

Nom et prénom : RIPERT Jerome    N° candidat : 02146743760

**BTS SERVICES INFORMATIQUES AUX ORGANISATIONS**

**EPREUVE E5**

– Conception et développement d'applications –

Réalisation professionnelle 2 : Appli Visiteur

Spécialité : Services informatiques aux organisations  
option B : Solutions logicielles et applications métiers

**SESSION 2022**

<b>DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE</b>		<b>N° réalisation :</b>
Nom, prénom : RIPERT Jerome		N° candidat : 02146746354
Épreuve ponctuelle <input checked="" type="checkbox"/>	Contrôle en cours de formation <input type="checkbox"/>	Date : ..... / ..... / .....
<b>Contexte de la réalisation professionnelle</b> Galaxy Swiss Bourdin est issu de la fusion entre deux grands du milieu pharmaceutique l'américain Galaxy et l'européen Swiss Bourdin. Il souhaite donc une modernisation de l'activité médicale.		
<b>Intitulé de la réalisation professionnelle</b> Application métier de gestion de comptes-rendus pour les visiteurs		
<b>Période de réalisation :</b> 04/10/2021-31/03/2022 <b>Lieu :</b> Ecole supérieure Aristée- La Valette du Var <b>Modalité :</b> <input type="checkbox"/> Seul(e) <input checked="" type="checkbox"/> En équipe		
<b>Compétences travaillées</b> <input checked="" type="checkbox"/> Concevoir et développer une solution applicative  <input checked="" type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative  <input checked="" type="checkbox"/> Gérer les données		
<b>Conditions de réalisation<sup>1</sup> (ressources fournies, résultats attendus)</b>  ressources fournies: Documentation du contexte, cahier des charges.  résultats attendus: Création d'un logiciel métier de gestion de compte-rendu de visite en Python objet, sous forme de client-serveur sécurisé et authentifié.		
<b>Description des ressources documentaires, matérielles et logicielles utilisées <sup>2</sup></b>  ressources documentaires: cahier des charges et contexte gsb.  ressources matérielles : Framework Flask, PyQt, Python, API.  logicielles utilisées: Visual Studio Code, GitHub, SQLite, Trello, Qt Designer, Discord, Postman.		

<sup>1</sup> En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

<sup>2</sup> Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

Modalités d'accès aux productions <sup>3</sup> et à leur documentation <sup>4</sup>

Modalités d'accès aux productions:

documentation: <https://cloud.aristeecampus.org/index.php/s/R7GliZCJZJh4XSc>

Mot de passe : Jury.2022

Puis RP2

**BTS SERVICES INFORMATIQUES AUX ORGANISATIONS**

**SESSION 2022**

**ANNEXE 7-1-B : Fiche descriptive de réalisation professionnelle  
(verso, éventuellement pages suivantes)**

**Épreuve E5 - Conception et développement d'applications (option SLAM) - Coefficient 4**

---

<sup>3</sup> Conformément au référentiel du BTS SIO « Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve. ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

<sup>4</sup> Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemple service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

### **Contexte:**

En 2003, le laboratoire Swiss Bourdin avait mené un travail d'informatisation des comptes-rendus, qui devait permettre aux visiteurs de disposer d'un outil de gestion personnelle de leur activité et, par un module d'exportation, d'adresser aux délégués régionaux une synthèse mensuelle. L'application était prévue sous Access et certaines parties ont été développées. Toutefois, par suite de diverses réorientations de la stratégie, ce développement n'a plus été une priorité et le système, en l'état, n'a été diffusé qu'à quelques visiteurs pour évaluation. On souhaite repartir de l'étude de 2003 (application web) et produire une nouvelle application de type client serveur. Des entretiens menés en début d'année actualisent le contexte depuis l'analyse de 2003.

Le laboratoire GSB souhaite moderniser l'activité de visite médicale et mettre à la disposition des visiteurs médicaux une application Web permettant la centralisation des comptes rendus (rapports) de visites.

Pour ce faire, il souhaite disposer d'une application Web donnant accès à toutes les informations sur les médicaments, les praticiens, les visiteurs et les rapports de visites.

Le laboratoire demande dès lors une application client-serveur sécurisée et authentifiée, de type client lourd.

La fenêtre d'accès ne doit proposer qu'une zone d'identification. L'application doit être accessible via le réseau de l'entreprise ou via une connexion sécurisée (ex/VPN). L'application ne devra en aucun cas permettre un accès direct aux bases de données pour des raisons de sécurité (elle devra passer par un programme de gestion des demandes).

Appli Frais est un projet en équipe j'ai principalement fait la conception les interface graphique des pages de consultation et saisie et aussi tout les schéma de la documentation

Fenêtre de connexion:

The image shows a login window design. It features a dark gray rounded rectangle centered on a light blue gradient background. At the top of the dark rectangle is a circular icon containing a black silhouette of a person. Below the icon, the text "SE CONNECTER" is displayed in white, uppercase letters. Underneath, there are two input fields. The first is labeled "Nom d'utilisateur" in white, with a light gray placeholder text "Nom d'utilisateur" below it. The second is labeled "Mot de passe" in white, with a light gray placeholder text "Mot de passe" below it. At the bottom of the dark rectangle is a light blue rounded button with the text "CONNEXION" in white, uppercase letters.

Fenêtre de saisie:

AppliVisiteur

GSB

APPLIVISITEUR

Saisie de rapport

Accueil

Praticien : Adèle Diallo

Date du rapport : 02/05/2022

Motif : Exemple : Devis

Bilan :

Sélection d'échantillons : -Aucun-

Nombre d'échantillons : 0

Ajouter

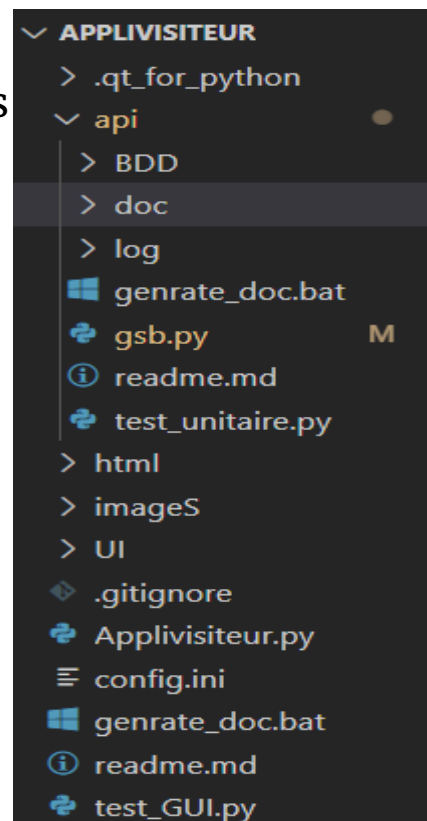
Nom	Quantité
-----	----------

OK

## Présentation du code:

Toute les fonctions faisant appel à l'api ou aux différentes pages sont regroupées dans Applivisiteur.py

L'api est dans gsb.py elle va faire des requêtes à la base de donnée et les renvoyer.



fonction qui renvoie à la fenêtre de consultation:  
chaque page est constituée de widgets qui va être stockés dans de UI  
pour créer la page

```
@staticmethod
def aller_vers_fenetre_consultation():
    """
        Fonction de redirection vers la fenetre consultation

        Returns
        -----
        None
    """
    fenetre_consultation = FenetreConsultation()
    win.stackedWidget.addWidget(fenetre_consultation)
    win.stackedWidget.setCurrentWidget(fenetre_consultation)

    return None
```

Fonction qui ajoute des médicaments à la liste :

```

# fonction pour ajouter des médicaments a la liste
def ajouter_medicament(self):
    """
    Fonction d'ajout de médicament dans le rapport'

    Returns
    -----
    None
    """
    # si il y a des éléments de sélectionnés
    if self.SelectionMedicamentNom.currentText() != "-Aucun-" and self.SelectionMedicamentQuantite.text() != "":

        # je crée un dictionnaire contenant tous les médicaments choisis
        liste_medicament_enregistre = {}
        for i in range(0, self.TableauOffreMedicaments.rowCount()):
            liste_medicament_enregistre[self.TableauOffreMedicaments.item(i, 0).text()] = i

        # si le médicament n'a pas déjà été entré, je l'intègre
        if self.SelectionMedicamentNom.currentText() not in liste_medicament_enregistre.keys():
            self.TableauOffreMedicaments.insertRow(self.TableauOffreMedicaments.rowCount())

            self.TableauOffreMedicaments.setItem(self.TableauOffreMedicaments.rowCount() - 1, 0,
                                                    QTableWidgetItem(self.SelectionMedicamentNom.currentText()))
            self.TableauOffreMedicaments.setItem(self.TableauOffreMedicaments.rowCount() - 1, 1,
                                                    QTableWidgetItem(self.SelectionMedicamentQuantite.text()))

        # sinon je modifie la valeur de la quantité
        else:
            self.TableauOffreMedicaments.item(
                liste_medicament_enregistre[self.SelectionMedicamentNom.currentText()], 1) \
                .setText(self.SelectionMedicamentQuantite.text())
            del liste_medicament_enregistre
    else:
        print("il manque des informations pour pouvoir ajouter un échantillon")

    return None

```



Exemple de requête à l'api, ici on veut obtenir les médecins:

On va donc demander à l'api grâce à une requête d'aller chercher les infos dans la BDD (ici, dans la table medecins)

```
# requete pour obtenir les medecins
requete = requests.get(
    Utils.url_api() + '/GSB/medecins',
    cookies=dict(session=visiteur.token))

if Utils.check_code_status(requete.status_code)['status']:

    medecins_json = json.loads(requete.text)
    del requete
```

### Conclusion:

L'application "APPLIVISITEUR" est fonctionnelle et prête pour toute utilisation, (saisie de rapport, consultation de rapport)

### Evolutions possibles:

création d'un calendrier agenda

fonctionnalité permettant de voir les médicaments

voir et trier les praticiens par secteur

--