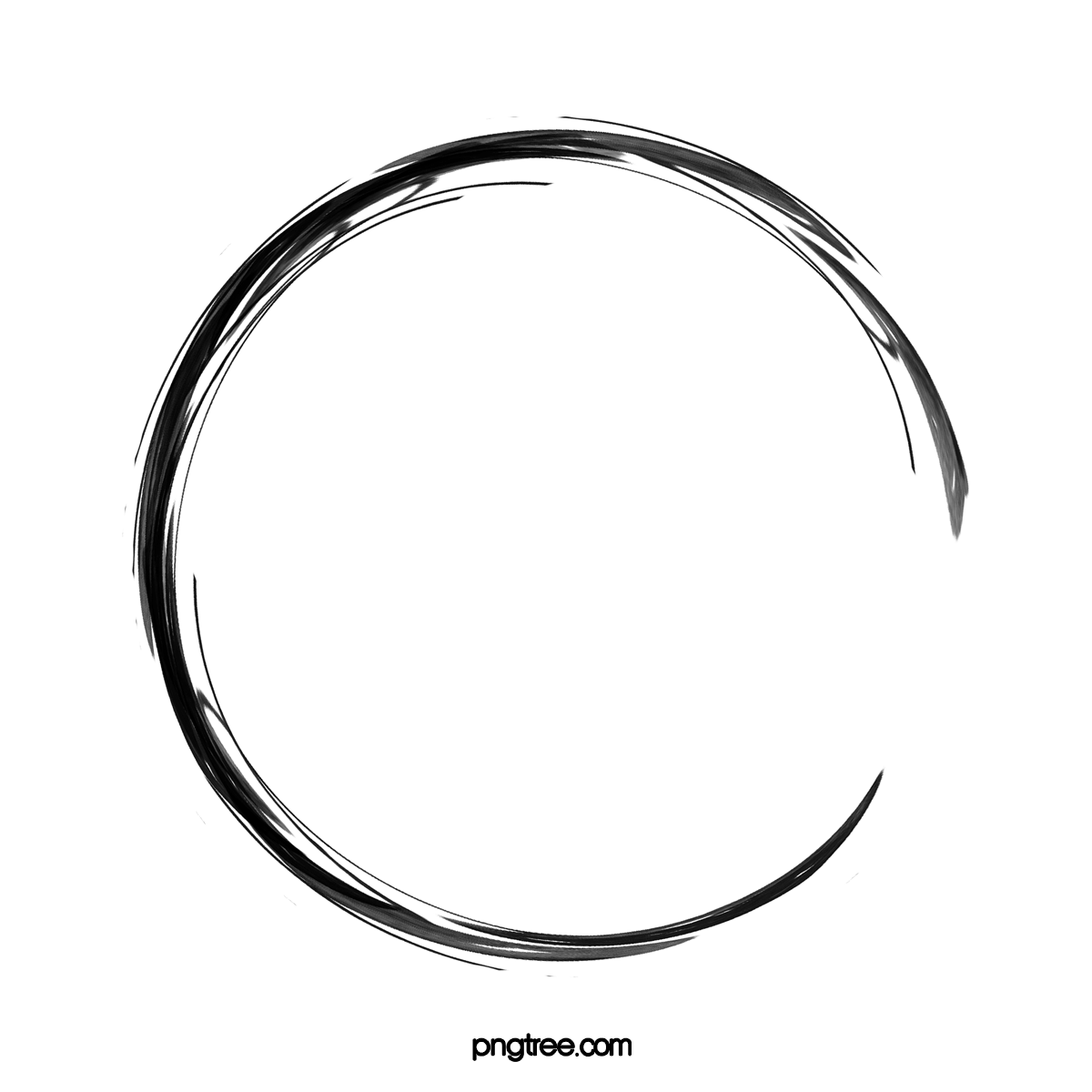
**S SFL5 : SYSTEME ESCAPE GAME   
113E PORTE**



Une image contenant dessin, signe

Description générée automatiquement

**Sommaire**

[I. Présentation du projet 2](#_Toc40619084)

[A. Enoncé général du besoin 2](#_Toc40619085)

[B. Aspect contractuel du projet 2](#_Toc40619086)

[II. Expression du besoin 3](#_Toc40619087)

[Système de supervision 3](#_Toc40619088)

[Gestion de la base de données 3](#_Toc40619089)

[III. Identification des équipements 4](#_Toc40619090)

[A. Matériels 4](#_Toc40619091)

[B. Schéma de câblage du système 5](#_Toc40619092)

[IV. Ressources mises à dispositions 7](#_Toc40619093)

[A. Logiciel /environnement de développement 7](#_Toc40619094)

[B. Librairies utilisées 7](#_Toc40619095)

[C. Documentations 7](#_Toc40619096)

[V. Analyse 8](#_Toc40619097)

[A. Tâches à réaliser 8](#_Toc40619098)

[B. Diagramme d’exigence 10](#_Toc40619099)

[C. Diagramme de cas d’utilisation 11](#_Toc40619100)

[D. Les mécanismes 12](#_Toc40619101)

[Mécanisme 1 : Echiquier 12](#_Toc40619102)

[Mécanisme 2 : Lion 14](#_Toc40619103)

[Mécanisme 3 : Terre 16](#_Toc40619104)

[Mécanisme 4 : Feu 18](#_Toc40619105)

[Mécanisme 5 : Eau 20](#_Toc40619106)

[Mécanisme 6 : Air 22](#_Toc40619107)

[Mécanisme 7 : Katana 23](#_Toc40619108)

[Mécanisme 8 : Riz 24](#_Toc40619109)

[Mécanisme 9 : Quatre Eléments 26](#_Toc40619110)

[E. Application Web de supervision 29](#_Toc40619111)

[Application Web : Visualiser l’état de la salle 29](#_Toc40619112)

[Application Web : Piloter les actionneurs 31](#_Toc40619113)

[F. I2C : Communication entre les Arduinos et la Raspberry 33](#_Toc40619114)

[G. La Base de Données 34](#_Toc40619115)

# Présentation du projet

## Enoncé général du besoin

Mme. Sterenn LE GOFFIC et M. Jean-Charles Douguet, les deux gérants de la société de la 13ème Porte, souhaitent que le système technique actuellement en place soit recréé entièrement afin de corriger les erreurs de conception et les différents bugs existants. Il est également demandé l’ajout de plusieurs fonctionnalités permettant d’améliorer le travail du superviseur.

Pour ce projet les capteurs et les actionneurs nécessaires nous seront mis à disposition.

Deux applications Web permettront au client de gérer les mécanismes pour un résultat optimal.

## Aspect contractuel du projet

|  |  |
| --- | --- |
| Commanditaire | Entreprise la 13eme Porte |
| Acteurs | Exploitant |
| Temps de réalisation | Du 8 janvier |
| Equipe de développement | 4 étudiants |
| Professeur référent Monsieur | Monsieur Sébastien ANGIBAUD |

# Expression du besoin

Actuellement, les nombreux mécanismes du jeu dans la salle sont gérés par quatre Arduino UNO. Chaque Arduino a la charge de plusieurs mécanismes distincts. Les gérants souhaitent revoir ce découpage dans le nouveau système en associant à chacune des neuf étapes du jeu un Arduino dédié. Une étape sera appelée mécanisme dans la suite de ce document.

Les différents mécanismes du jeu sont les suivants :

* Mécanisme 1. L’échiquier
* Mécanisme 2. Le lion
* Mécanisme 3. L’élément TERRE
* Mécanisme 4. L’élément FEU
* Mécanisme 5. L’élément EAU
* Mécanisme 6. L’élément AIR
* Mécanisme 7. Le Katana
* Mécanisme 8. Le riz
* Mécanisme 9. Les quatre éléments.

### Système de supervision

Le client souhaite disposer de deux fonctionnalités dédiées à la supervision et utiles à la maintenance:

* Visualiser depuis un ordinateur l’état du système. L’interface devra afficher à la fois l’état d’avancement dans le jeu, mais aussi la valeur de chaque capteur et actionneur.
* Piloter chaque actionneur depuis une interface afin de faciliter la maintenance

### Gestion de la base de données

La base de données située sur un serveur à part permet de stocker différentes informations des mécanismes c’est-à-dire l’état des mécanismes, l’état des actionneurs et la valeur des capteurs.

# Identification des équipements

## Matériels

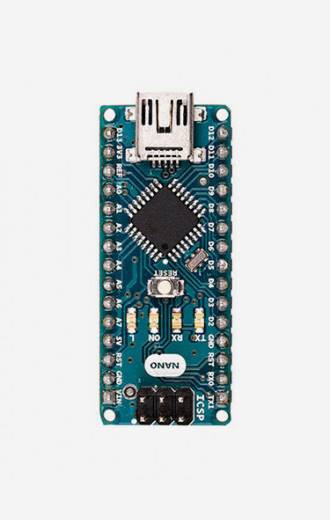
#### Raspberry

Une Raspberry Pi 3B est nécessaire pour notre projet et se présente sous cette forme :



#### Arduino

Neuf Arduinos Nanos sont nécessaires pour les 9 mécanismes du jeu.



Et chaque Arduino nécessite également du matériel.

Voici la liste globale du matériel nécéssaire pour toutes les Arduinos :

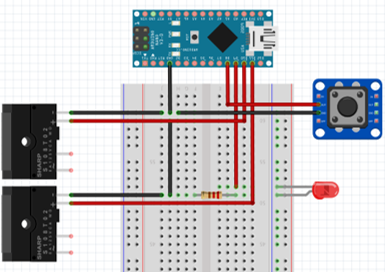
* Quatre capteurs à effet hall A3144
* Un module de commutateur de clé 3 pin 1NO1NC
* Un afficheur 7 segments
* Un capteur d’humidité
* Un interrupteur à bascule
* Sept capteurs photosensibles
* Des boutons poussoir
* Des leds
* Un capteur de fin de course
* Un capteur de poids HX711
* Des relais

#### Un poste informatique pour le serveur 13e Porte/supervision

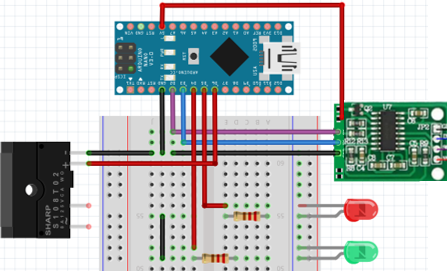
Un poste informatique est nécessaire pour l’application Web et le Serveur BDD.

## Schéma de câblage du système

#### Mécanisme 4 : le Riz



#### Mécanisme 8 : Le Riz



#### Liaison i2c



# Ressources mises à dispositions

## Logiciels/environnement de développement

* Excel pour le diagramme de GANTT
* MagicDraw/StarUML pour les diagrammes UML
* Fritzing pour les schémas câblages
* Arduino pour le développement Arduino
* MySQL pour la base de données
* NetBeans pour l’application Web
* Putty pour une connexion SSH avec la Raspberry
* FileZilla pour une connexion SFTP avec la Raspberry
* Notepad++ pour le développement python
* Github Desktop pour se partager le travail

## Librairies utilisées

Arduino :

* Wire
* HX711

Python :

* Smbus
* Time
* Thread

## Documentations

* sfl5\_Système\_Escape\_Game\_13e\_Porte.pdf
* description\_des\_sous\_systèmes\_V2.pdf

# Analyse

## Tâches à réaliser

Le projet est décomposé en 4 tâches distinctes.

Etudiant 1 : Constantin MINOS

* Installer Configurer la Raspberry
* Gérer le Mécanisme 4
* Gérer le Mécanisme 8
* Gestion du bus I2c entre Arduino et Raspberry :

Envoie des ordres de pilotage de la Raspberry à l’Arduino et les traiter

Envoi des mesures et des états des actionneurs sur la Raspberry et les traiter

Etudiant 2 : Corentin BRENNY

* Créer la base de données
* Installer les serveurs Apache et MySql
* Gérer le Mécanisme 6
* Gérer le Mécanisme 7
* Récupérer la valeur des capteurs et l’état des actionneurs de la Raspberry au serveur

Etudiant 3 : Joshua PINNEAU

* Configuration réseau du matériel
* Création d’une application Web sur l’affichage de l’état de la salle
* Gérer le Mécanisme 1
* Gérer le Mécanisme 2
* Gérer le Mécanisme 3

Etudiant 4 : Thomas CADEAU

* Création d’une application Web de pilotage des actionneurs
* Gérer le Mécanisme 5
* Gérer le Mécanisme 9
* Envoie des ordres de pilotage du PC de supervision à la Raspberry



**Application Web de pilotage des actionneurs**

**Application Web sur l’affichage de l’état de la salle**

Constantin

Thomas

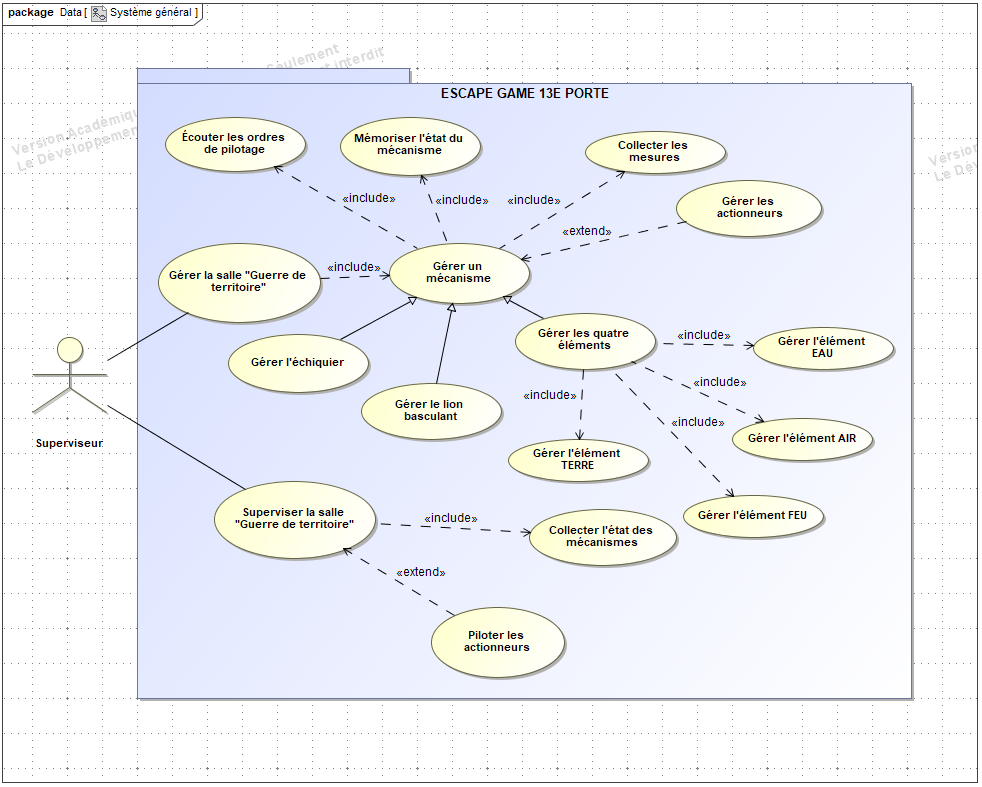
Joshua

Corentin

## Diagramme d’exigence



## Diagramme de cas d’utilisation



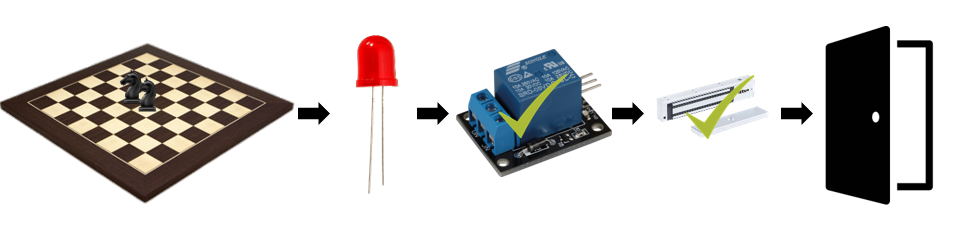
## Les mécanismes

### Mécanisme 1 : Echiquier

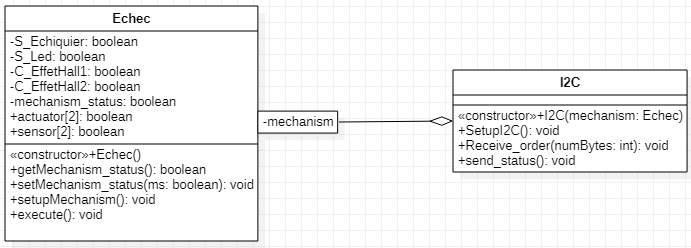
#### Synopsis du mécanisme

Le mécanisme repose sur deux **capteurs à effet Hall**.

Sous un plateau d’échec sont fixés deux capteurs à effet Hall. Sur deux pièces (cavalier noirs) sont fixés deux aimants. Quand les joueurs placent les deux cavaliers sur les bonnes cases les aimant fixés sur ces derniers ont pour effet d’activer les deux capteurs à effet de hall. Résultat : La LED de contrôle sur le tableau s’allume. Le relais s’active afin que le verrou magnétique libère la porte qui mène vers la seconde salle



#### Diagramme de classe



#### Diagramme de séquence

Une image contenant texte, carte, ordinateur

Description générée automatiquement

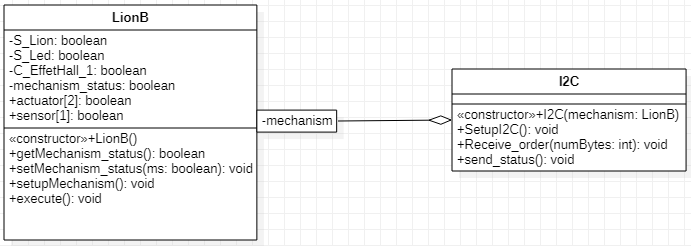
### Mécanisme 2 : Lion Basculant

#### Synopsis du mécanisme

Le mécanisme repose sur un **capteur à effet Hall**.

Selon la mesure, un électroaimant est activé ou désactivé via un relais et une LED s’allume ou non sur le tableau de contrôle.

#### Diagramme de classe



#### Diagramme de séquence

Une image contenant texte, carte

Description générée automatiquement

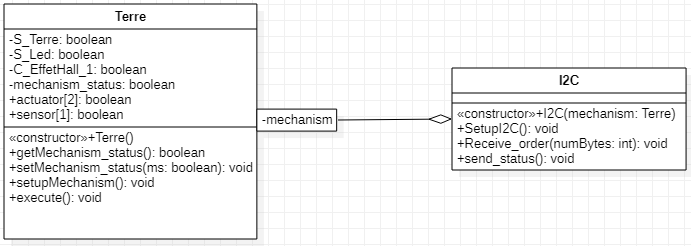
### Mécanisme 3 : Terre

#### Synopsis du mécanisme

Le mécanisme repose sur un **capteur à effet Hall**.

Selon la mesure, (i) un moteur et une LED sont activés ou désactivés via un relais pendant un laps de temps (ii) une sortie est paramétrée en destination du mécanisme des quatre éléments.

#### Diagramme de classe



#### Diagramme de séquence

Une image contenant texte, carte

Description générée automatiquement

### Mécanisme 4 : Feu

#### Synopsis du mécanisme

Le mécanisme 4 repose sur **un interrupteur à clef**.

S’il y a un changement de position positif :

1. Une LED témoin s’allume sur le tableau de contrôle.
2. Une « Tête de Dragon » sort du plafond (S2). En effet, la désactivation de l’électroaimant a pour effet de libérer une trappe au plafond.
3. Une machine à fumée (220 Volt) est allumée via un relais. Cette machine à fumée est dissimilée dans la « tête de dragon » afin que ce dernier crache de la fumée lorsqu’il apparait aux joueurs.
4. 5 LED blanches s’allument afin au-dessus afin d’éclairer cette dernière.
5. L’élément FEU (LED) s’allume sur la tablette à destination des joueurs.



#### Diagramme de classe

Une image contenant texte, capture d’écran

Description générée automatiquement

#### Diagramme de séquence



### Mécanisme 5 : Eau

#### Synopsis du mécanisme

Le mécanisme repose sur **un capteur d’humidité** et **un interrupteur à bascule**.

Selon la mesure du capteur d’humidité, (i) un électroaimant est activé ou désactivé via un relais (ii) un moteur d’une fontaine est activé ou désactivé via un relais (iii) une sortie est paramétrée en destination du mécanisme des quatre éléments.

Selon l’état de l’interrupteur à bascule, un électroaimant est activé ou désactivé via un relais

#### Diagramme de classe

Une image contenant capture d’écran

Description générée automatiquement

#### Diagramme de séquence

Une image contenant texte, carte

Description générée automatiquementUne image contenant texte, carte

Description générée automatiquement

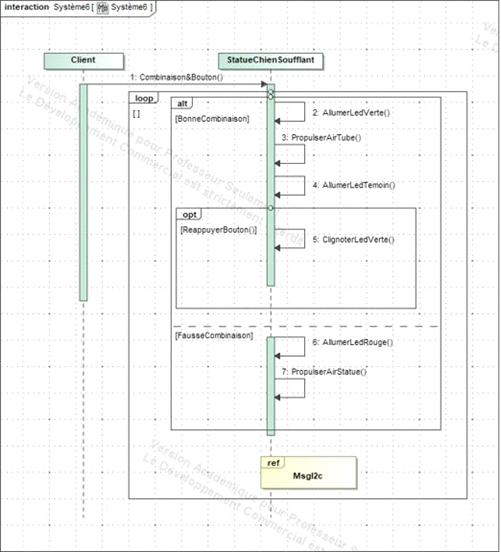
### Mécanisme 6 : Air

#### Synopsis du mécanisme

Le mécanisme repose sur **sept capteurs photosensibles** et **un bouton poussoir**.

Lorsque le bouton est pressé, selon des conditions portant sur les valeurs de sept capteurs photosensibles, (i) deux électrovannes sont activées ou désactivées via un relais (ii) une led est allumée ou non (ii) une sortie est paramétrée en destination du mécanisme des quatre éléments.

#### Diagramme de séquence



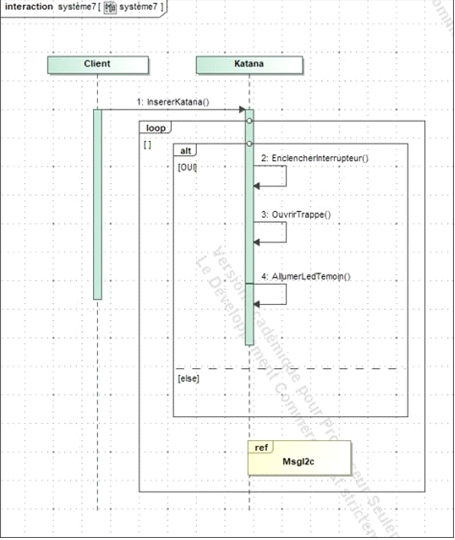
### Mécanisme 7 : Katana

#### Synopsis du mécanisme

Le mécanisme repose sur **un interrupteur fin de course**.

Selon l’état de l’interrupteur fin de course, un solénoïde poussant est activé ou désactivé via un relais.

#### Diagramme de séquence



### Mécanisme 8 : Riz

#### Synopsis du mécanisme

Le mécanisme 4 repose sur **un capteur de poids.**

Si le capteur détecte une mesure comprise entre 48 et 52g:

1. La LED rouge s’éteint et la LED verte s’allume.
2. Une LED témoin s’allume au panneau de contrôle.
3. Un électroaimant (12V) est désactivé via un relais (5V) libérant ainsi la chute d’un tableau.



#### Diagramme de classe

Une image contenant capture d’écran

Description générée automatiquement

#### Diagramme de séquence



### Mécanisme 9 : Quatre Eléments

#### Synopsis du mécanisme

Le mécanisme repose sur un bouton poussoir et les quatre sorties liées aux éléments (mécanismes 3, 4, 5 et 6).

Selon les valeurs de ces quatre sorties, un électroaimant et une led sont activés ou désactivés.

#### Diagramme de classe

Une image contenant capture d’écran

Description générée automatiquement

#### Diagramme de séquence

Une image contenant texte, carte

Description générée automatiquement

Une image contenant texte, carte

Description générée automatiquement

## Application Web de supervision

### Application Web : Visualiser l’état de la salle

#### Synoptique

#### Diagramme de séquence

#### Une image contenant texte, carte Description générée automatiquementStory Board de l’appliation Web



### Application Web : Piloter les actionneurs

#### Synoptique

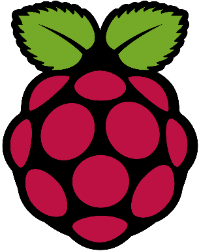
Pour cette partie, le PC de supervision doit pouvoir gérer et visualiser à distance l’état de chacun des mécanismes via une application WEB. De plus il doit pouvoir récupérer les informations transmis par les mécanismes depuis la Raspberry.

Une image contenant horloge, dessin

Description générée automatiquement

La Raspberry relié par liaison I2C avec tous les mécanismes, (les 9 Arduino Nano) reçois toutes les informations des mécanismes qui ensuite les envois par sockets au PC de supervision. L’installation et la configuration de la Raspberry seras faites en commun avec l’étudiant 1.

Une image contenant dessin

Description générée automatiquement

* **Langage de développement :** Programmation par sockets en Python.
* **Logiciel utilisé :** Putty (Émulateur de terminal).

#### Programmation par socket

Typiquement, un socket respecte un flux spécifique d'événements pour qu'elle fonctionne. Pour un modèle client-serveur orienté connexion, le socket du processus serveur attend la demande d'un client. Pour ce faire, le serveur doit d'abord établir une adresse que les clients peuvent utiliser pour trouver et se connecter au serveur. Lorsqu'une connexion est établie avec succès, le serveur attend que les clients demandent un service. L'échange de données client-serveur aura lieu si le client se connecte au serveur via le socket. Le serveur répondra alors à la demande du client et lui enverra une réponse.

#### Développement

Une application WEB doit être crée pour le poste de supervision de l’administrateur. Une interface pour pouvoir piloter les différents actionneurs doit être réalisé. Le superviseur pourra démarrer ou arrêter chaque actionneur. L’ordre sera tout d’abord transmis par liaison WIFI à la Raspberry, qui transmettra par liaison I2C cet ordre au mécanisme visés (Arduino nano correspondant au mécanisme). Le pilotage à distance des actionneurs devra inhiber la décision décrite dans la section Gérer les neuf mécanismes.

Une image contenant assiette, dessin

Description générée automatiquementUne image contenant signe, vert, peint, assis

Description générée automatiquement

#### Maquette de l’interface

Une image contenant capture d’écran

Description générée automatiquement

## I2C : Communication entre les Arduinos et la Raspberry

#### Synopsis de la liaison i2c

Afin d’établir une communication entre les Arduino à la Raspberry nous allons passer par le protocole i2C, cela va nous servir pour :

* Envoyer l’état des actionneurs et les valeurs des capteurs des Arduino à la Raspberry
* Envoyer les ordres de pilotage du PC de supervision de la Raspberry aux Arduino.

Le protocole I2C permet à plusieurs composants de dialoguer entre eux de manière bidirectionnelle mais en half-duplex uniquement. Et grâce à seulement trois fils : Un signal de donnée (SDA), un signal d’horloge (SCL), et un signal de référence électrique (Masse).

Chaque périphérique sur le bus I2C est adressable, avec une adresse unique pour chaque périphérique du bus

Il y a un principe de maitre et esclave. Ici nous utiliseront le Raspberry Pi en tant que maitre et les Arduino en tant qu’esclave.

Le maitre (la Raspberry) est le composant qui initialise un transfert, génère le signal d’horloge et termine le transfert. Dans notre cas ilsera récepteur et émetteur.

Les esclaves (Les Arduinos) sont les composants adressés par un maître. Dans notre cas ils seront récepteurs et émetteurs.

#### Diagramme de séquence

Une image contenant texte, carte

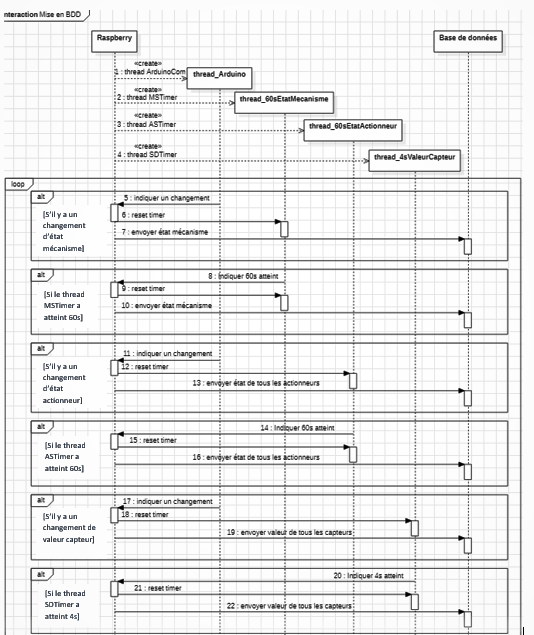
Description générée automatiquement

## La Base de Données

#### Synopsis de la base de données

#### Maquette de la Base de Données

#### Diagramme de séquence

****