

**PROGRAMMATION MOBILE**

# **ANDROID**

**PROJET**

Domotique

[charles.meunier@u-bourgogne.fr](mailto:charles.meunier@u-bourgogne.fr)

# INFORMATIONS GENERALES

## OBJECTIFS

Réaliser une application Android complète à partir d'un cahier des charges.

## TRAVAIL A RENDRE

Le dossier de votre projet Android compressé dans une archive ZIP [groupe]-[nom]-[prenom].zip

## CONSIGNES

- Le projet peut être réalisé seul ou **maximum** à deux.
- Chaque réalisation sera originale et ne contiendra pas de section récupérée d'autres groupes. Si le cas se présentait, les projets concernés se verraient attribuer une note de 0/20.
- Des maquettes sont fournies à titre indicatif dans le sujet. N'hésitez pas à proposer vos propres interfaces en mettant l'accent sur l'ergonomie.

## CAHIER DES CHARGES

Votre projet consiste en la réalisation d'une application mobile de domotique permettant de contrôler des dispositifs et des capteurs présents dans votre maison.

## FONCTIONNALITES

Les fonctionnalités suivantes sont attendues :

### COMPTE UTILISATEUR

Afin de sécuriser l'application, il sera nécessaire de se connecter pour pouvoir avoir accès aux fonctionnalités domotiques de la maison.

- Créer un compte utilisateur
- Se connecter avec le compte utilisateur

### PIECES DE LA MAISON

Une maison est constituée de pièces que l'utilisateur peut ajouter ou supprimer.

- Ajouter une pièce
- Supprimer d'une pièce

### CAPTEURS

Une pièce peut contenir plusieurs capteurs. Chaque capteur est d'un type particulier (thermomètre, hydromètre, ...). Les capteurs retournent la dernière valeur mesurée ainsi que l'unité de mesure.

- Ajouter un capteur
- Supprimer un capteur
- Lire la dernière valeur mesurée par un capteur

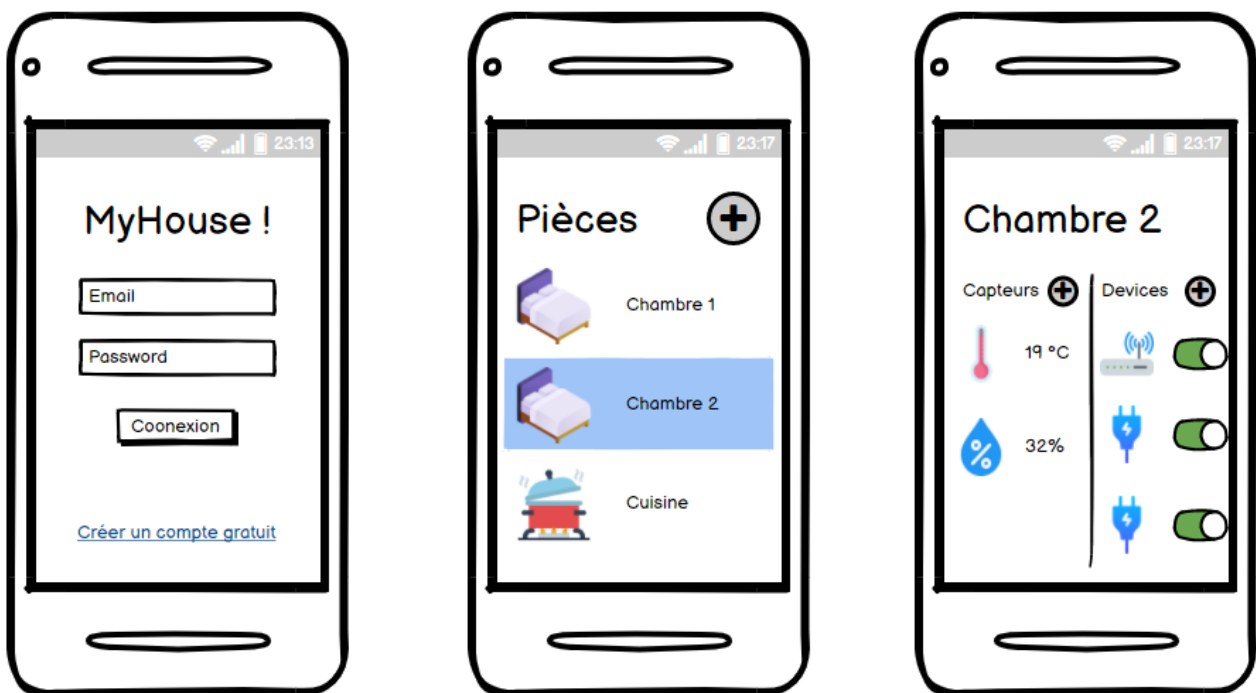
## PERIPHERIQUE

Une pièce peut également contenir des périphériques. Chaque périphérique est d'un type particulier (prise électrique, éclairage, internet, ...). Un périphérique peut être allumé ou éteint.

- Ajouter un périphérique
- Supprimer un périphérique
- Allumer un périphérique
- Eteindre un périphérique

## MAQUETTES

Les maquettes suivantes sont données à titre d'exemple pour décrire le comportement attendu de l'application.



## API REST

Pour accéder aux données de la maison, vous devrez appeler les API web ci-dessous.

### CREATION D'UN COMPTE UTILISATEUR

URL de l'API : <https://myhouse.lesmoulinsdudev.com/register>

Méthode de connexion : POST

Paramètres attendus :

- name : string
- login : string
- password : string

Code de réponse : 

- Code 200 si enregistrement OK
- Code 500 si erreur lors de l'enregistrement
- Code 400 si la requête est mauvaise

Valeur de retour : Aucune

## CONNEXION

URL de l'API : <https://myhouse.lesmoulinsdudev.com/auth>

Méthode de connexion : POST

Paramètres attendus : 

- login : string
- password : string

Code de réponse : 

- Code 200 = identifiants corrects
- Code 401 = le compte est inconnu

Valeur de retour (JSON) : { "token" : "token-utilisateur" }

## LISTE DES IMAGES DISPONIBLES

URL de l'API : <https://myhouse.lesmoulinsdudev.com/pictures>

Méthode de connexion : GET

Paramètres attendus : 

- type : "room" | "device" | "sensor"

Code de réponse : 

- Code 200 = requête correcte
- Code 400 = requête incorrecte

Valeur de retour (JSON) : {  
 "pictures" : [  
 { "id" : 1, "url" : "url-of-the-picture-1" },  
 { "id" : 2, "url" : "url-of-the-picture-2" },  
 ...  
 ]  
}

## LISTER LES PIECES

URL de l'API : <https://myhouse.lesmoulinsdudev.com/rooms>

Méthode de connexion : GET

Header : Authorization : Bearer {token}

Paramètres attendus : Aucun

Code de réponse : 

- Code 200 = requête correcte

- Code 400 = requête incorrecte

Valeur de retour (JSON)    {  
    "rooms" : [  
        { "id" : 1, "name" : "...", "picture" : "url" },  
        ...  
    ]  
}

## CREER UNE PIECE

URL de l'API : <https://myhouse.lesmoulinsdudev.com/room-create>

Méthode de connexion : POST

Header : Authorization : Bearer {token}

Paramètres attendus :

- name : string
- idPicture : int

Code de réponse :

- Code 200 = requête correcte
- Code 400 = requête incorrecte

Valeur de retour (JSON)    Aucune

## SUPPRIMER UNE PIECE

URL de l'API : <https://myhouse.lesmoulinsdudev.com/room-delete>

Méthode de connexion : POST

Header : Authorization : Bearer {token}

Paramètres attendus :

- idRoom : int

Code de réponse :

- Code 200 = requête correcte
- Code 400 = requête incorrecte

Valeur de retour (JSON)    Aucune

## LISTER LES TYPES DE CAPTEURS

URL de l'API : <https://myhouse.lesmoulinsdudev.com/sensor-types>

Méthode de connexion : GET

Paramètres attendus : Aucun

Code de réponse :

- Code 200 = requête correcte

Code 400 = requête incorrecte

Valeur de retour (JSON) {  
 "sensor-types" : [  
 { "id" : 1, "name" : "...", picture : "url" },  
 ...  
 ]  
}

## LISTER LES CAPTEURS

URL de l'API : <https://myhouse.lesmoulinsdudev.com/sensors>

Méthode de connexion : GET

Header : Authorization : Bearer {token}

Paramètres attendus :

- idRoom : int

Code de réponse :

- Code 200 = requête correcte
- Code 400 = requête incorrecte

Valeur de retour (JSON) {  
 "sensors" : [  
 { "id" : 1, "name" : "...", "type" : "...", picture : "url" },  
 ...  
 ]  
}

## CREER UN CAPTEUR

URL de l'API : <https://myhouse.lesmoulinsdudev.com/sensor-create>

Méthode de connexion : POST

Header : Authorization : Bearer {token}

Paramètres attendus :

- name : string
- idSensorType : int
- idRoom : int

Code de réponse :

- Code 200 = requête correcte
- Code 400 = requête incorrecte

Valeur de retour (JSON) Aucune

## SUPPRIMER UN CAPTEUR

URL de l'API : `https://myhouse.lesmoulinsdudev.com/sensor-delete`

Méthode de connexion : POST

Header : Authorization : Bearer {token}

Paramètres attendus :

- idSensor : int

Code de réponse :

- Code 200 = requête correcte
- Code 400 = requête incorrecte

Valeur de retour (JSON) Aucune

## RECUPERER LA DERNIERE VALEUR MESUREE PAR UN CAPTEUR

URL de l'API : `https://myhouse.lesmoulinsdudev.com/sensor-value`

Méthode de connexion : GET

Header : Authorization : Bearer {token}

Paramètres attendus :

- idSensor : int

Code de réponse :

- Code 200 = requête correcte
- Code 400 = requête incorrecte

Valeur de retour (JSON) `{ "value" : float, "unit" : "..." }`

## LISTER LES PERIPHERIQUES

URL de l'API : `https://myhouse.lesmoulinsdudev.com/devices`

Méthode de connexion : GET

Header : Authorization : Bearer {token}

Paramètres attendus :

- idRoom : int

Code de réponse :

- Code 200 = requête correcte
- Code 400 = requête incorrecte

Valeur de retour (JSON) 

```
{
  "devices" : [
    { "id" : 1, "name" : "...", "type" : "...", "picture" : "url", "status" : 1 },
    ...
  ]
}
```

## LISTER LES TYPES DE PERIPHERIQUES

URL de l'API : <https://myhouse.lesmoulinsdudev.com/device-types>

Méthode de connexion : GET

Paramètres attendus : Aucun

Code de réponse :

- Code 200 = requête correcte
- Code 400 = requête incorrecte

Valeur de retour (JSON)

```
{
  "device-types": [
    { "id": 1, "name": "...", picture: "url" },
    ...
  ]
}
```

## CREER UN PERIPHERIQUE

URL de l'API : <https://myhouse.lesmoulinsdudev.com/device-create>

Méthode de connexion : POST

Header : Authorization : Bearer {token}

Paramètres attendus :

- name : string
- idDeviceType : int
- idRoom : int

Code de réponse :

- Code 200 = requête correcte
- Code 400 = requête incorrecte

Valeur de retour (JSON) : Aucune

## SUPPRIMER UN CAPTEUR

URL de l'API : <https://myhouse.lesmoulinsdudev.com/device-delete>

Méthode de connexion : POST

Header : Authorization : Bearer {token}

Paramètres attendus :

- idDevice : int

Code de réponse :

- Code 200 = requête correcte
- Code 400 = requête incorrecte

Valeur de retour (JSON) : Aucune



## ALLUMER/ETEINDRE UN PERIPHERIQUE

URL de l'API : `https://myhouse.lesmoulinsdudev.com/device-status`

Méthode de connexion : `POST`

Header : `Authorization : Bearer {token}`

Paramètres attendus :

- `idDevice : int`
- `status : int (0 / 1)`

Code de réponse :

- `Code 200 = requête correcte`
- `Code 400 = requête incorrecte`

Valeur de retour (JSON) `Aucune`