TP Internet of Things

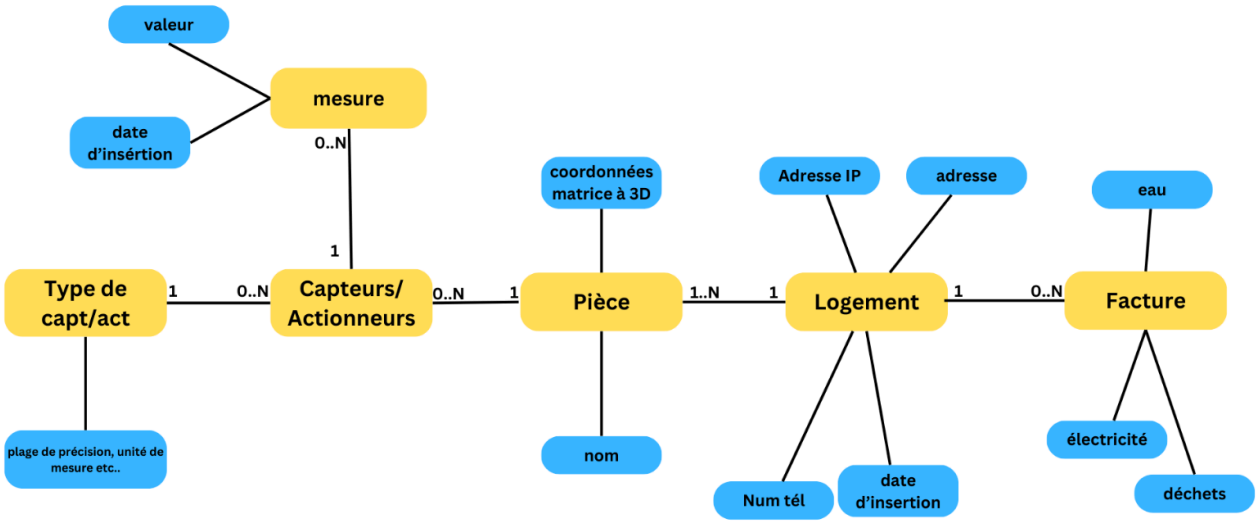
LOGEMENT ECO-RESPONSABLE

Ioana CACAU
TP A
EI4

Internet of Things

1 Base de données

Question 1: modèle relationnel de la base de données



Question 2: Suppression des tables

Pour cette question j'ai créé des ordres SQL pour supprimer toutes les tables existantes sont inclus en haut du fichier logement.sql.

Lignes 8-16 dans le code: Suppression des tables avec DROP TABLE IF EXISTS.

Question 3:Création des tables

Pour cette question j'ai créé des tables, définies dans logement.sql en utilisant la commande CREATE TABLE.
Lignes 20-26 : Table logement.
Lignes 28-36 : Table pièce.
Lignes 38-54 : Table TypeCapteur et Capteur.
Lignes 56-62 : Table mesure.
Lignes 64-73 : Table facture

Question 4 : Création d'un logement avec 4 pièces

J'ai créé un logement avec 4 pièces, définies dans logement.sql en utilisant la commande INSERT TO.
Lignes 92-99 : Insertion d'un logement.
Lignes 156-159 : Ajout des 4 pièces associées.

Question 5: Création des types de capteurs/actionneurs

J'ai ajouté 4 types de capteurs/actionneurs dans la table capteur_actionneur: **lignes 144-150** : (température, électricité, eau, luminosité) dans le fichier logement.sql.

Question 6:Création de capteurs/actionneurs

Pour cette question j'ai créé plusieurs capteurs/actionneurs dans le fichier logement.sql (commande : INSERT TO) : **lignes 167-175**

Question 7: Ajout de mesures

Mesures ajoutées pour chaque capteur/actionneur (commande : INSERT TO) **lignes 178-186**

Question 8: Création de factures

Quatre factures insérées, (commande : INSERT TO) **lignes 100-126** (eau, électricité, déchets..).

1.2 Remplissage de la base de données (remplissage.py)

Pour cette question, j'ai modifié le fichier replissage.py disponible sur moodle, qui ajoute des mesures et des factures supplémentaires dans la base de données. Pour cela, j'ai utilisé des commandes telles que execute et/ou executemany afin d'ajouter plusieurs factures/mesures en une seule opération.

2 Serveur RESTful

Exercice 1 : remplissage de la base de données

Pour cette partie, j'ai créé le fichier serveurREST.py. J'ai commencé par définir les classes pour structurer les données, notamment pour les factures, mesures, et capteurs. Ces classes permettent d'organiser les données échangées via l'API.

Ensuite, j'ai implémenté les requêtes GET et POST.

J'ai utilisé GET pour consulter les factures et les mesures. l**ignes 178-186**
J'ai ajouté un POST pour permettre l'ajout d'une mesure, un autre pour ajouter une facture et enfin un POST pour ajouter un capteur. l**ignes 178-186**

Exercice 2 : serveur web

Pour cet exercice, j'ai utilisé Google Charts afin de créer une page HTML affichant un camembert basé sur les valeurs des factures. Cette page est accessible via l'URL /factures/chart.

Pour y parvenir, j'ai utilisé les factures enregistrées dans ma base de données en exécutant une requête SQL avec cursor.execute. Ensuite, j'ai créé une liste contenant les données nécessaires à partir des résultats de la requête, que j'ai utilisée avec Google Charts pour afficher ces informations sous forme de camembert.

ligne 624 - 625 (@app.get("/factures/chart", response_class=HTMLResponse))

Exercice 3 : météo

Pour cet exercice, j'ai intégré une API REST de météo afin de récupérer les données nécessaires pour afficher à la fois la météo actuelle et les prévisions pour les cinq prochains jours.

Les données météo proviennent de l'API définie dans mon code par API_URL:

```
   "https://api.open-meteo.com/v1/forecast?"
"latitude=48.8534&longitude=2.3488&current_weather=true"
   "&hourly=temperature_2m,relative_humidity_2m,precipitation,wind_speed_10m"
"&daily=temperature_2m_max,wind_speed_10m_max&timezone=Europe/Paris"
)
```

ligne 869-875 (serveurREST.py)

J'ai extrait les informations nécessaires, comme la température, la vitesse du vent, et les précipitations, à l'aide de data.get(). Enfin, j'ai utilisé un code HTML intégré pour afficher ces données sur une page web, qui peut être trouvée à l'URL **/meteo**.

ligne 82 - 147 (@app.get("/meteo", response_class=HTMLResponse))
(serveurREST.py)

3 HTML/CSS/Javascript

Le site est composé de plusieurs fichiers situés dans le dossier WEBSITE : la

- page d'accueil (index.html), qui sert d'introduction et donne accès aux fonctionnalités principales via des liens,
- la page principale de configuration (home.html), qui regroupe des liens vers les quatre sections principales : consommation, état des capteurs, comparatif des économies, et configuration.
- navbar.html : configure la barre de navigation commune à toutes les pages du site.
- styles.css : définit le style général du site. (dans le dossier CSS)
- Dossier image : Contient les images utilisées sur le site.

Les pages consommation, état des capteurs, comparatif des économies, et configuration sont des pages que j'ai créées dans le fichier **serveurREST.py**. Ces pages sont générées à l'aide de f-strings intégrant les données récupérées via des requêtes GET.

- La page consommation, accessible à l'URL /consommation/chart, permet à l'utilisateur de visualiser la consommation du logement (électricité, eau, déchets, etc.) sous forme de graphiques interactifs. L'utilisateur peut choisir une adresse, une période (mois ou année), ainsi qu'un type de consommation (électricité, déchets, etc.), et les données sont affichées sous forme de graphiques.
- La page état des capteurs, accessible à l'URL /mesures/view, affiche des informations détaillées sur les capteurs installés dans le logement, comme le type de capteur, la valeur mesurée, la pièce associée, et la date de la mesure.
- La page comparatif des économies, accessible à l'URL /economies/comparison, présente un comparatif des économies réalisées en fonction des adresses et des

périodes. Les utilisateurs peuvent filtrer les données par adresse et période, et les informations sont affichées sous forme de graphique.

- La page configuration, accessible à l'URL /configuration, permet d'ajouter un capteur à une pièce spécifique. Un formulaire permet de sélectionner le type de capteur, la référence commerciale, le port de communication, et la pièce associée.