

SAE 3.02

Développer des Applications Communicantes

CAHIER DES CHARGES:

1) Contexte et présentation du sujet

Dans cette SAE de programmation, l'objectif est de développer une application mobile qui permet un échange d'informations au sein d'un environnement technologique en liaison avec la gestion simplifiée de dossiers étudiants.

Le but de l'application sera de récupérer des données d'étudiants, comme l'année d'étude, l'id, le nom, les matières et une photo de chaque étudiant que l'on veut ajouter à une base de données, puis de pouvoir l'afficher en consultant cette même base de données. Ainsi on reproduit une version simplifiée des dossiers étudiants.

Comme vous auriez pu le comprendre le but de ce projet est de faire communiquer un programme python avec une base de données qui sauvegardera les données échangées pour en faire une application mobile pour smartphone android , une sorte d'application client/serveur

Il s'agit de reproduire dans ce projet certains exercices de tp vue en classe, mais avec un approfondissement sur certains aspects, tel que la programmation python des BDD.

Pour ce faire nous allons utiliser plusieurs modules tels que le

- R3.10 Gestion d'un système de Base de Données
- R3.09 Programmation Événementielle
- R3.08 Consolidation de la Programmation

2) Besoins et contraintes liés au projet

Dans le cadre de ce projet nous devons respecter quelques consignes qui sont les suivantes:

- Présenter un cahier des charges en mentionnant les technologies à utiliser et éventuellement leur impact environnemental et économique
- Développer une application client-serveur
- Authentifier les utilisateurs
- Sauvegarder les données échangées
- Concevoir une interface graphique, une application mobile
- Présenter les résultats lors d'une soutenance (rapport écrit et code source des programmes)
- L'objectif de ce genre d'exercice est de mettre le(s) participant(s) dans un environnement proche du réel.
- Les résultats du travail doivent être documentés au fur et à mesure de l'avancement sur le projet avec les commandes et la preuve de l'exécution en ScreenShots (captures d'écran).

Pour décrire la solution, on peut citer les trois aspects suivant :

- **L'analyse et l'expression des besoins fonctionnels :**

L'application doit pour fonctionner être capable de récupérer des données renseignées dans les champs réservés à cet usage, de les transmettre à un serveur en les renseignant dans une bdd. On pourra alors récupérer dans cette bdd les informations que l'on veut pour les traiter et les réafficher par la suite du côté client sur une application mobile.

- **Les contraintes techniques et l'architecture :**

Le projet doit respecter des contraintes, telles que la relation et transmission de données client/serveur. L'application devra être développée à l'aide de python et kivy, et les bases de données seront créées et manipulées avec python. Pour que l'application passe d'ordinateur à mobile nous devons employer buildozer. Ainsi il y aura au total deux bases de données, un script kv, un script python pour kv, et un script python pour les bdd. Chaque étudiant renseignera dans la base de données aura son propre et unique "id", qui seront également les clés primaires des

bdd. Il y aura au minimum 10 étudiants renseignés ainsi qu'au moins 5 matières pour la moyenne générale de l'étudiant. Pour résumer, chaque étudiant possède au moins, un id, un nom et prénom, une année d'étude, une moyenne générale calculée à partir des 5 moyennes renseignées et une photo.

- **Les exigences particulières inhérentes au projet :**

Pour les exigences particulières, plusieurs points sont à respecter. Premièrement, le respect des informations des étudiants, notamment avec la photo dans le champ blob. Deuxièmement, La bdd devra contenir deux tables, l'une "étudiant" avec les informations suivantes(au moins):id, nom, prenom, année, moyenne, photo. L'autre "matière" avec les informations suivantes(au moins):id, matière 1, matière 2..., matière 5.. Troisièmement le résultat sera une application mobile utilisable sur android. Enfin, il sera possible de voir la liste d'étudiants à l'aide d'une liste déroulante.

3) Résultats attendus

Pour mener à bien ce projet il faudra utiliser plusieurs base de données SQL/MariaDB

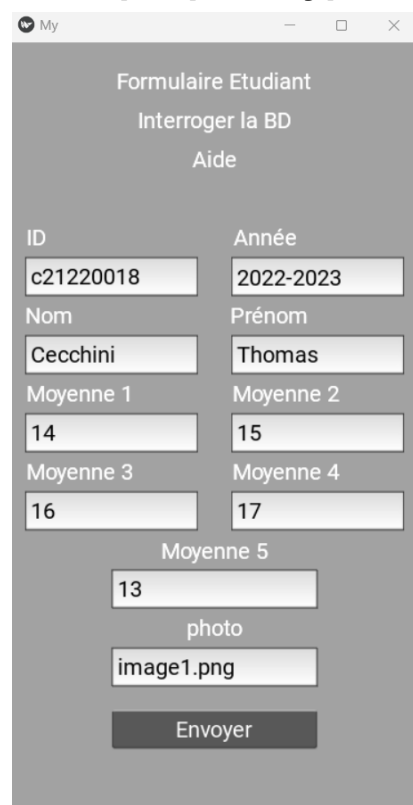
Ainsi que des interfaces python tels que Kivy ou encore Tkinter

la bibliothèque de stockage BLOB pour stocker les images dans la base de données

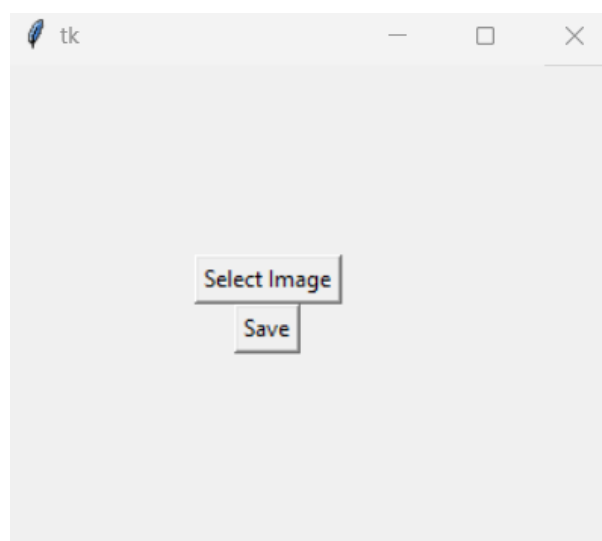
Ainsi que bulldozer pour réaliser l'application android

Ainsi que mysql.connector et/ou sqlite3 pour réaliser la connexion entre la BD et le programme python/Kivy

Exemple prototype de l'interface kivy et Tkinter



The screenshot shows a Kivy application window titled "My". The interface is titled "Formulaire Etudiant" and "Interroger la BD". It includes a link for "Aide". The form contains several input fields: "ID" (c21220018), "Année" (2022-2023), "Nom" (Cecchini), "Prénom" (Thomas), "Moyenne 1" (14), "Moyenne 2" (15), "Moyenne 3" (16), "Moyenne 4" (17), and "Moyenne 5" (13). There is a "photo" field with the value "image1.png" and an "Envoyer" button at the bottom.



Dans lequel on peut voir que l'application va récupérer les données suivantes :
numéro étudiant (ID) ;
Année d'études de l'étudiant ;
Nom de l'étudiant ;
Prénom de l'étudiant ;
Ainsi que les différentes notes dans chacune des moyennes et qui à l'intérieur du programme calcule par la suite la moyenne générale de l'étudiant et lui affichera par la suite
Ainsi que la photo de l'étudiant
Et toutes ces données seront transmises à l'intérieur de la BDD

Sur le plan économique et environnementale la solution n'est pas très coûteuses, elle ne demande qu'un poste de travail client ainsi qu'une machine virtuelle sur laquelle fonctionnera quelque base de données sql ainsi que un smartphone android
On estime le temps de travail du projet à environ 20 heures

Suivie de Projet :

Python :

Premier jet :

Tout d'abords nous avons essayer avec sqlite sans grand succès :

```
sql=( "INSERT INTO
etudiant( identifiant,annee,nom,prenom,moyenne1,moyenne2,moyenne3,moyenne4,moyenne5,photo)
VALUES(?,?,?,?,?,?,?,?,?,?)")
mydata=( identifiant,annee,nom,prenom,moyenne1,moyenne2,moyenne3,moyenne4,moyenne5,photo)
cur.execute(sql,mydata)
conn.commit()
conn.close()
```

N'étant pas encore bien avancé et étant encore pendant les vacances de Noël, nous avons décidé de reprendre directement lors de la rentrée, avec les VM en classe.

Deuxième jet :

Nous avons ensuite essayé avec la VM en se connectant avec python :

```
mydb = mysql.connector.connect(
    host="10.10.10.40",
    user="iutrt",
    password="Promo2022",
    database="etudiant"
)
```

Avec cette méthode, nous avons eu des difficultés à communiquer, notamment avec la configuration de la VM, la création des user avec les bon droits, etc...

Nous avons donc essayé un autre moyen qui s'avère être bien plus sécurisé que les méthodes précédentes.

Troisième jet :

Puis nous avons essayé avec request, une librairie http python, via pedaweb et myphpadmin :

```
url = "https://thomas-cecchini-etu.pedaweb.univ-amu.fr/extranet/base.php"

insert_etudiants = {
    "table": "etudiants", # La table qu'on veut affecté
    "id": identifiant,
    "nom": nom, # la clef et sa valeur
    "prenom": prenom, # la clef et sa valeur
    "annee": annee, # la clef et sa valeur
    "password": mdp # la clef et sa valeur
}

response_etudiants = requests.post(url, json=insert_etudiants) # requests.post() pour faire une
requêtes SQL INSERT INTO

respose_matiieres = requests.post(url, json=insert_matiieres) # requests.post() pour faire une requêtes
SQL INSERT INTO
```

Avec cette librairie et cette méthode, nous nous connectons à une base de données sur pedaweb, via http et cela nous permet en même temps qu'être bien plus sécurisé, de pouvoir travailler d'où on veut. Il y a cependant des désavantage à utiliser cela, notamment la programmation php que nous maîtrisons moins que celle de python. De plus les requêtes à utiliser avec python vers la base de données sont très différentes et de ce fait subjectivement plus compliquées étant donné que nous ne les avons pas vu en cours contrairement à la technique précédente. On peut également rencontrer plus de problèmes avec les images blob, et demande ainsi de manière générale plus de temps. En revanche cela vaut le coup comme je l'ai dit plus tôt pour l'aspect plus sécurisé étant donné de plus vu notre formation en cybersécurité.

Kivy :

Voici à quoi ressemblait nos premiers modèles de l'interface kivy, comme vous pouvez le voir nous avons plutôt réussi à reproduire notre modèle de base qui devait plus ou moins ressembler à cela :



On arrive après quelques améliorations au résultat suivant qui nous permet d'effectuer plus de choses, comme de cacher le mot de passe, ou encore le bouton vérifier qui nous permet de savoir si quand on rentre les informations du l'i et du mdp, que l'utilisateur existe.

My

fibre réseaux télécoms

Identifiant

Enter votre identifiant

Mot de passe

Enter votre mot de passe

Hide

Vérifier

Connexion

Retour

Nous sommes également passer de ce model :

Formulaire

Aide

Identifiant

Année

Nom

Prénom

Moyenne

Envoyer

A celui ci :

Choisit une options

ID	Année
<input type="text"/>	<input type="text"/>
Nom	Prénom
<input type="text"/>	<input type="text"/>
Anglais	Mathématiques
<input type="text"/>	<input type="text"/>
Informatique	Réseaux
<input type="text"/>	<input type="text"/>

Mot de passe

Choisir Image

Enregistrer

Retour

Cette interface nous permet d'entrer plus d'informations que la précédente et nous permet de choisir grâce à une liste déroulante, d'être un première année, deuxième année ou un prof.