



HOME IT

**RAPPORT
APPMOBILE**

2020 / 2021

**AZIZ RAYANE - BERNARD BENJAMIN
- HERAULT JEREMIE - HUANG CHEN
LONG - MOREAU LORIE**

204 - 207

SOMMAIRE

Application Mobile Objets Connectés

Présentation.....

..3

Arborescence.....

...4

Fonctionnalités.....

..6

Difficultés rencontrées et améliorations

possibles.....

...8

Conclusion.....

..9

PRESENTATION

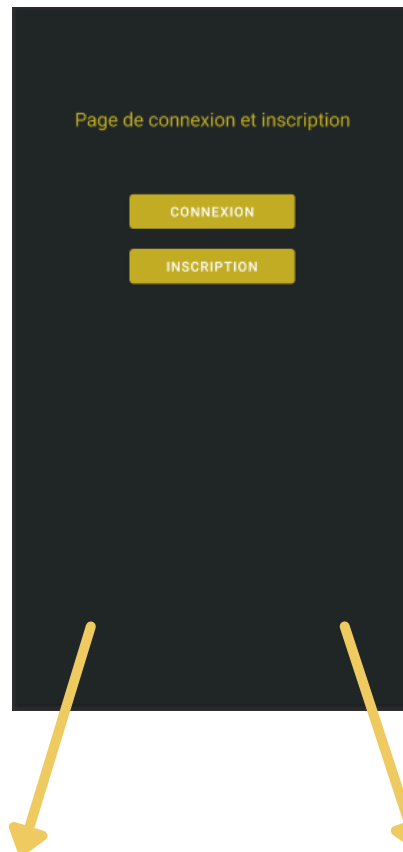
Application Mobile Objets Connectés

Dans le cadre du module d'Application Mobile, nous avons dû imaginer et réaliser notre propre application pour smartphone. Nous avons alors décidé de nous orienter vers le domaine "IoT Objets Connectés" et notre application sert ainsi à des utilisateurs à connecter des objets de leur maison (télévision, cafetière ...) pour les contrôler à distance simplement avec leur téléphone.


Nous avons développé notre application dans l'environnement Android Studio avec la partie Design en XML, les connexions entre les différentes activités (grâce aux boutons et images cliquables) et les méthodes pour agir sur les appareils (allumer/éteindre) en Java, et nous avons réalisé les simulations d'appareils grâce à une carte Arduino.



.ARBORESCENCE DE L'APPLICATION



Login



Login

[Register Here](#)

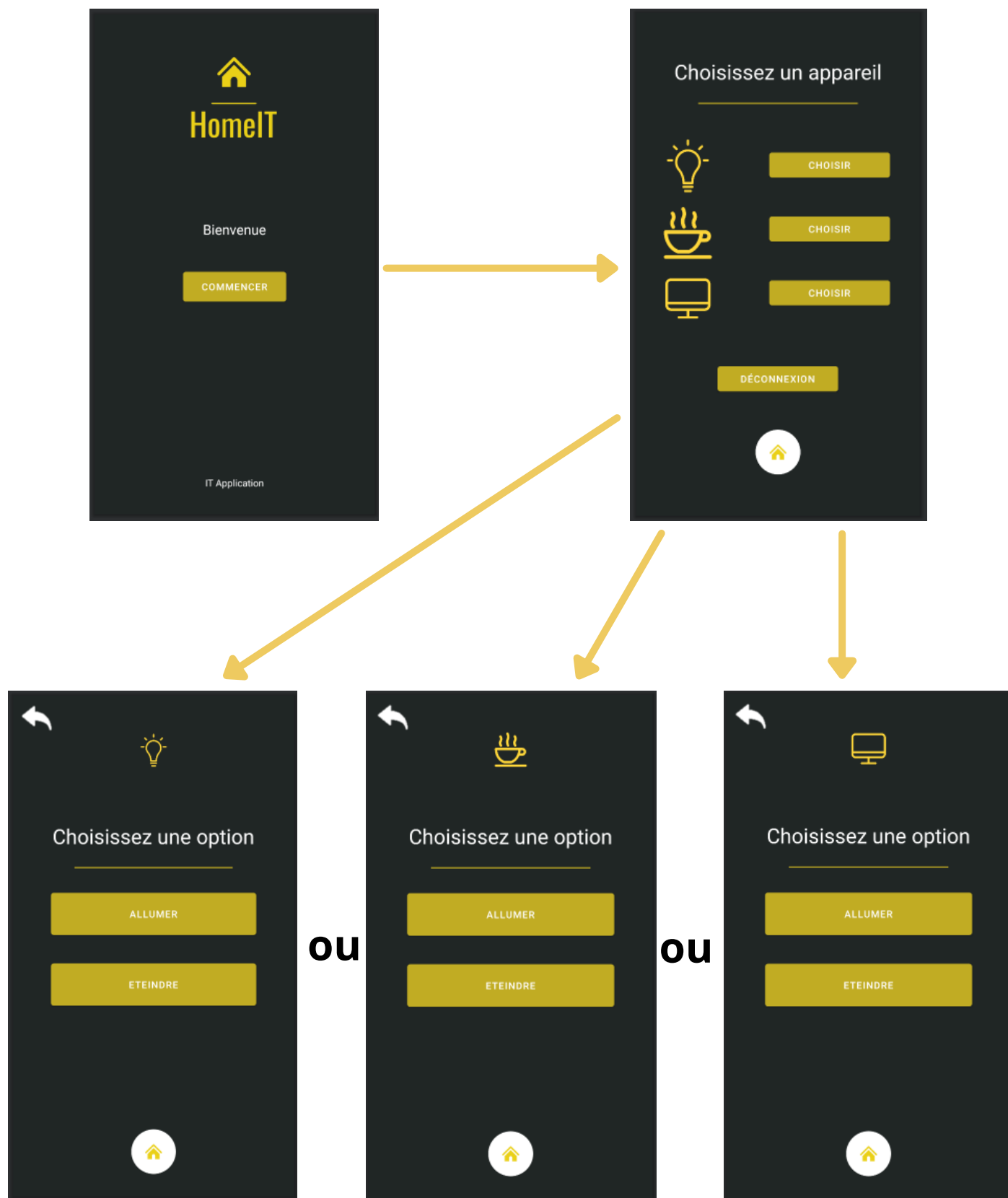
ou

SignUp

Register

[Already Registered ? Login Here](#)

.ARBORESCENCE DE L'APPLICATION



.FONCTIONNALITÉS DE L'APPLICATION

Pour réaliser notre application, nous avons dû gérer plusieurs aspects. Tout d'abord, l'utilisateur devait avoir la possibilité de s'inscrire (obtenir un compte utilisateur) pour gérer ses propres appareils électroniques. L'utilisateur devait ensuite pouvoir naviguer entre les différentes pages de l'application pour contrôler ses objets. Nous avons donc utilisé des "Intent" dans le code Java pour créer les liens entre les activités (sur des boutons, ou des images).

Exemple :



Renvoie à la page précédente



Renvoie à la page "Bienvenue"

Inscription / Connexion

Nous avons utilisé l'outil Google Firebase pour stocker les données des utilisateurs. Pour cela, nous avons connecté notre projet Android avec notre projet Firebase et nous avons ajouté les dépendances nécessaires pour utiliser les objets de Firebase. Ainsi, nous avons pu utiliser les méthodes `createUserWithEmailAndPassword()` et `signInWithEmailAndPassword()` pour créer l'inscription et la connexion. Par ailleurs, nous gérons les exceptions liés à ces deux services. Ainsi, si l'utilisateur essaie de s'inscrire alors que celui-ci l'est déjà, un message d'erreur sera renvoyé à l'écran.

Design, Visuel et Layouts

La partie visuelle de notre application a été codée en XML. Pour que nos pages restent 'fixes' pendant l'utilisation, nous avons utilisé deux layouts en particulier : le `RelativeLayout`, qui sert à afficher les éléments de la page les uns à la suite des autres, et le `LinearLayout` qui lui, affiche les éléments sur une même ligne en fonction de l'orientation qu'on lui attribue (vertical/horizontal).

Aussi, pour des raisons de clarté et de maintenabilité du code, nous avons stocké les couleurs et textes que nous souhaitons utiliser pour nos pages dans des fichiers 'Colors.XML' et 'Strings.XML'.

. FONCTIONNALITÉS DE L'APPLICATION

Colors.XML :

```
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFF</color>
    <color name="grey_dark">#202625</color>
    <color name="yellow">#C1AC23</color>
    <color name="dark">#11000000</color>
</resources>
```

Strings.XML :

```
<resources>
    <string name="app_name">AppDev</string>
    <string name="bienvenue_mess">Bienvenue</string>
    <string name="commencer_mess">Commencer</string>
    <string name="footer_mess">IT Application</string>
    <string name="choix_mess">Choisissez un appareil</string>
    <string name="clique_mess">Choisir</string>
    <string name="allumer_mess">Allumer</string>
    <string name="eteindre_mess">Eteindre</string>
    <string name="choix_option_mess">Choisissez une option</string>
    <string name="sign_up">Inscription</string>
    <string name="full_name">Nom Complet</string>
    <string name="email">Email</string>
    <string name="password">Password</string>
    <string name="phone">Téléphone</string>
    <string name="register">Enregistrer</string>
    <string name="already_register">Déjà Enregistré ? Connectez-vous ici</string>
    <string name="login">Connexion</string>
    <string name="deconnexion">Déconnexion</string>
    <string name="register_here">Inscription ici</string>
</resources>
```

Connexion, simulation Raspberry Pi et carte relais

Pour simuler le pilotage par l'application Android d'un équipement (lampe par exemple), nous avons créé un petit Web Service sur Raspberry Pi écrit en Python.

Il se base sur la librairie Flask pour le WS et sur la librairie gpiozero qui pilote les ports GPIO du Raspberry Pi.

Les ports GPIO 2 et 3 du Raspberry Pi ont ensuite été reliés à une carte avec des relais pouvant aller jusqu'à 220V.

Le Web Service pour piloter est très simple. Il prend en entrée :

- Une IP et un port
- Un numéro de device
- une action par exemple 'on' ou 'off'

En retour il renvoie juste un 'status' en JSON avec 1 si l'action a été exécutée ou 0 sinon.

Exemple : <http://192.168.0.189:5000/action/1/on>

Ce qui pilote la lampe 1 et qui l'allume. Pour l'éteindre :

<http://192.168.0.189:5000/action/1/off>

Le code du Web Service est fourni, une version simplifiée de ce code sans la partie GPIO a permis de travailler sans avoir le Raspberry Pi.

A noter également dans le code l'état ON/OFF est inversé au niveau de l'API appelée car la logique 0V/5V de la carte que nous avons est inversée.

Nous avons aussi testé le mode HTTPS, par contre la configuration des certificats et d'un serveur DNS était compliquée pour le travail d'équipe.

Nous avons donc préféré autoriser le HTTP en clair dans l'application car Android impose le HTTPS par défaut (modification des fichiers AndroidManifest.xml et Network_security_config.xml).

Les résultats sont concluants et l'application Android pilote bien les relais qu'on entend claquer et on voit les LED s'allumer.

.DIFFICULTÉS ET AMÉLIORATIONS

Améliorations

A l'avenir, cette application pourrait permettre à l'utilisateur de manipuler plusieurs objets connectés comme une cafetière voir même une télévision. Nous pourrions également rendre l'expérience utilisateur encore plus agréable.

Difficultés

Au cours du développement de cette application, nous avons rencontrés des difficultés. Tout d'abord, nous devions prendre en main l'environnement de développement qui était tout nouveau pour nous. Par ailleurs, en termes de technicité nous avons dû nous approprier de nouveaux outils. Par exemple, l'utilisation de Firebase pour la gestion de notre base de données.

Néanmoins la partie la plus complexe à réaliser était celle concernant le service principal que nous proposons, c'est-à-dire la gestion d'une lampe via l'application. Nous avons dû manipuler de l'électronique et la connexion entre la lampe et l'application était complexe à réaliser.

CONCLUSION

Application Mobile Objets Connectés

Au final, nous sommes plutôt satisfait de notre réalisation. C'est la première fois que nous développons une application pour mobile et ce fut une très bonne première expérience. Nous avons réussi à travailler sur tous les aspects de l'application, de la base de données au design, et jusqu'à la simulation d'objets connectés. Nous sommes parvenue à nous organiser et répartir les tâches comme nous le voulions.

Ainsi, ce projet a pu nous donner une nouvelle fois l'occasion de travailler en groupe et sur des nouvelles notions, et donc d'en tirer tous les bénéfices que ce genre de travail propose.