Cahier des charges fonctionnel

Entreprise : EOS Development
Produit : Super Elevator v1.0
Groupe : Alexandre Rousseau - Fabien Billard - Julie Nginn - Safia Gobet - Wilfried Atride

Table des matières

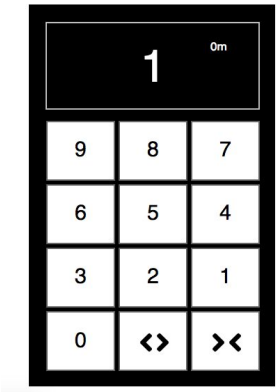
1. Présentation du projet

Pour notre projet nous avons choisi comme objet connecté de faire un ascenseur. Cet objet sert à monter et descendre des étages :3

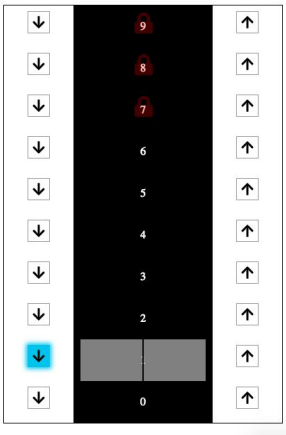
Le déplacement de l'ascenseur est commandé à distance sur l'interface web en React. Elle permet de contrôler à distance la montée et la descente des étages ainsi que l'ouverture et la fermeture des portes.

2. L'interface web

L'ouverture et la fermeture des portes ainsi que le déplacement de l'ascenseur sont commandés à distance sur l'interface web en React. Les étages peuvent être directement choisis sur la représentation du boîtier de commande.



L'interface représente également l'état actuel de l'ascenseur. L'état est directement envoyé par le serveur via une socket.



Certains étages de notre ascenseur ne sont accessibles que par badge. Ils sont représentés par un cadenas sur les étages.

3. Le serveur - API Express JS

Le backend se compose d'une API Express JS et d'une entité "Elevator" qui se comporte comme si elle était indépendante du serveur : les communications pour signaler une mise à jour de son état au serveur se fait via des sockets.
Tous les scripts de commande du matériel se trouvent dans src/components et sont appelés par src/models/elevator.js.

L'état de l'ascenseur est persisté dans le cache pour ne pas le perdre si le serveur redémarre.

Une API en Express JS permet de recevoir des instructions provenant du front-end react, puis de commander l'ascenseur.

L'algorithme de l'ascenseur a été optimisé pour organiser au mieux les priorités. Par exemple, si une personne au 7ème étage demande à descendre au 3ème et qu'entre temps une demande de descente est faite au 5ème, l'ascenseur s'arrêtera au 5ème, puis au 3ème.

A chaque fois que le serveur reçoit une mise à jour de l'ascenseur, il transmet le nouvel état au front.

4. La maquette

Pour la réalisation de ce projet nous avions comme matériaux :

- Un Raspberry PI 3 B + pour héberger un serveur NodeJs et un serveur Apache pour servir l'interface de commande.
- Un écran LCD affichant l'état de l'ascenseur
- Un Servo moteur pour simuler la montée et la descente de l'ascenseur
- Un lecteur RFID pour la lecture des badges
- 2 LED pour signaler les autorisations d'accès à l'étage
- Un sonar détectant l'ouverture et la fermeture de la porte

Pour la conception de notre ascenseur nous avons utilisé une boite en carton vide qui représente la cage d'ascenseur. La cabine de l'ascenseur est faite en mousse et elle est soutenue par deux fils qui permettent la montée et la descente de l'ascenseur. Ces deux fils sont soutenus par un crayon qui est lui-même relié à un moteur (Servomoteur) qui permet la rotation du cylindre et donc la montée ou la descente de la cabine.
Nous avons également équipé l'ascenseur de deux LED. Une rouge et une verte. Ces LED sont utilisées pour les étage sécurisés 7, 8, et 9. Lorsque l'on arrive à l'un de ces 3 étages sécurisé la lumière rouge clignote signifiant qu'il faut passer le badge devant le lecteur RFID. Au passage du badge la lumière devient verte et l'accès à l'étage est autorisé, ce qui provoque le mouvement de l'ascenseur.
Enfin, pour la lecture de l'étage nous avons ajouté un écran LCD qui va permettre d'afficher l'étage auquel on se trouve.