



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 de l'entreprise
2. Les objectifs du produit
3. Les logiciels concurrents
4. Développement du logiciel
5. Les spécificités et les livrables
 - 5.1. Les contraintes techniques
 - 5.2. Les livrables
 - 5.3. Le planning
6. Chiffrage du projet

1. Présentation de l'entreprise

L'entreprise EOS a été créée en décembre 2018 afin de développer un logiciel de contrôle à distance des ascenseurs.

Les principaux collaborateurs dans l'entreprise sont les suivants :

- Alexandre Rousseau : Développeur back-end NodeJS
- Fabien Billard : Lead développeur back-end
- Julie Nginn : Lead développeuse front-end ReactJS
- Safia Gobet : Développeuse front-end ReactJS
- Wilfried Atride : Développeur front-end

2. Les objectifs du produit

Le produit Super Elevator proposé par l'entreprise est un système innovant de contrôle à distance des ascenseurs. Il permet d'optimiser les déplacements de l'ascenseur et de le contrôler à distance, via une application. Il est principalement destiné aux immeubles nécessitant un contrôle des accès.

3. Les logiciels concurrents

Les algorithmes des ascenseurs actuels ne sont pas tous optimisés et la plupart ne permettent pas d'intégrer une commande à distance pour permettre un contrôle par le personnel assurant la sécurité des immeubles.

4. Développement du logiciel

Le logiciel développé permet de contrôler les ascenseurs à distance sur un réseau interne filaire et isolé.

Le projet se divise en deux parties : une interface permettant d'avoir un aperçu de l'état de l'ascenseur et un back-end permettant de piloter l'ascenseur.

L'ascenseur sera simulé par un Raspberry Pi 3b+ permettant de servir le code nodeJS et les composants suivant :

- Un écran LCD indiquant l'état actuel de l'ascenseur
- Un servo-moteur simulant le passage aux étages suivants
OU
- Un stepper-moteur actionnant une maquette
- Un capteur sonar HC-SR04 permettant de simuler le capteur de porte
- Un lecteur RFID permettant de scanner les badges
- Des led montrant les autorisations d'accès

5. Les spécificités et les livrables

5.1. Les contraintes techniques

5.1.1. Le back-end en NodeJS

- Le back-end doit être capable de piloter l'ascenseur :
 - monter et descendre des étages en actionnant un moteur
 - ouvrir et fermer la porte de l'ascenseur en détectant les obstacles
 - contrôler les accès à certains étages avec un badge
 - signaler l'état de l'ascenseur via les leds et l'écran LCD
- Le back-end doit proposer une API permettant de commander l'ascenseur
- L'état de l'ascenseur doit être persisté en cas de coupure du serveur

5.1.2. Le front-end en React

- Le front-end devra permettre de donner des instructions à l'ascenseur via des appels à une API NodeJS Express.
- Le front-end devra recevoir en temps réel l'état de l'ascenseur via des sockets Socket.io
- Le front-end devra représenter le plus fidèlement possible l'état actuel de l'ascenseur même en cas d'actualisation de la page

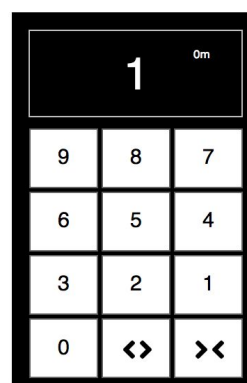
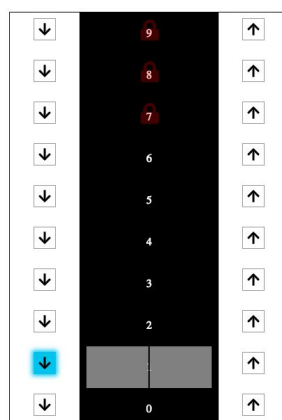
5.2. Les livrables

- Un front-end en React communiquant avec le back-end via une API et des sockets
- Un back-end respectant les contraintes décrites ci-dessus
- Une simulation de l'ascenseur avec le Raspberry Pi 3b+

La maquette du front-end est la suivante :



Super Elevator



5.3. Le planning

Le projet sera dans un délai de 3 semaines. Une première version sera livré le 17 janvier 2019.

La première semaine sera consacrée aux tests du matériel afin de confirmer qu'il peut être commandé avec NodeJS. Une interface pour l'API devra être également définie entre le front-end et le back-end (format des données échangées et adresse à appeler).

La seconde semaine servira à développer en parallèle le front-end et le back-end en ajoutant progressivement les fonctionnalités requises.

La dernière semaine permettra de tester la communication entre les deux parties et de corriger les derniers problèmes.

6. Chiffrage du projet

Facturation des heures			
Rôle	Quantité	Coût journalier (€/jour)	Coût total (€)
<i>Lead developer Front</i>	15	410	6 150
<i>Lead developer Back</i>	15	410	6 150
<i>Développeur Front</i>	15	360	5 400
<i>Développeur Back</i>	15	360	5 400
<i>Développeur Full Stack</i>	15	390	5 850
Total Facturation des heures			28 950 €
Matériel			
<i>Raspberry pi 3b+</i>			40
<i>Freenove Ultimate Starter Kit</i>			46
Total Matériel			86
TOTAL			29 036€