Domotique Raspberry

Explication brève

Domotiser des anciennes ou des nouvelles infrastructures par la création de « kits » prémontés, où différents modules pourront être choisis, en fonction des besoins des utilisateurs.

Fonction :

- contrôle de la lumière

- contrôle de l'eau

- contrôle de la présence

- contrôle ménager (Relay)

- contrôle vocal

- gestion d'agenda

- contrôle par application (web ou Android)

- vidéosurveillance

- table connectée

- miniaturisation

Contrôle de lumière

Il existe des lampes demandant une ampoule spécifique ; il sera donc nécessaire de gérer l'allumage par le courant.

Contrôle de l'eau

Contrôle de l'ouverture de tuyaux avec des électrovannes, ...

Contrôle Vocal

Pouvoir effectuer des actions en fonction de la demande, que ce soit sur un module annexe ou sur l'application tiers.

Contrôle de la présence

Pouvoir effectuer des actions telles que, allumer la lumière, allumer la caméra

ou dire bonjour ; si un mouvement est détecté dans la pièce.

Contrôle par application

Pouvoir effectuer des actions sur les modules secondaires grâce à une application connectée soit en Bluetooth ou via le web.

Contrôle ménager

Contrôler les appareils du quotidien, comme allumer la machine à café à l'heure que l'on aura prévu, relever des volets électriques, …

Logiciel et langage :

- linux (Raspbian)

- python

- android studio

- html, CSS, JavaScript

- sublime text

- visual studio

- putty

- VMware

- SketchUp

Matériel

**Module primaire** *composé de* : Raspberry pi 3 B, écran, abaisseur de tension 220 V to 5 V

**Modules secondaires** *composé de* : Raspberry pi zéro w, abaisseur de tension 220 V to 5 V / 12V,

*Au choix* : Relay, capteur infrarouge, haut-parleur, micro, boutons, caméra, Dimmer, …

Le « kit » comprend un module primaire pour plusieurs modules secondaires connecté à celui-ci

Définir les différentes contraintes à respecter

Bien définir les différentes limites

**Contrôle de la lumière**

**Page web**

Le contrôle se fera par le biais d’une page web. Ainsi le contrôle sera disponible par tout appareil pouvant utiliser un navigateur web.

Cette application laissera deux choix de contrôles :

* Par la voix
* Par des boutons

On ne veut pas que n’importe qui utilise la page web et éteint les lumières. Avant de pouvoirs utiliser les options de contrôles, l’utilisateur devra d’abord se connecter.

On devra alors avoir :

* Une page pour la connexion
* Une page listant les lampes disponibles.
* Une page listant les utilisateurs.

Seulement un administrateur (client) sera capable de consulter la liste des utilisateurs et de modifier les données et aura le contrôle de toutes les lampes.  
Les utilisateurs auront accès que à leurs données et les lampes auquel ils auront la permission de contrôle.

Afin d’avoir une idée de l’emplacement des différents composants de l’application. Une maquette pour chaque page a été réaliser.

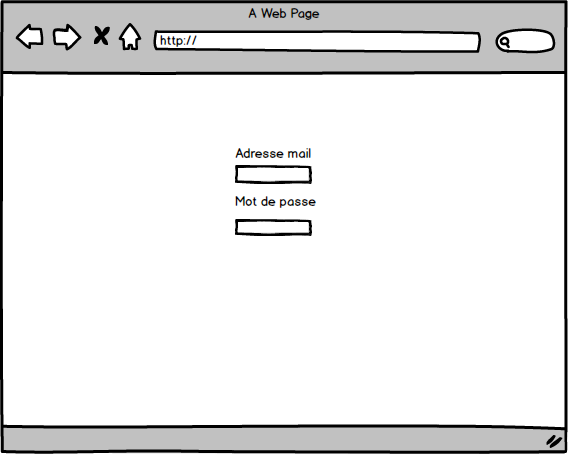
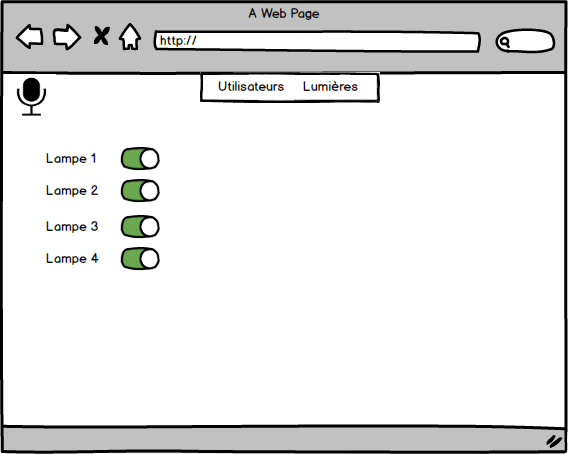


Figure 1 : Page connexion. Figure 2 : Page Lumières

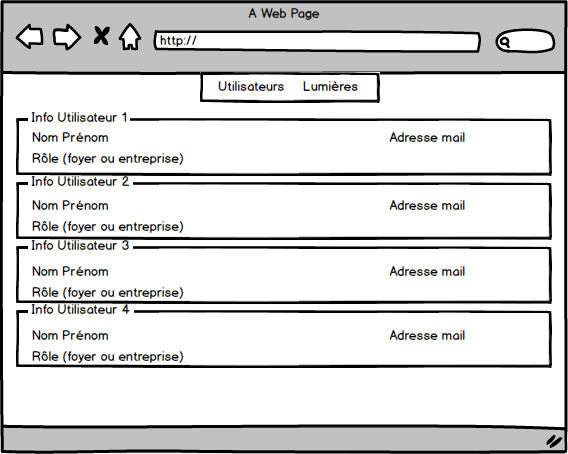


Figure 3 : Page Utilisateurs

**Hardware**

Pour la partie matérielle du contrôle de la lumière, le module contiendra trois éléments de base :

* Un abaisseur de tension HLK-PM01
* Un Raspberry pi zéro W
* AC Light Dimmer Module, 1 Channel, 3.3V/5V logic, AC 50/60hz, 220V/110V

Le module sera donc placé à la place d’un interrupteur dans le blochet. Pour l’alimentation du Raspberry, il faudra brancher la phase et le neutre sur l’abaisseur, pour que celui-ci transforme le 220V AC en 5V DC qui me permettra donc d’alimenter mon Raspberry.

*(ATTENTION : Pas toujours le neutre dans le blochet donc il faut trouver une solution alternative.)*

Pour allumer la lumière, le Dimmer sera relié au Raspberry sur ses pins de commande. De l’autre côté, il sera relié à la phase et le fil de commande de la lampe. Le Dimmer permettra un contrôle progressif de l’intensité lumineuse des ampoules, ce que nous utiliserons pour faire des fonctions et des modes pour les utilisateurs.

Nb : Pour des modules n’ayant pas besoin d’avoir accès aux fils de commande des lampes, il sera préférable de choisir une prise comme emplacement, car le neutre s’y trouvera toujours.

**Généraux**

**Serveur Linux**

Le Raspberry pi jouera le rôle du serveur. Il devra fournir différents services.

* Web : permettant d’héberger la page web.
* DNS : pour permettre que le site web soit accessible autrement qu’en notant l’adresse IP.
* BDD : qui contiendra toute les données (les utilisateurs et les lampes).
* DHCP : qui sera configuré pour donner dynamiquement les informations de connexions aux modules.

Le serveur devra également être sécurisé, avec par exemple : un antivirus, SE Linux et la configuration du firewall

**La base de données**

Une base de données sur un serveur (Raspberry Pi) sera utilisée contenant 2 tables.

Une table utilisateur avec les colonnes :

* Nom
* Prénom
* Mot de passe
* Type d’utilisateur : admin, compagne, invité ou autre …

Une table de la base données représentera les lumières avec les colonnes :

* Emplacement
* Permission

On voudrait adapter la mise en place de notre produit partout que ce soit dans une entreprise ou un foyer.

Une entreprise possède généralement au moins un informaticien. Or un informaticien pourrait vouloir créer ses propres scripts pour contrôler nos produits. Tous les scripts contrôlant nos produits devront alors donner la possibilité d’être utilisé en ligne de commande.

Script

Nous aurons un script gérant la lumière.

Script lumière

* Ajout lampe
* Suppression lampe
* Allumage lampe
* Extinction lampe

**DNS**

Le DNS sera principalement utilisé pour permettre l’accès au site web autrement que par l’adresse IP.

Il sera également utilisé pour associer l’adresse IP d’un module à un nom.

Exemples :

* 10.0.0.10 = lampe salon
* 10.0.0.11 = lampe cuisine
* 10.0.0.12 = lampe salle de bain
* 10.0.0.13 = lampe salle à manger

Ainsi chaque module pourra être accessible par son nom et utilisable sans devoir changer une adresse IP dans le code.