville.text()
88         x = requests.put(f'http://127.0.0.1:8000/update_medecin/{id_dr}', json={
89             "nom":nom,
90             "spe":spe,
91             "ville":ville
92         })
93         self.set_list()
94     def deleteDr(self):
95         id_dr = self.dr_id.text()
96         x = requests.delete(f'http://127.0.0.1:8000/delete_medecin/{id_dr}')
97         self.set_list()
98
  
```

Une fois le CRUD réalisé nous avons changé la structure du code de l'application.

```
widget = QtWidgets.QStackedWidget()
login = Window()
menu = Screen2(login)
CpRendu = CpRendu()
widget.addWidget(login)
widget.addWidget(menu)
widget.addWidget(CpRendu)
```

Avant la modification, l'application fonctionnait comme un liste dans laquelle on se déplaçait de cette façon :

```
def gotoscreen2(self):
    """
    Change la fenêtre à afficher

    Affiche la fenêtre suivante
    """
    widget.setCurrentIndex(widget.currentIndex()+1)

def login(self): ...
```

```
16 class User():
17     def __init__(self, access_token, user_id): ...
23
24     def getUserDatas(self): ...
32
33 class Login_page(QtWidgets.QWidget):
34     def __init__(self): ...
38
39     def login(self): ...
69
70     def doSomethingNext(self): ...
77
78 class Index_page(QtWidgets.QWidget):
79     def __init__(self): ...
89
90     def goToAdmin(self): ...
95
96
97     def doSomethingNext(self): ...
104
105     def setUserDatas(self): ...
115
116
117     def setRapportList(self): ...
160
161     def editRapport(self, RAP_NUM): ...
175
176     def goToRapport(self): ...
181
182     def suppr(self, id_RAP): ...
190
191 class Admin_page(QtWidgets.QWidget):
192     def __init__(self): ...
196
197     def goToIndex(self): ...
202
203     def doSomethingNext(self): ...
226
227     def setVisRapports(self, id_vis): ...
269
270     def suppr(self, id_RAP): ...
278
279     def goToRapport(self): ...
284
285     def editRapport(self, RAP_NUM): ...
299
```

```
300 class Rapport_page(QtWidgets.QWidget):
301     def __init__(self): ...
310
311     def setRapportResume(self): ...
317
318
319     def setMedecins(self): ...
332
333     def setMedicaments(self): ...
342
343
344     def doSomethingNext(self): ...
349
350     def goToIndex(self): ...
355
356     def sendRapport(self): ...
402
403 class Stack(QtWidgets.QStackedWidget):
404     def __init__(self): ...
416
417     def deconnection(self): ...
426
427     def initCurrent(self): ...
430
431     def launchIndex(self, access_token, id_user): ...
444
445
446 if __name__ == "__main__":
447
448
449     today = date.today()
450     todayFr = today.strftime("%Y-%m-%d")
451     app = QApplication(sys.argv)
452     appStack = Stack()
453     appStack.show()
454
455     sys.exit(app.exec())
```

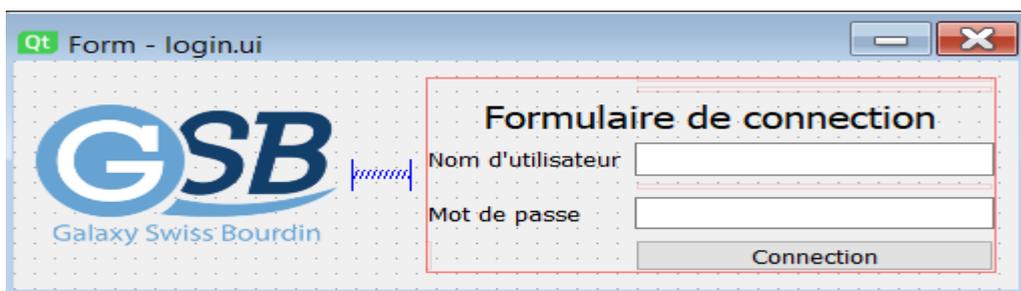
De cette façon, la class Stack peut être représenté comme un liste associative, désormais afin de changer de fenêtre il faut que la faut ajouter la class de la fenêtre comme attribut de la class " Stack " et appeler la fenêtre à l'aide de la méthode " setCurrentWidget() " .

```
427 def launchIndex(self,access_token,id_user):
428     """
429     Initie les pages de l'application
430     """
431     self.user = User(access_token,id_user)
432     self.admin_page = Admin_page()
433     self.index_page = Index_page()
434     self.rapport_page = Rapport_page()
435     self.addWidget(self.admin_page)
436     self.addWidget(self.index_page)
437     self.addWidget(self.rapport_page)
438     self.setCurrentWidget(self.index_page)
439
```

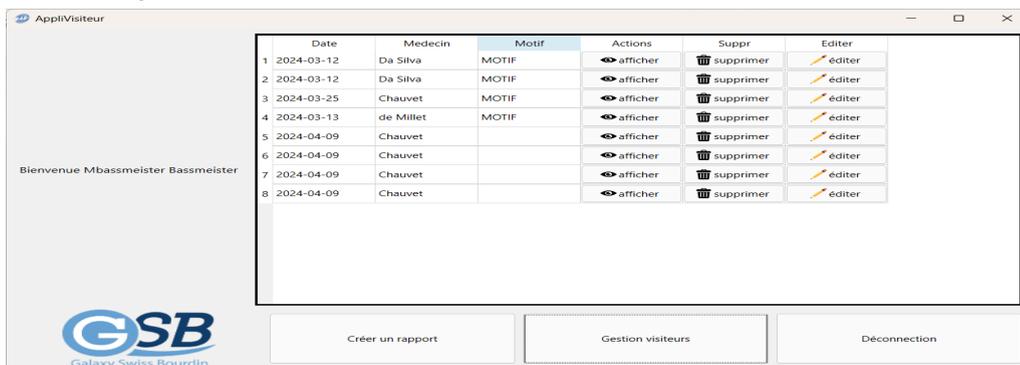
```
90 def goToAdmin(self):
91     """
92     Affiche la fenêtre "Admin"
93     """
94     appStack.setCurrentWidget(appStack.admin_page)
95
```

Liste des pages de l'application :

Page d'authentification :



Page d'accueil :



Page de gestion des visiteurs (pour les chef de secteur) :

Qt Form - admin.ui

Date	Medecin	Motif	Action	Suppr
------	---------	-------	--------	-------

Nom TextLabel
Prénom TextLabel
Nombre de rapports TextLabel Voir

Nom TextLabel
Prénom TextLabel
TextLabel TextLabel Voir

Nom TextLabel
Prénom TextLabel
TextLabel TextLabel Voir

Retour

Page d'insertion de rapport :

AppliVisiteur

Médecins
Delaunay Valérie-Monique

Échantillons Quantité

ANASTROZOLE ACCORD 0

ANASTROZOLE ACCORD 0

Motif :

Commentaire :

Envoyer le rapport Retour à l'accueil

Page d'accueil :

Un tableau sur la page d'accueil de l'application permet de visualiser et de gérer les rapport insérés par l'utilisateur connecté :

	Date	Medecin	Motif	Actions	Suppr	Editer
1	2024-03-12	Da Silva	MOTIF	 afficher	 supprimer	 éditer
2	2024-03-12	Da Silva	MOTIF	 afficher	 supprimer	 éditer
3	2024-03-25	Chauvet	MOTIF	 afficher	 supprimer	 éditer
4	2024-03-13	de Millet	MOTIF	 afficher	 supprimer	 éditer
5	2024-04-09	Chauvet		 afficher	 supprimer	 éditer
6	2024-04-09	Chauvet		 afficher	 supprimer	 éditer
7	2024-04-09	Chauvet		 afficher	 supprimer	 éditer
8	2024-04-09	Chauvet		 afficher	 supprimer	 éditer

On peut y voir certaines informations du rapport qui permettent d'identifier précisément un rapport et trois boutons sont présent sur chaque ligne :

- Supprimer :

Permet de supprimer le rapport de la base de donnée

- Editer :

Permet de modifier les informations du rapport

- Afficher :

Permet d'afficher les données du rapport en pdf et d'enregistrer le document.



Fiche de présentation rapport 9

Praticien : Dr. Chauvet Michelle

Médicament(s) :

TRAMADOL CRISTERS

Quantité : 50

Composition : MACROGOL 3350

Effet(s) : Nisi illum inventore aspernatur velit. Molestiae in inventore

Contre-indication(s) : liste I

Prix : 10

TRAMADOL CRISTERS

Quantité : 20

Composition : MACROGOL 3350

Effet(s) : Nisi illum inventore aspernatur velit. Molestiae in inventore

Contre-indication(s) : liste I

Prix : 10

Page de gestion des visiteurs :

The screenshot shows the 'AppliVisiteur' application window. On the left, there are three visitor detail cards. The first card for 'Russian' shows 'Rvillageboys' as the first name and '2' reports. The second card for 'Jack TenaciousD' shows '0' reports. The third card for 'Saez' shows '1' report. A 'Voir' button is present on each card. At the bottom left of the sidebar is a 'Retour' button. The main area contains a table with the following data:

	Date	Medecin	Motif	Action	Suppr	Editer
1	2024-04-18	de Millet		👁️ afficher	🗑️ supprimer	✏️ éditer
2	2024-04-18	Le Laine		👁️ afficher	🗑️ supprimer	✏️ éditer

Cette page permet au chefs de secteur de pouvoir gérer et visualiser les rapport renseignés par les 3 visiteurs qui lui sont attribués.

On peut y voir le nom, prénom et le nombre de rapports des visiteurs, cliquer le sur bouton "voir" permet d'afficher les rapport du visiteur dans le tableau ci-dessus.

Page d'insertion de rapport :

Médecins
Delaunay Valérie-Monique

Échantillons Quantité
ANASTROZOLE ACCORD 0

ANASTROZOLE ACCORD 0

Motif :

Commentaire :

Envoyer le rapport Retour à l'accueil

Liste déroulante des médecins:

Médecins

Delaunay Valérie-Monique

Delaunay Valérie-Monique

Le Laine Nathalie

Petitjean Étienne

Pruvost-Bouchet Julie

Leclercq Suzanne

Chauvet Michelle

de Millet Adrien

Da Silva Margot

Franck Thibault

Rolland Martine

Liste déroulante des médicaments :

ANASTROZOLE ACCORD

ANASTROZOLE ACCORD

RANITIDINE BIOGARAN

ACTAEA RACEMOSA

CURCUMA XANTHORRHIZA

BECLOSPIN 800

FENOFIBRATE TEVA

TRAMADOL EG

CABAZITAXEL REDDY

FAMOTIDINE EG

RHODODENDRON FERRUGINEUM

Évolutions possibles :

- Afficher les médecins non visités depuis X temps.
- Visualisation des rapports de tout les visiteurs, uniquement pour les responsable de secteur.
- Afficher les médecins qui ont reçu X échantillon.