

# Projet Ascenseur - Cahier des charges



**Equipe 23** : Régis Luu-Vu / Yann Miguel / Jonathan Senes / Adrien Tokoglu / Alexandre Wery

**Git** : [https://github.com/Zepheniah/Projet\\_Ascenseur\\_GL](https://github.com/Zepheniah/Projet_Ascenseur_GL)

**Version 5**



## Sommaire

### Table des matières

Sommaire .....	3
Gestionnaire de versions.....	4
1) Données générales .....	5
a) Exposé du problème .....	5
b) Objectifs.....	5
c) Responsabilités des MOA (maîtrise de l'ouvrage) et MOE (maîtrise de l'œuvre) .....	5
d) Critères d'acceptabilité .....	5
e) Contraintes d'environnement .....	6
2) Données techniques .....	7
a) Description produit .....	7
b) Processus de développement .....	7
c) Fonctions à satisfaire.....	8
d) Evolution en cours de réalisation .....	8
e) Extensions envisagées .....	9
3) Données économiques.....	10
a) Délais : dates de début de fin du projet .....	10
b) Coût en de développement (effort) .....	10
c) Coût financement : le budget nécessaire à la réalisation du projet, comment le projet sera financé ? .....	10
d) Moyens ressources : deux types de ressources : humaines (l'équipe de développement, etc.) et matérielles (ordinateurs, intranet, etc.).....	10
4) Données commerciales .....	11
a) Qualité de vente et attrait du produit.....	11
b) Qualité d'utilisation : essentiellement l'utilisation du produit par le client .....	11
c) Phase de transfert : lieu et date de la remise du produit .....	11

## Gestionnaire de versions

Date	Tâches effectuées	Numéro de version
21/09/2020	Création du document	V1
28/09/2020	Refonte complète du CDC en accord avec les conseils reçus	V2
6/10/2020	Mise à jour des objectifs et critères d'acceptabilité.	V3
8/10/2020	Mise à jour des évolutions en cours de réalisation	V4
9/10/2020	Structuration des fonctions à satisfaire Mise à jour finale du CDC	V5

## 1) Données générales

### a) Exposé du problème

Notre équipe est engagée dans le développement du contrôle-commande d'un ascenseur pour le compte d'une entreprise d'installations d'ascenseurs. C'est-à-dire que nous devons mettre au point un logiciel censé piloter l'ascenseur en fonction de requêtes qu'il recevra de la part des utilisateurs. Il les transformera ensuite en requêtes pour la partie opérative. Le fonctionnement basique d'un ascenseur étant connu de toute l'équipe, nous pensons qu'il n'est pas nécessaire d'en faire une description détaillée.

### b) Objectifs

Développer un logiciel de contrôle commande pour ascenseur en appliquant des méthodes de travail rigoureuses est le principal objectif. De ce fait, si son implémentation est naturellement importante, les procédés de réflexions tout comme les apports de documentation le sont tout autant. Ainsi, la réalisation de diagrammes ou de livrables intermédiaires est attendue en vue de structurer le plus nettement possible ce projet. Avant d'aller plus loin, il est important de rappeler que seuls le contrôle-commande et GUI de test nous sont demandés. Pour simuler le fonctionnement d'un ascenseur en lieu et place d'une cabine et d'une interface, nous utiliserons une GUI.

### c) Responsabilités des MOA (maîtrise de l'ouvrage) et MOE (maîtrise de l'œuvre)

La maîtrise d'ouvrage comme la maîtrise d'œuvre est attribuée à l'équipe de développement.

### d) Critères d'acceptabilité

Le contrôle commande doit pouvoir gérer efficacement les requêtes à envoyer à la partie opérative et celles reçues par les demandes utilisateurs. Le langage de programmation demandé pour ce faire est Java. De la documentation bien rédigée, des commentaires ainsi que des tests rigoureux sont également attendus afin de témoigner de la qualité du produit. Enfin, les délais de rendus ne doivent pas être dépassés.

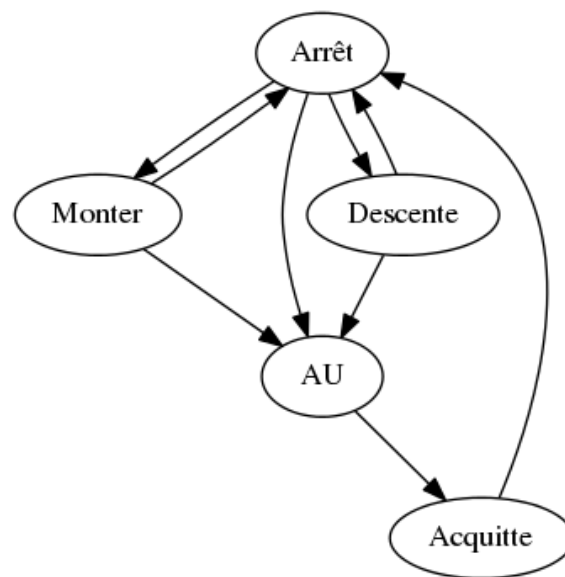
### e) Contraintes d'environnement

Le but principal de notre algorithme est qu'il puisse simuler efficacement les déplacements d'un ascenseur grâce aux requêtes qu'il récupère. On ne peut pas demander au contrôle-commande des requêtes qu'il ne peut pas effectuer. Par exemple, se rendre à un étage au-delà du dernier étage ou se rendre à un étage inférieur au RDC (en supposant qu'il n'y ait pas d'étages négatifs) ne doivent pas être possibles. La présence d'un bouton d'arrêt d'urgence est à prévoir afin que les utilisateurs puissent signaler tout problème dans la cabine.

## 2) Données techniques

### a) Description produit

Comme dit tout à l'heure, le produit à réaliser est un contrôle-commande avec une GUI, visant à simuler un ascenseur. Ce dernier est soit en train de monter, soit en train de descendre, soit à l'arrêt, soit en arrêt d'urgence (AU). Si l'ascenseur entre dans l'état AU, il faudra alors qu'un technicien intervienne pour le débloquent. Pour spécifier les interactions entre les différents états, nous avons utilisé un graphe détaillant les transitions possibles entre chaque état de l'ascenseur.



**Titre** : Interaction entre les différents états d'un ascenseur.

L'arrêt d'urgence peut se faire à tout moment grâce à un bouton dans la cabine, à un étage comme entre deux étages. Un bouton à chaque étage permet à l'utilisateur d'appeler la cabine d'ascenseur. Enfin, pour sélectionner le numéro de l'étage où se rendre, l'utilisateur utilise, un pavé numérique que l'on a interprété par un input number dans la GUI. Lorsqu'une requête est envoyée et que l'ascenseur n'a rien à faire, l'ascenseur la traitera jusqu'à satisfaction. Sinon, les requêtes qu'il reçoit pendant qu'il est en mouvement sont enregistrées, puis elles seront traitées de telle sorte à minimiser le déplacement de la cabine.

### b) Processus de développement

Le développement a commencé en même temps que l'implémentation de la GUI. La structure de notre contrôle-commande et de notre partie opérative a également été mise au point, puis implémenté. Des tests ont en parallèles été ajoutés pour permettre de valider leur fonctionnement. Un enrichissement des tests peut encore être attendu afin de mettre davantage en avant l'efficacité de notre algorithme

ainsi que notre rigueur de travail. Compte tenu de la charge de travail actuelle, nous n'avons pas utilisé de processus de développement particuliers tel que le modèle en V ou le modèle en spirale. En effet, plusieurs étapes de développements ont pu se chevaucher durant notre processus de développement, mais nous assurons une rigueur de travail.

### c) Fonctions à satisfaire

Dans notre implémentation, nous avons décidé d'implémenter une donnée "Direction" de l'ascenseur pouvant uniquement admettre 3 états :

Direction	
UP	L'ascenseur est en train de monter.
DOWN	L'ascenseur est en train de descendre.
NONE	L'ascenseur ne bouge pas.

Si l'ascenseur a simulé l'état d'urgence, sa Direction est NONE, mais conformément au graphe que nous avons établi, seule un acquittement pourra le débloquent.

En ce qui concerne la contrôle-commande, les fonctions suivantes sont principalement attendues :

Contrôle-commande	
Fonction	Description brève
acquit()	Débloque l'ascenseur.
reachFloor(etaage1, etage2)	Effectue le déplacement requis entre l'étage1 et l'étage2.
emergencyBreak()	Simule l'arrêt d'urgence et bloque l'ascenseur.
addFloorRequest()	Ajoute une requête utilisateur dans une file de requêtes.
evaluateCommand()	Traite la première commande de la file de requêtes.
setDirection()	Met à jour la direction puis envoie l'indication à la partie opérative.
getFloor()	Retourne le numéro étage actuel.

Dans la partie opérative, nous nous contentons d'établir les différentes mises à jour de notre GUI et de renvoyer quelques données particulières.

### d) Evolution en cours de réalisation

La GUI a été premièrement implémentée. Même si nous la trouvons intéressante pour le moment, elle peut être sujette à des changements, le choix entre l'input number et la grille de boutons restant encore à définir. Comme déjà dit précédemment, la partie opérative et le contrôle-commande ont été mise au point. En plus de cela, nous avons implémenté deux « algorithmes de tris » pour déterminer le meilleur traitement de requêtes. Le premier est une approche naïve d'algorithme de tri très simple. L'autre, plus complexe, privilégiera toujours les requêtes indiquant le souhait de se rendre au RDC. La correction de certains bugs est également en cours et nous demandera plus de temps.



#### e) Extensions envisagées

Nous souhaitons implémenter davantage d’algorithmes et plusieurs sont actuellement à l’étude. Par exemple, nous avons envisagé de gérer de cet ensemble de requêtes avec un algorithme qui privilégie les requêtes à destination du dernier étage tout en minimisant les déplacements de l’ascenseur. Nous pouvons éventuellement songer à des fonctionnalités traitant de manière différentes les requêtes envoyées à partir du RDC et/ou du dernier étages. Toutes ces extensions ne sont pas prioritaires mais peuvent être pourquoi pas, envisagées dans le futur.

### 3) Données économiques

#### a) Délais : dates de début de fin du projet

Le projet a officiellement commencé le 21 septembre et devrait se terminer le 9 octobre. Cette date pourra changer en fonction des observations des MOE/MOA ou des besoins du client.

#### b) Coût en de développement (effort)

Jusqu'à cinq personnes travaillent sur ce développement d'applications. Un diagramme de Gantt est disponible dans le livrable, mais les répartitions de tâches ont pu varier ou se faire à plusieurs au vu de la situation inédite que nous subissons tous.

#### c) Coût financement : le budget nécessaire à la réalisation du projet, comment le projet sera financé ?

Aucun financement n'est prévu pour ce projet... A moins bien sûr que de généreux donateurs n'aient envisagé le contraire. Il sert avant tout à former l'équipe de développement aux outils de gestion de projets aux rôles de MOA et MOE.

#### d) Moyens ressources : deux types de ressources : humaines (l'équipe de développement, etc.) et matérielles (ordinateurs, intranet, etc.)

Nous sommes une équipe de 5 étudiants et nous disposons chacun d'un ordinateur personnel. A moins que l'on nous demande de changer, nous nous servirons de Github pour la gestion de version ainsi que de Discord pour communiquer.

## 4) Données commerciales

### a) Qualité de vente et attrait du produit

Le produit que nous livrons est censé rendre l'utilisation de l'ascenseur plus agréable. Doté d'un algorithme efficace, les temps de calcul sont plus faibles, optimisés et la partie mécanique sera moins utilisée. De ce fait, les maintenances seront moindres et la durée de vie de l'ascenseur sera nettement meilleure.

### b) Qualité d'utilisation : essentiellement l'utilisation du produit par le client

Si l'algorithme pilotant l'ascenseur est rapide, les utilisateurs n'auront pas à attendre trop longtemps, ce qui ne peut que leur être bénéfique. La consommation énergétique de l'ascenseur n'en sera alors que plus faible par rapport à ces concurrents.

### c) Phase de transfert : lieu et date de la remise du produit

La date de rendu est le vendredi 9 octobre. La date de rendu est le vendredi 9 octobre. Un cahier des charges, un diagramme Gantt et un document de spécifications est attendu.