

# DOMODOOR

## Dossier de conception générale

LOISON Laureen – VIREY Maxime – GUERON Antoine  
UNIVERSITÉ DE NANTES  
MASTER MIAE ALTERNANCE

DEVELOPPEMENT LOGICIEL: APPROCHE FORMELLE ET A OBJET

<b>ÉQUIPE</b>	<b>3</b>
<b>INTRODUCTION</b>	<b>3</b>
<b>I - CONCEPTION GENERALE</b>	<b>3</b>
<b>A- MODELISATION DE L'ARCHITECTURE DU SYSTEME</b>	<b>3</b>
1) ARCHITECTURE EN COUCHES	3
2) DIAGRAMME DE COMPOSANTS ET DE DEPLOIEMENT	4
<b>B- MODELISATION DU COMPORTEMENT DU SYSTEME</b>	<b>4</b>
1) DIAGRAMME DE SEQUENCE	4
<b>II - CHOIX TECHNIQUES</b>	<b>6</b>
<b>A) CHOIX DU LANGAGE (SPECIFICATION ET DEVELOPPEMENT)</b>	<b>6</b>
<b>B) CHOIX DE L'IDE ET OUTIL DE CONSTRUCTION</b>	<b>6</b>
<b>C) CHOIX DE LA BASE DE DONNEES</b>	<b>6</b>
<b>III - ROADMAP</b>	<b>7</b>
<b>CONCLUSION</b>	<b>7</b>

# Équipe

LOISON Laureen  
VIREY Maxime  
GUERON Antoine

## Introduction

DomoDoor est un projet de gestion automatique et manuelle d'une porte de garage à l'aide d'une télécommande. La détection de la fermeture ou de l'ouverture de la porte se fera via un capteur qui s'activera lors des changements nécessaires afin d'actionner le moteur en mode tirage (ouvrir la porte de garage) ou poussée (fermer la porte). Chaque action sera transmise aux différents acteurs via le contrôleur dédié à cet effet.

Notre rôle est de concevoir l'architecture de ce système et d'en exprimer les contraintes associées. Pour cela, nous expliciterons dans ce rapport nos choix en matière de conception et de développement.

## I - Conception générale

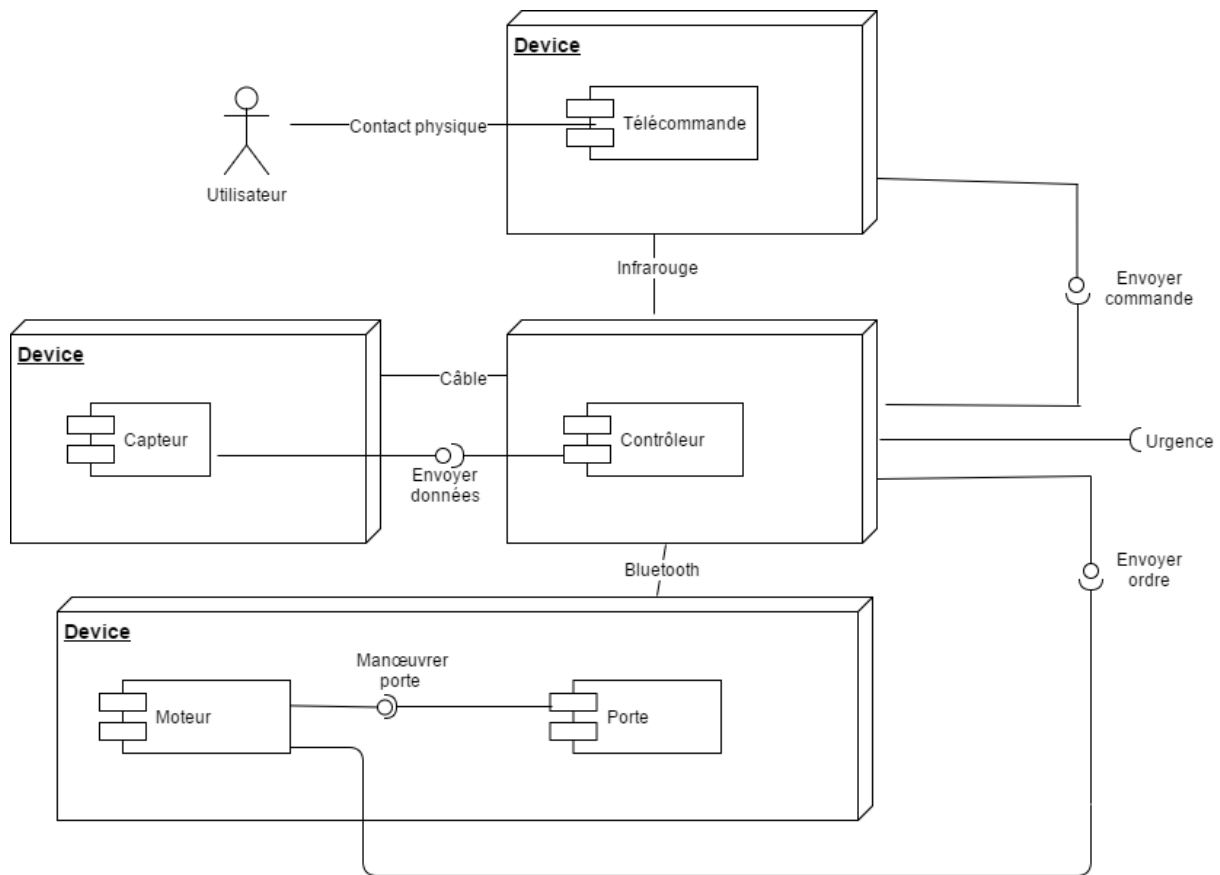
### A-Modélisation de l'architecture du système

#### 1) Architecture en couches

Notre architecture comportera 5 couches :

- Une couche **modèle** représentant les différents devices (capteur, télécommande, porte et moteur)
- Une couche **contrôleur** représentant le contrôleur de la porte du système
- Une couche **service** représentant les classes d'interaction et d'authentification avec le système
- Une couche **infrastructure** représentant la base de données MySQL (pour les utilisateurs)
- Une couche **présentation** représentant l'interface utilisateur (si les délais nous le permettent)
- Une couche **dao** représentant les entités de la base (principalement les utilisateurs ici)

## 2) Diagramme de composants et de déploiement



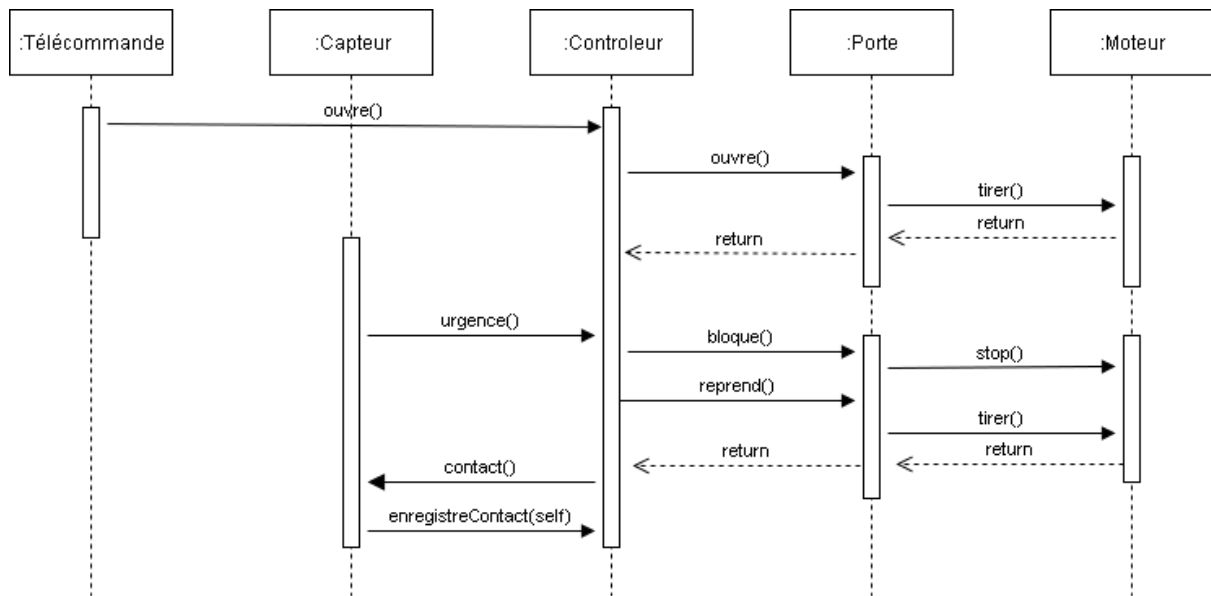
## B-Modélisation du comportement du système

### 1) Diagramme de séquence

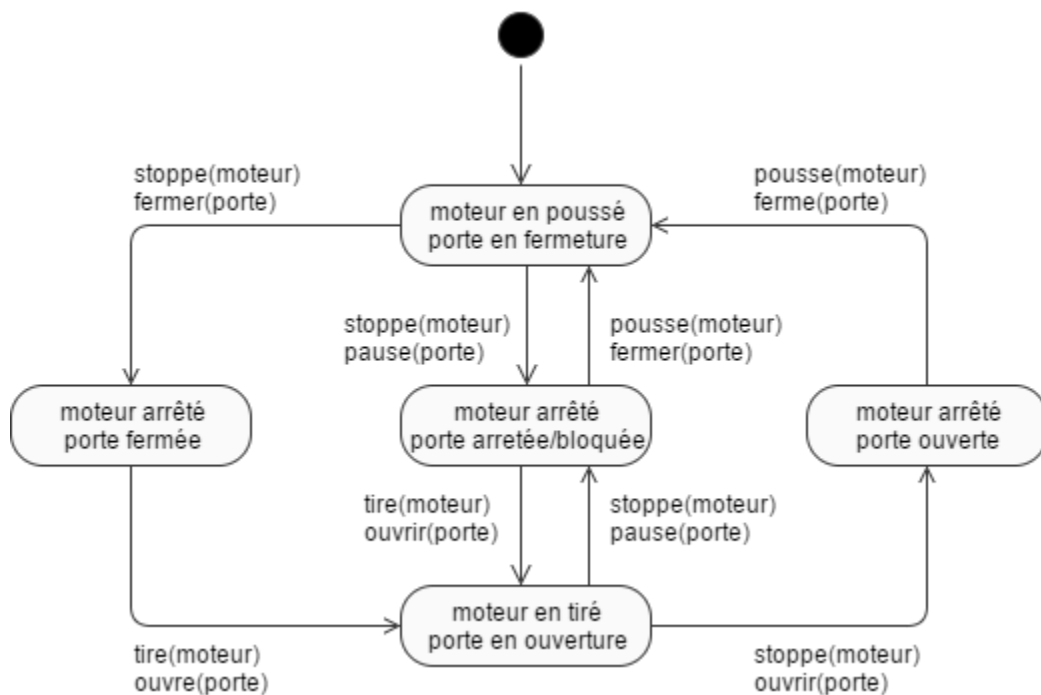
Afin de mieux appréhender les différents scénarios de notre application, nous avons réalisé des diagrammes de séquence pour les représenter.

Ce diagramme représente une porte qui s'ouvre par un appel émis depuis une télécommande, puis une urgence est détectée par le contrôleur et la porte se bloque. Enfin, la porte se débloque et reprend son ouverture.

## Conception générale - Projet Z DomoDoor



Le système peut être dans différents états. En effet, d'après le dossier de spécifications, chaque composant peut être dans différents états. Par exemple, la porte peut être dans un état "fermée" ou "ouvert" ou bien "en ouverture", le moteur peut être "en poussée" ou en train de "tirer". Bien sûr, chaque état de chaque composant est relié aux autres composants. En effet, si le moteur est dans un état "en poussé" cela signifie que la porte est dans un état "en fermeture". Le diagramme suivant illustre ces interactions.



## II - Choix techniques

### A) Choix du langage (spécification et développement)

Pour prouver notre conception, nous allons utiliser la notation que nous avons pu découvrir dans ce module : la notation Z. La spécification en Z va permettre de prouver les différents concepts que l'on va mettre en place.

Les preuves seront effectuées via un logiciel dédié à la vérification des preuves : Z/EVES.

Pour le choix du langage de développement, nous avons retenu plusieurs critères pour effectuer le choix :

- Les connaissances que nous possédons dessus
- L'activité de la communauté sur ce langage
- La documentation fournie et compréhensible du langage.

Le langage Java est celui qui répondait le plus à nos différents critères. De plus, l'intégration avec un outil de construction est facile à réaliser. Enfin, le langage Java permet d'intégrer de manière aisée une interface utilisateur.

### B) Choix de l'IDE et outil de construction

Ayant des IDE différents au sein du groupe, nous avons choisi de conserver nos propres IDE à savoir IntelliJ et Eclipse. Les deux IDE étant compatibles au niveau projet, cela ne posera pas de problème théoriquement.

De plus, nous avons fait le choix d'utiliser Maven comme outil de construction afin de gérer les dépendances plus aisément et du fait que celui-ci est compatible avec les deux IDE que nous utilisons.

### C) Choix de la base de données

Concernant la création de notre base de données, nous avons fait le choix d'utiliser une base de données relationnelle MySQL. Ce choix est motivé d'après les connaissances que nous avons en base de données et par le fait que nous considérons qu'une base de données relationnelle est la plus adaptée pour notre projet.

## III - RoadMap

06/04/2017 : Réalisation de la conception générale  
01/05/2017 : Réalisation de la conception détaillée (spécifications en Z)  
20/05/2017 : Conception Java  
20/06/2017 : Application et livrable

## Conclusion

La rédaction du dossier de conception générale nous a permis de comprendre l'utilité de l'utilisation des méthodes de spécifications formelles. Ce dossier permet de réfléchir à la conception de notre application, des problèmes que certaines solutions pourraient causer et donc de trouver la solution la plus adaptée à nos besoins.

Le dossier de conception préliminaire permet de décider de la conception générale de notre application, de définir quelles technologies nous allons utiliser et pour quelles raisons nous avons fait ces choix. Il permet de proposer une solution de base à partir de laquelle nous devrons établir une conception plus détaillée.