

SFL6

Escape Game Téléthon :

Rapport d'analyses et de spécifications



Projet d'étude de BTS Systèmes Numériques I.R.

2020

Table des matières

I.	Présentation du projet :	3
a)	Enoncé général :	3
b)	Aspect contractuel du projet :	3
II.	Expression du besoin :	3
a)	Gestion des bases de données :	3
b)	Gestion des comptes superviseurs et des créneaux :	4
c)	Gestion de la détection des médailles :	4
d)	Gestion de la supervision :	4
III.	Identification des équipements :	5
a)	Synoptique des systèmes :	5
b)	Description des équipements :	7
IV.	Identification des besoins :	8
c)	Besoins fonctionnels :	8
d)	Besoins non fonctionnels :	10
V.	Ressources mises à disposition :	11
a)	Ressources matérielles :	11
b)	Ressources logicielles :	11
c)	Librairies utilisées :	11
d)	Documentation étudiée :	11
VI.	Analyse :	12
a)	Tâches réparties :	12
b)	Diagramme d'exigences :	13
c)	Diagramme de cas d'utilisation :	14
d)	Diagrammes de séquences :	15
e)	Diagrammes de classes :	24
f)	Modèle relationnel de la base de données de l'application Web :	26

I. Présentation du projet :

a) Enoncé général :

L'association du Téléthron de la ville de Saint Colomban (Loire-Atlantique) organise des escapes games à temps court durant la période du Téléthron. Au vu de l'intérêt que porte le public sur ces jeux, l'association souhaiterait créer un escape game plus conséquent d'une durée d'une heure. Les recettes de celui-ci seront ensuite reversées au Téléthron.

Le groupe de bénévoles composant l'association n'ayant pas les connaissances requises en informatique, il a été choisi de confier l'entière réalisation de ce projet aux étudiants du campus Saint Félix La Salle de Nantes.

Le matériel sera acheté par le campus et mis à disposition du Téléthron pour une durée de deux mois. Ensuite, celui-ci sera utilisé à des fins pédagogiques (Travaux Pratiques) par les enseignants du BTS Systèmes Numériques.

b) Aspect contractuel du projet :

<i>Commanditaire :</i>	Association du Téléthron de Saint Colomban
<i>Interlocuteur :</i>	Monsieur Gérard HERVOUET
<i>Acteurs :</i>	Exploitant
<i>Temps de réalisation :</i>	Du 14 janvier à fin juin 2020
<i>Equipe de développement :</i>	3 étudiants
<i>Professeur référent :</i>	Monsieur Sébastien ANGIBAUD

II. Expression du besoin :

a) Gestion des bases de données :

La première base de données est essentielle à l'utilisation de l'Application web. En effet celle-ci doit pouvoir permettre à l'administrateur et aux superviseurs de se connecter grâce à un mot de passe et un login. Ces informations de connexion vont aussi permettre à l'administrateur de modifier, supprimer ou ajouter des comptes superviseurs. La base de données doit aussi contenir les informations des créneaux pour permettre aux superviseurs de les modifier, supprimer ou d'en ajouter, mais aussi consulter les informations des créneaux réservés (nom du joueur, numéro de téléphone du joueur, nombre de joueurs).

La seconde base de données est nécessaire à l'utilisation de l'application de supervision. En effet, celle-ci doit pouvoir enregistrer les indices qui peuvent être envoyés à la base de données mais également stocké les équipes avec différents champs (nom de l'équipe, score, date et heure de début de jeu, date et heure de fin de partie). L'utilisateur n'a pas à accéder à cette base de données, les données auxquelles il souhaiterait accéder sont accessibles depuis des fenêtres de l'application.

b) Gestion des comptes superviseurs et des créneaux :

L'application web doit permettre à l'administrateur de modifier, supprimer ou ajouter des comptes superviseurs. Il est impossible de supprimer un superviseur pour lequel il existe un créneau non passé et réservé. L'application de réservation doit aussi permettre aux superviseurs de modifier, supprimer ou ajouter des créneaux. Le superviseur est prévenu lorsqu'il supprime un créneau déjà réservé. La durée entre deux créneaux doit être supérieurs à 30 minutes. Tout visiteur pourra réserver un créneau libre en précisant son nom, son numéro de téléphone ainsi que le nombre de joueurs qui doit être entre trois et cinq.

c) Gestion de la détection des médaillons :

L'application du sous-système médaillons doit pouvoir lire les UID (User Identifier) et allumer des LED correspondantes dans la seconde selon l'état du capteur. Si les quatre UID sont détectés, donc si les badges RFID sont correctement placés l'ouverture de la porte doit se faire via une gâche électrique contrôlée par un module relais. Dans un même temps un ordre de fin de partie doit être envoyé à l'application de supervision. Cet ordre doit être ensuite interprété par celle-ci afin de mettre fin à la partie.

d) Gestion de la supervision :

L'application de supervision doit permettre au superviseur la surveillance et l'interaction avec les joueurs. Cela se fera par l'intermédiaire d'une caméra IP installée dans la salle ainsi qu'un afficheur géant de type LCD. La fin de partie est engagée si l'un des paramètres suivants est présent : le chronomètre arrive à son terme, l'application reçoit l'ordre de fin de partie émis par le sous-système médaillons ou le superviseur décide de stopper prématurément et définitivement la partie en cours. Dans ces trois situations, on affiche le score de l'équipe, son temps, ainsi que le classement vis-à-vis des autres équipes.

III. Identification des équipements :

a) Synoptique des systèmes :

Schématisation du réseau local :

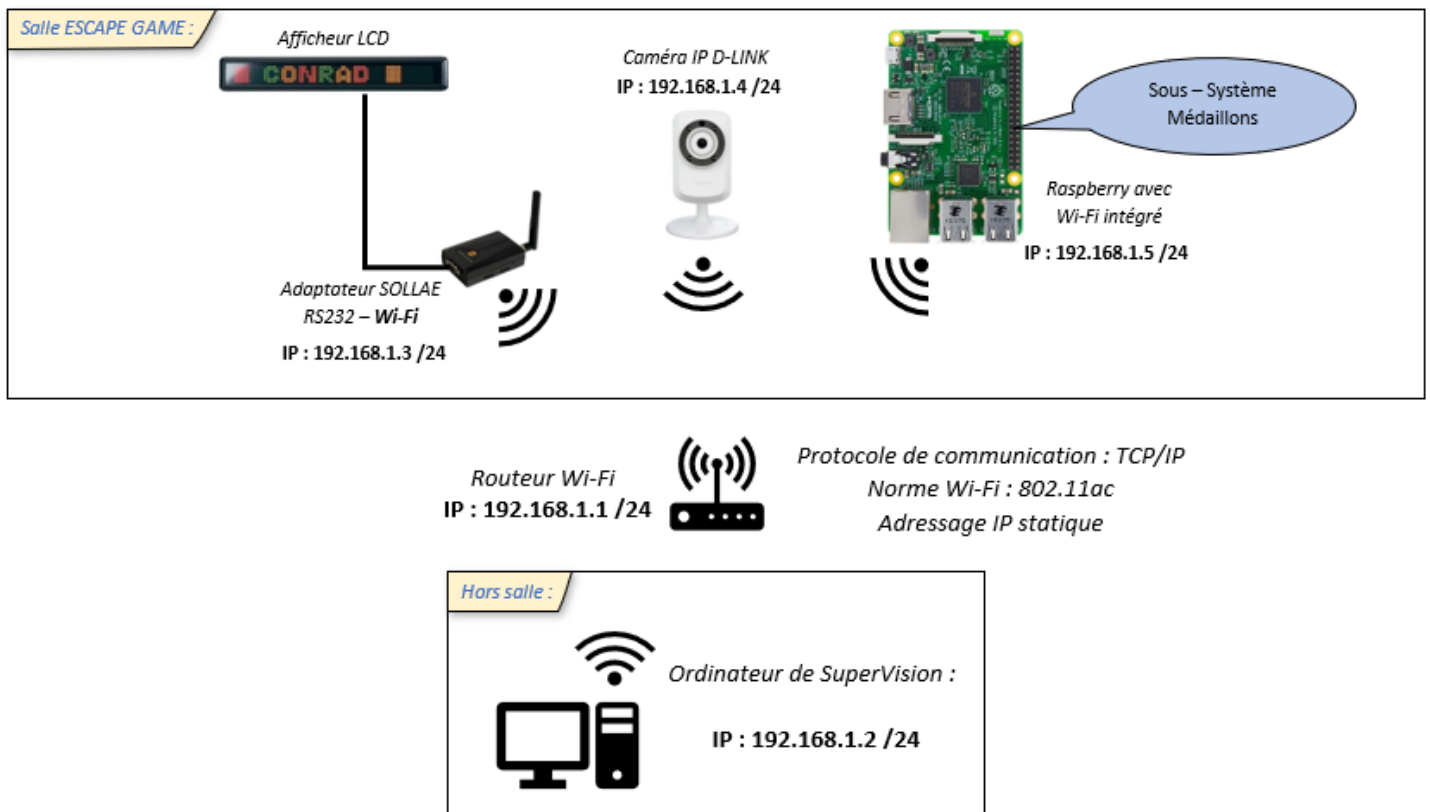


Image 1 : Schéma du réseau local.

Schématisation du câblage du sous-système médailles :

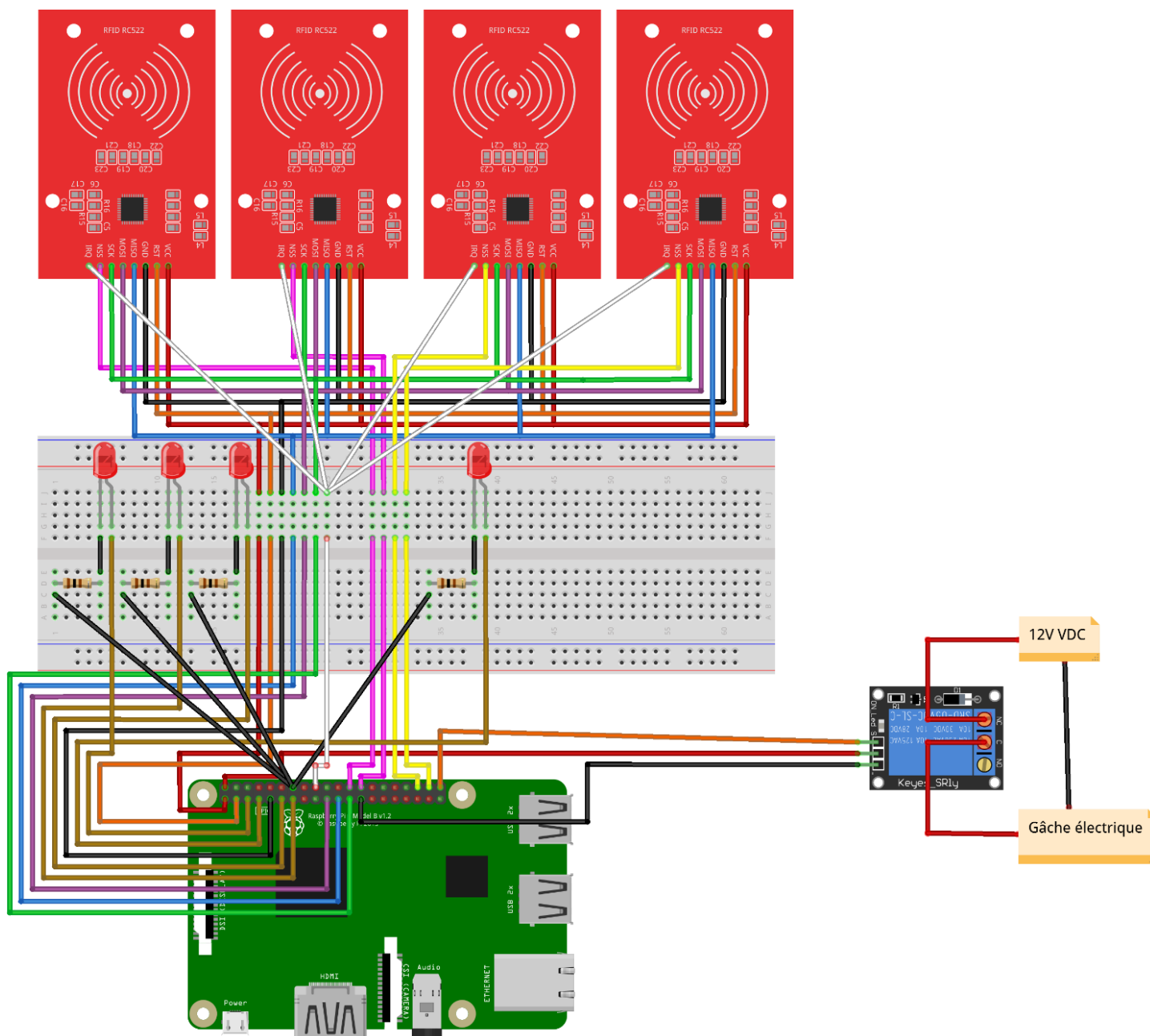


Image 2 : Schéma de câblage du sous-système médaille

Schématisation du câblage du sous-système énigme :

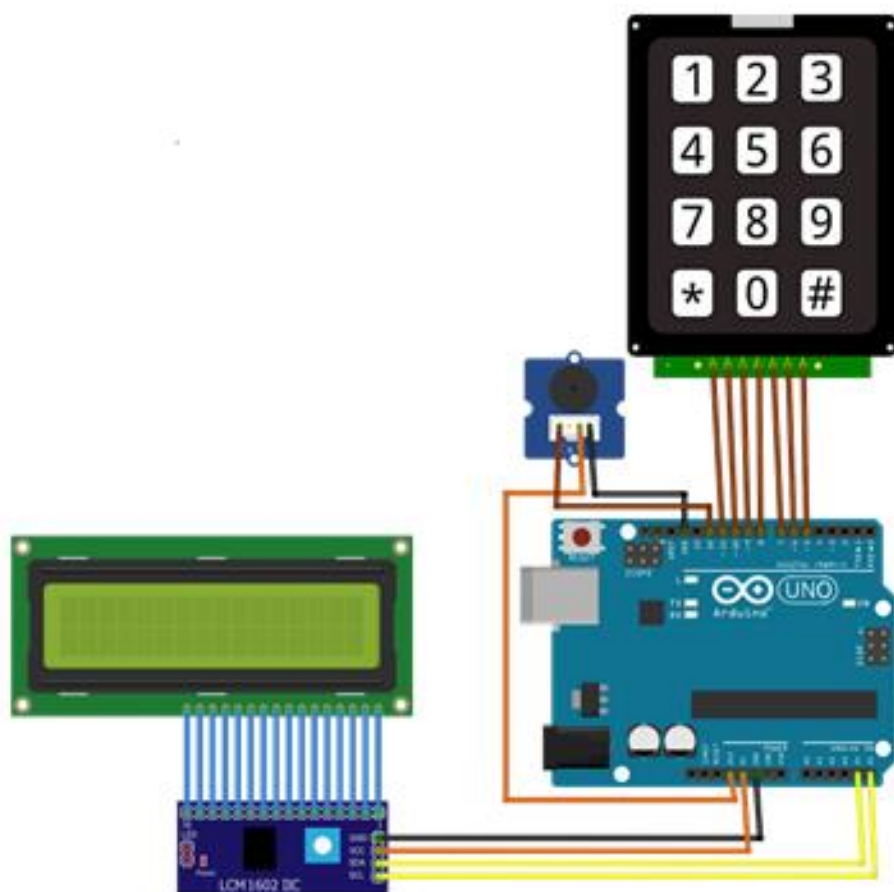


Image 3 : Schéma de câblage du sous-système énigme musicale

b) Description des équipements :

IV. Identification des besoins :

c) Besoins fonctionnels :

Application de supervision :

Lors du lancement d'une partie, l'utilisateur saisit le nom de l'équipe. Celui-ci est ensuite immédiatement sauvegardé dans la base de données ainsi que l'heure du début de la partie.

En fin de partie, il est nécessaire de pouvoir récupérer ces données pour permettre leur traitement. On envoie ainsi l'heure de fin de partie et le score de l'équipe. On récupère ensuite ces informations pour la création de statistiques d'équipes.

Lors du déroulement de la partie, plusieurs communications sont nécessaires au bon fonctionnement :

La communication avec la caméra IP :

Par l'intermédiaire du protocole TCP/IP, on récupère l'image émise et on l'actualise toutes les secondes pour une meilleure fluidité.

La communication avec l'afficheur LCD :

Par l'intermédiaire du protocole TCP/IP, on envoie la trame avec le message souhaité à l'adaptateur SOLLAE. Celui-ci le convertit en RS232 pour l'envoyer à l'afficheur qui décrypte le message et l'affiche avec le style (police d'écriture, couleur) demandé.

Programme sous Raspberry :

Le programme doit pouvoir :

- Récupérer les UID des badge RFID
- Allumer une LED associée en fonction de l'état du capteur
- Si les 4 UID sont valides, ouvrir la porte et envoyer l'ordre de fin de partie

Application Web :

Le système doit permettre au **superviseur** :

- De se connecter
- D'ajouter des créneaux
- De modifier des créneaux
- De supprimer des créneaux

Le système doit permettre à l'**administrateur** :

- De se connecter
- D'ajouter des comptes superviseurs
- De modifier des comptes superviseurs
- De supprimer des comptes superviseurs

Le système doit permettre aux **joueurs** :

- De réserver un créneau
- De consulter son créneau
- De modifier son créneau
- D'annuler son créneau

Système énigme musique :

Le système doit permettre aux **joueurs** :

- De taper un code
- Visualiser un message
- Entendre une musique

d) Besoins non fonctionnels :

Contraintes liées au développement :

L'application sous Raspberry doit être développée en Python et l'utilisation d'un bus SPI était obligatoire. Les LEDs doivent s'allumer ou s'éteindre dans la seconde en fonction de l'état du capteur.

La carte Arduino Uno nécessite de programmer sur le logiciel Arduino en programmation Arduino.

L'application web est développée en HTML/PHP/CSS sous NetBeans. L'authentification pour le superviseur et l'administrateur est imposée.

Les planifications ont été réalisées sous MSProject et les schémas d'analyse réalisées sous MagicDraw.

L'application de supervision est quant à elle développée en C# sous l'environnement de Visual Studio (licence fournie pour les étudiants). Les classes sont représentées sous forme de fenêtres WPF ou de formulaire de type Form.

Contrainte temporelle :

Le temps accordé pour ce projet est d'environ 200h.

Contrainte esthétique :

Le système doit avoir une interface accueillante permettant un dialogue simple entre l'utilisateur et l'application.

Les informations figurant dans l'application web et l'application de supervision doivent être claires et précises. Les interfaces doivent être facile à prendre en main et sont destinées à des personnes n'ayant pas de connaissances approfondies en informatique.

Contraintes liées au confinement :

Au début de la phase de développement nous avons fait face au confinement dû au Covid-19.

Cela nous a demandé d'emmener le maximum de matériel nécessaire au projet chez nous.

Malheureusement, nous n'avons pas eu la possibilité d'emporter l'intégralité de celui-ci. Nous avons dû travailler avec nos ordinateurs personnels et nos connexions internet qui peuvent être limités et matière de puissance ou de débit. Chaque étudiant du projet a organisé ses temps de travail comme il le souhaite sur la base d'un minimum de 15 heures par semaine. Il était impératif de compléter régulièrement de journal d'activités et de réaliser des mises à jour sous GitHub.

De plus dû à un manque de matériel (jumper), la commande et donc l'attente pour la livraison à grandement augmentée, il a fallu attendre plus de deux semaines pour faire réception d'un colis et enfin des tests unitaires.

V. Ressources mises à disposition :

a) Ressources matérielles :

- Un afficheur LCD géant de type McCrypt
- Un adaptateur RS232/Wi-Fi de type SOLLAE H85N
- Une caméra IP Dlink
- Une carte Arduino Uno
- Un buzzer
- Un pavé numérique 3*4
- Un écran LCD 16*2
- Kit Raspberry pi 3 B
- 4 lecteurs RFID
- Module relais 12V
- Gâche électrique

b) Ressources logicielles :

- MagicDraw UML (création de diagrammes UML et SysML)
- Visual Studio (environnement de développement)
- Microsoft Project (diagramme de Gantt)
- MySQL (gestion des bases de données)
- Suite Microsoft Office 365 (Word, Excel, Powerpoint) pour la création des comptes-rendus et documents de présentation (revues de projet)
- NetBeans (environnement de développement HTML, CSS, PHP, ...)
- Thonny (environnement de développement)

c) Librairies utilisées :

- Keypad.h (Gestion des claviers)
- Wire.h (Gestion bus I2C)
- LiquidCrystal_I2C.h (Gestion LCD textuels)
- Password.h (Gestion des mots de passe)
- Pi-rc522 (Gestion des lecteurs RFID)

d) Documentation étudiée :

- MSDN – Microsoft Développeur Network (ressources pour le développement en C# sous Visual Studio)

VI. Analyse :



Les diagrammes UML qui sont présentés dans cette partie ne sont pas à jour. En effet, nous ne disposons pas de licence pour utiliser le logiciel MagicDraw depuis chez nous pendant le confinement. Nous vous prions de nous excuser.

a) Tâches réparties :

Le projet est décomposé en trois tâches distinctes. Vous trouverez, ci-dessous, la répartition de notre travail :

Étudiant 1 : Sous-système médaillon & détection de l'ordre de fin de partie

- ✓ Concevoir le sous-système médaillons : Récupérer UID, allumer ou éteindre les LED en fonction des lecteurs, ouvrir la gâche électrique et envoyer l'ordre de fin de partie.
- ✓ Dans l'application de supervision détecter l'ordre de fin de partie émis par le sous-système médaillon.

Étudiant 2 : Application web & mécanisme énigme musique

- ✓ Gérer le mécanisme énigme musique : Détecter le code numérique, afficher l'énigme, jouer la musique.
- ✓ Développer l'application web : Se connecter, gérer des comptes superviseurs, gérer des créneaux, réserver un créneau.
- ✓ Créer une base de données permettant la gestion des créneaux et des comptes superviseurs.

Étudiant 3 : Application de supervision & configuration du réseau

- ✓ Établir une schématisation du réseau.
- ✓ Étudier les protocoles réseaux pour s'assurer de leur compatibilité.
- ✓ Choisir un environnement de développement et un langage adapté au besoin.
- ✓ Développer l'application de supervision en veillant au respect du cahier des charges.

b) Diagramme d'exigences :

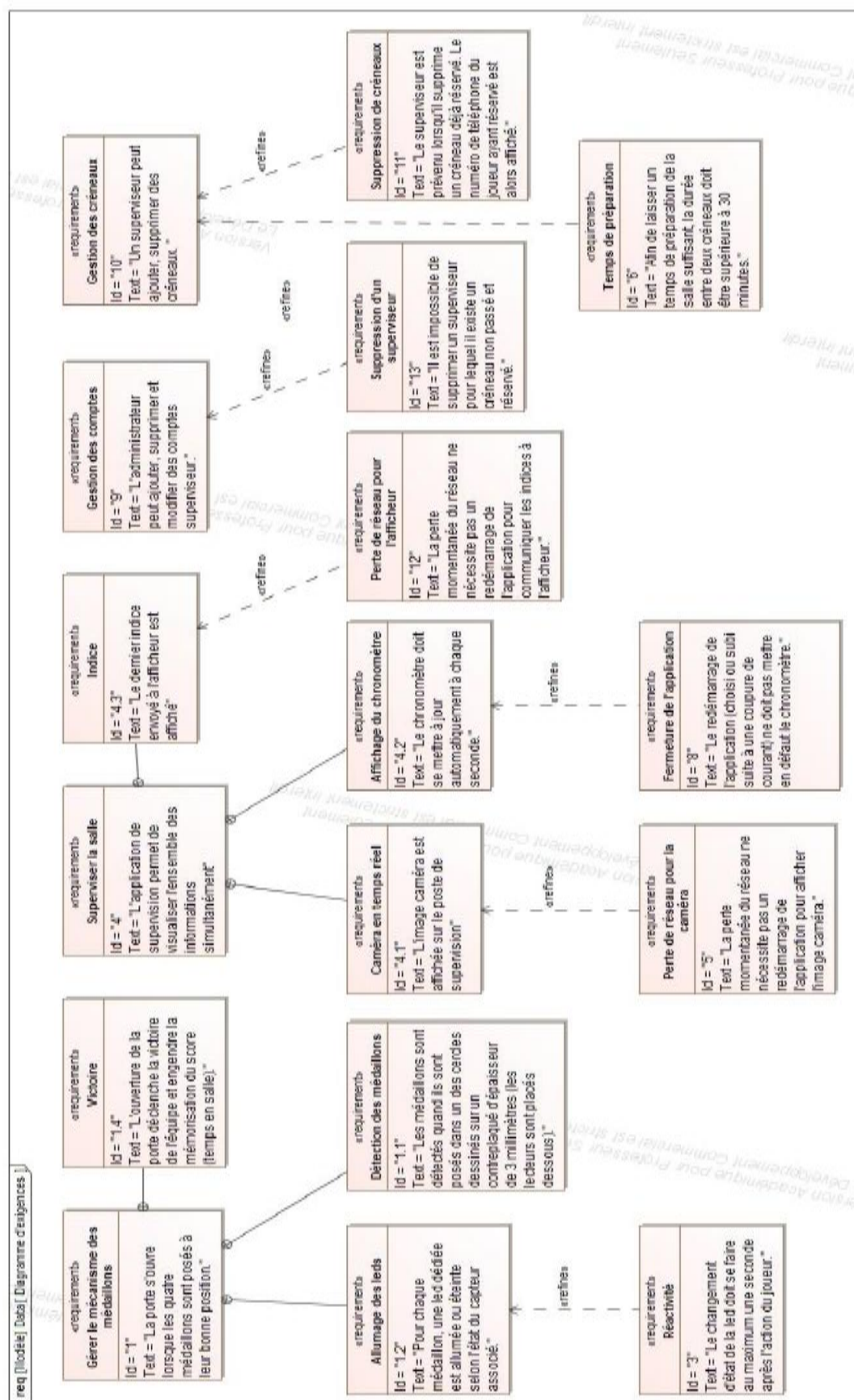


Image 4 : Diagrammes d'exigences du projet

c) Diagramme de cas d'utilisation :

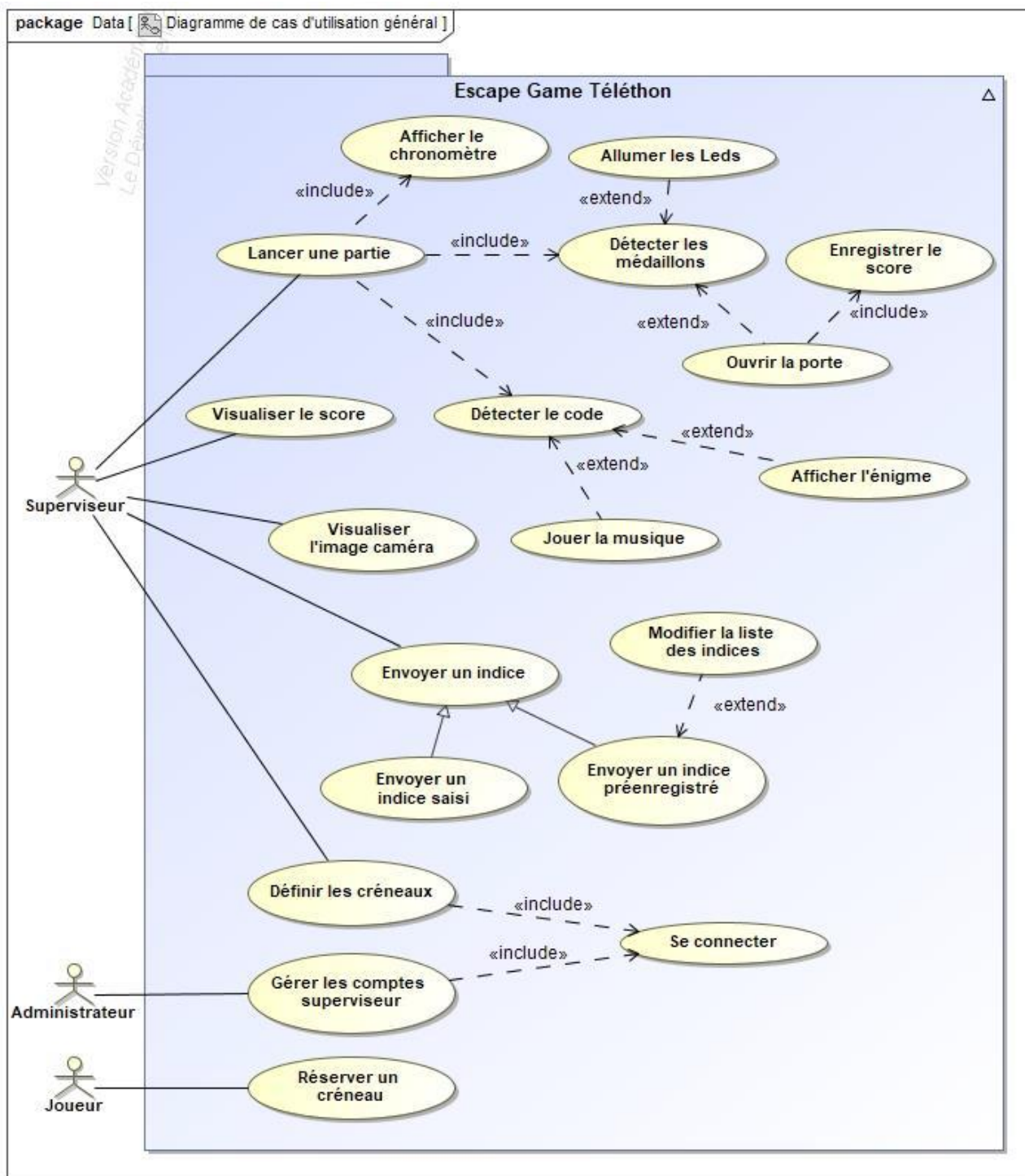


Image 5 : Diagrammes de cas d'utilisation du projet

d) Diagrammes de séquences :

Étudiant 1 :

Mise en situation 1 : Médaillon 1

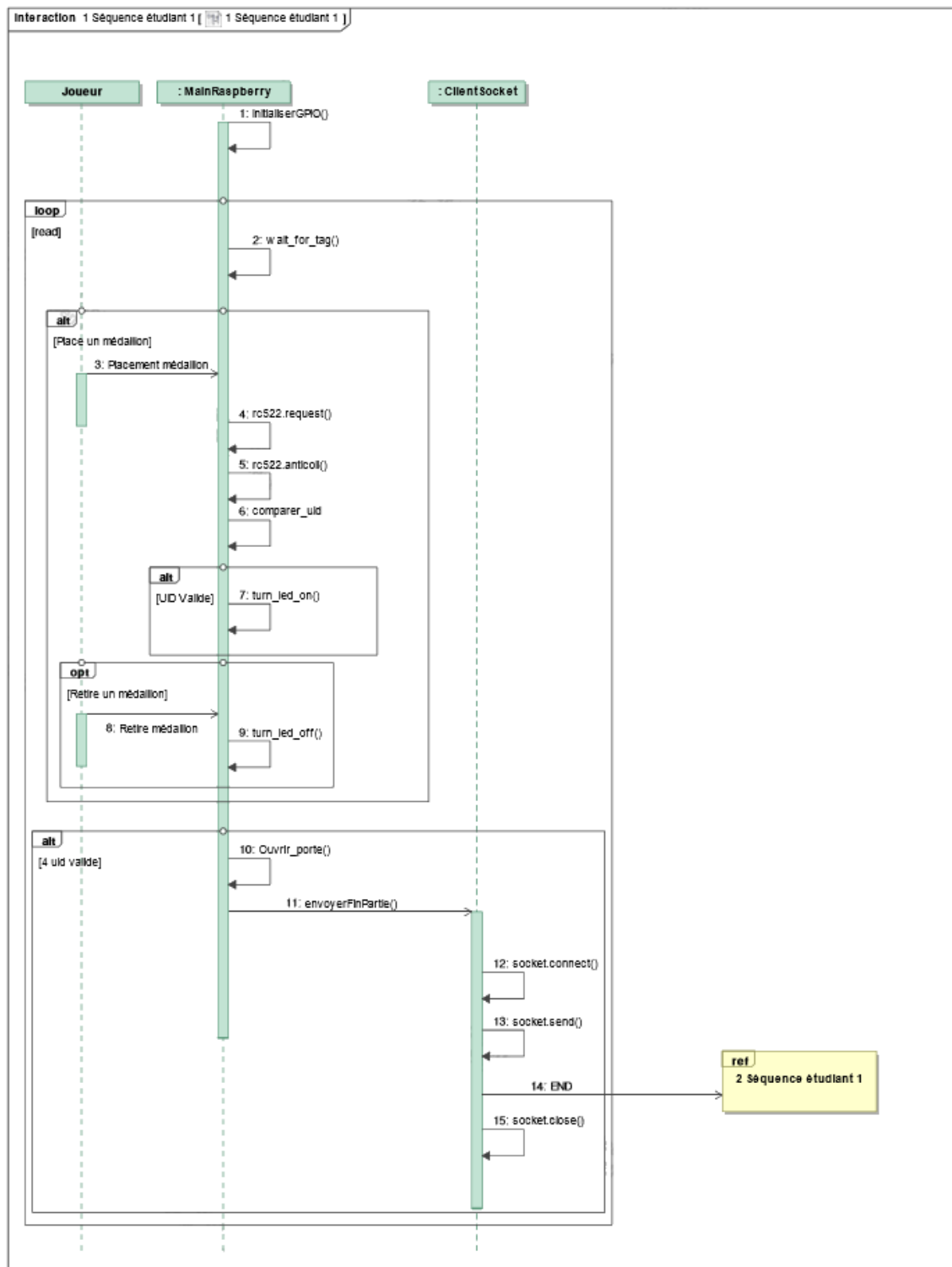


Image 6 : Diagramme de séquence du comportement du sous-système médaillon 1

Mise en situation 2 : Serveur socket asynchrone

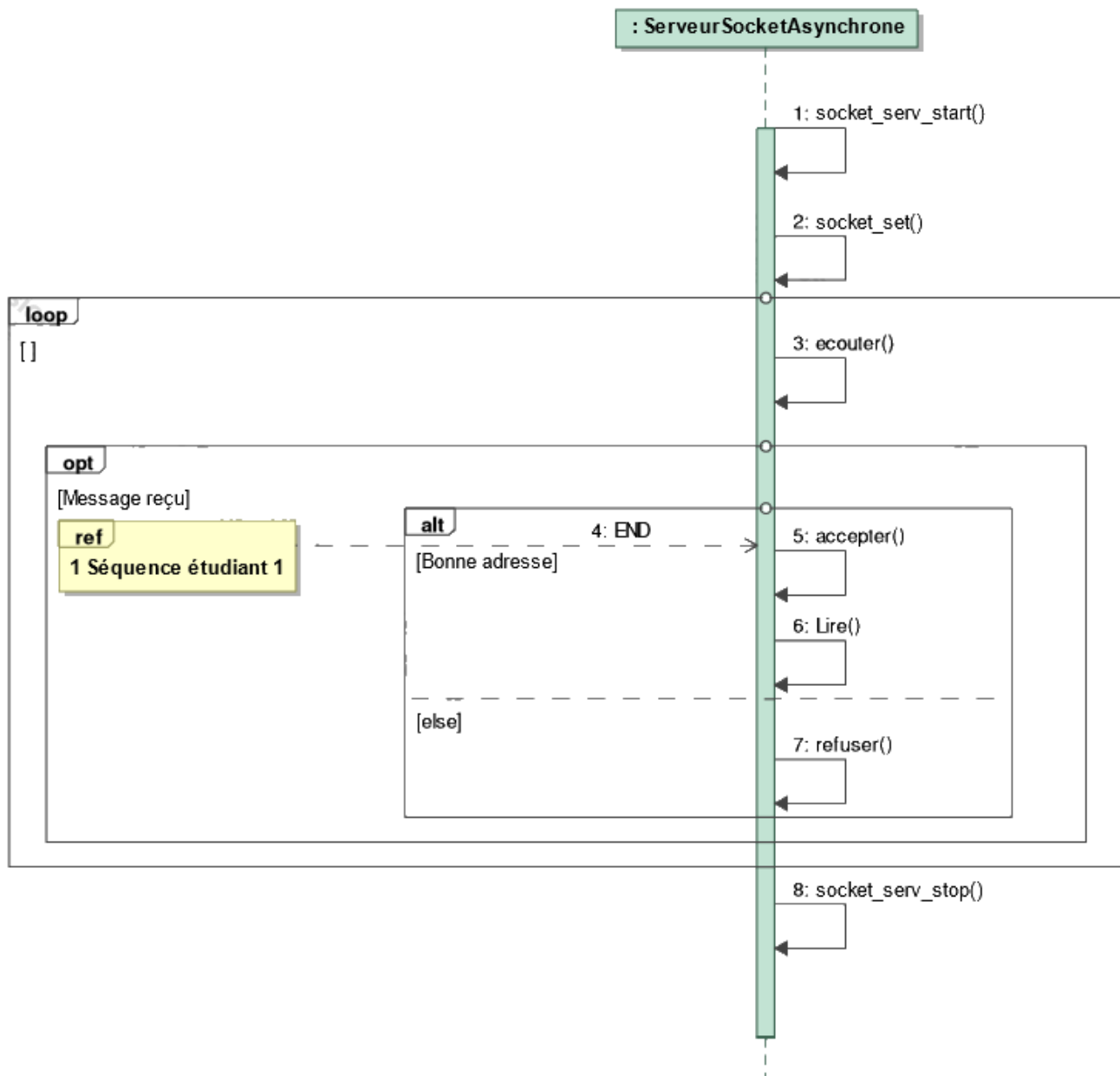


Image 7 : Diagramme de séquence du serveur socket asynchrone

Étudiant 2 :

Mise en situation 3 : Réservation d'un créneau

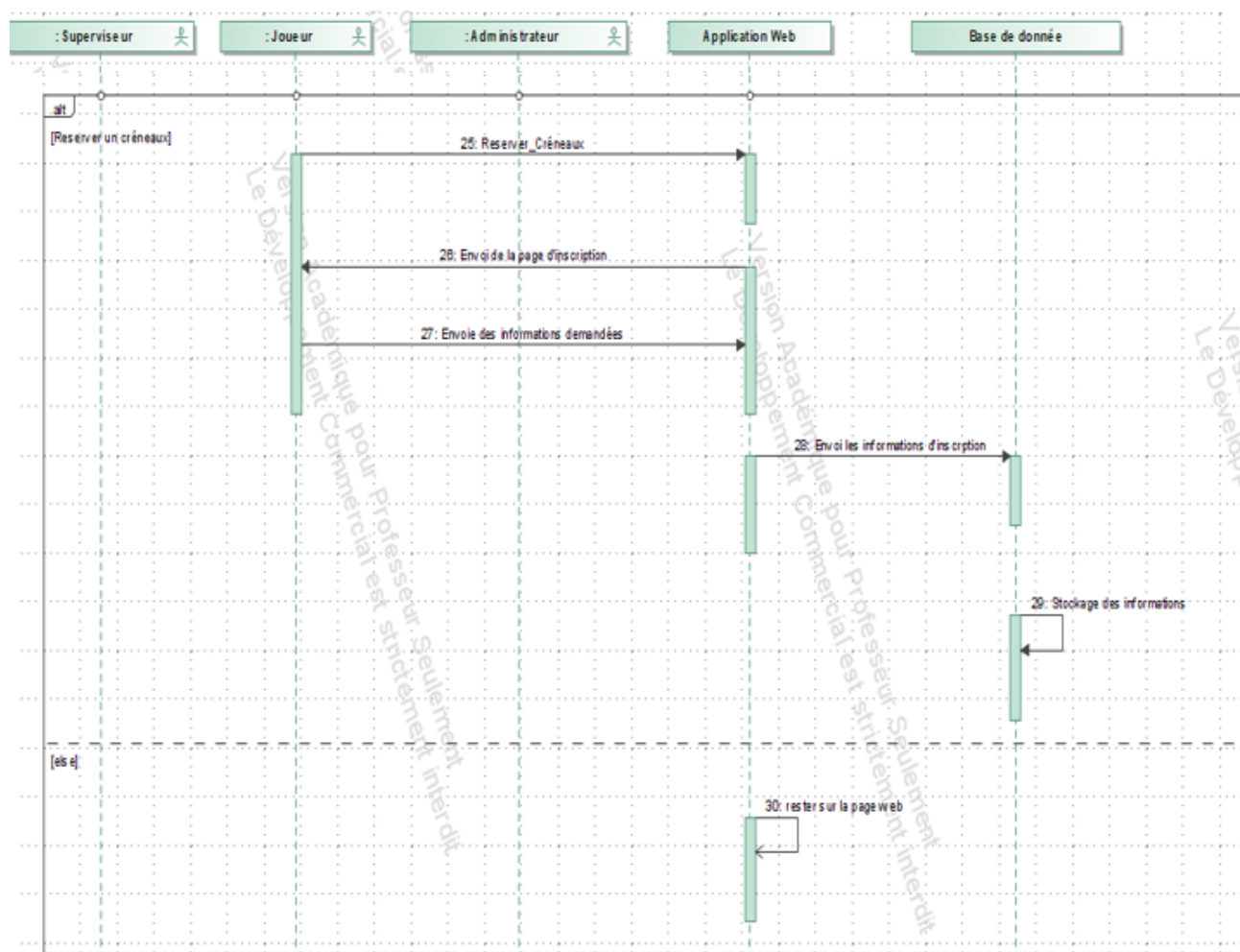


Image 8 : Diagramme de séquence de la réservation d'un créneau

Mise en situation 4 : Gestion d'un créneau

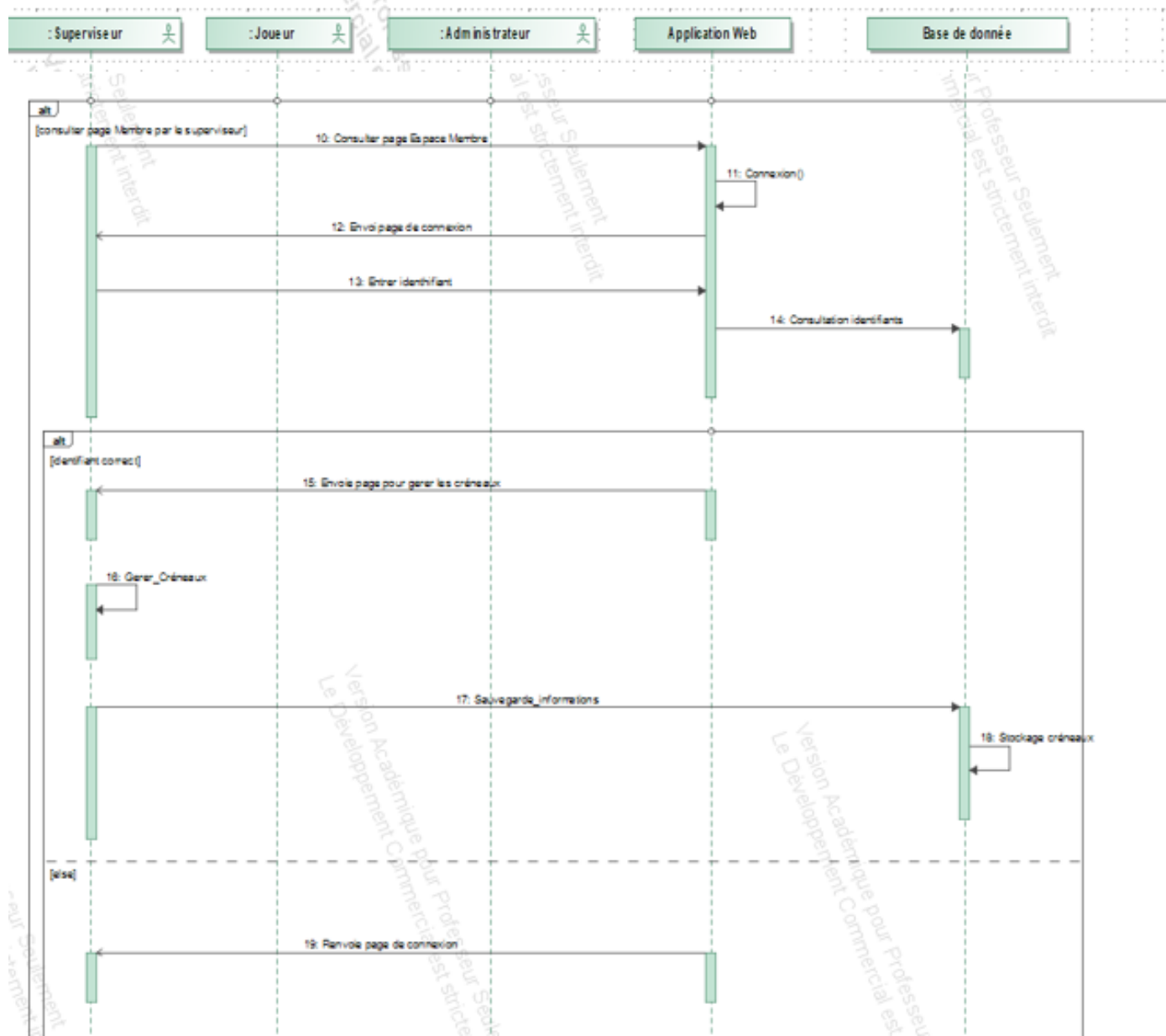


Image 9 : Diagramme de séquence de la gestion d'un créneau

Mise en situation 5 : Gestion d'un compte superviseur

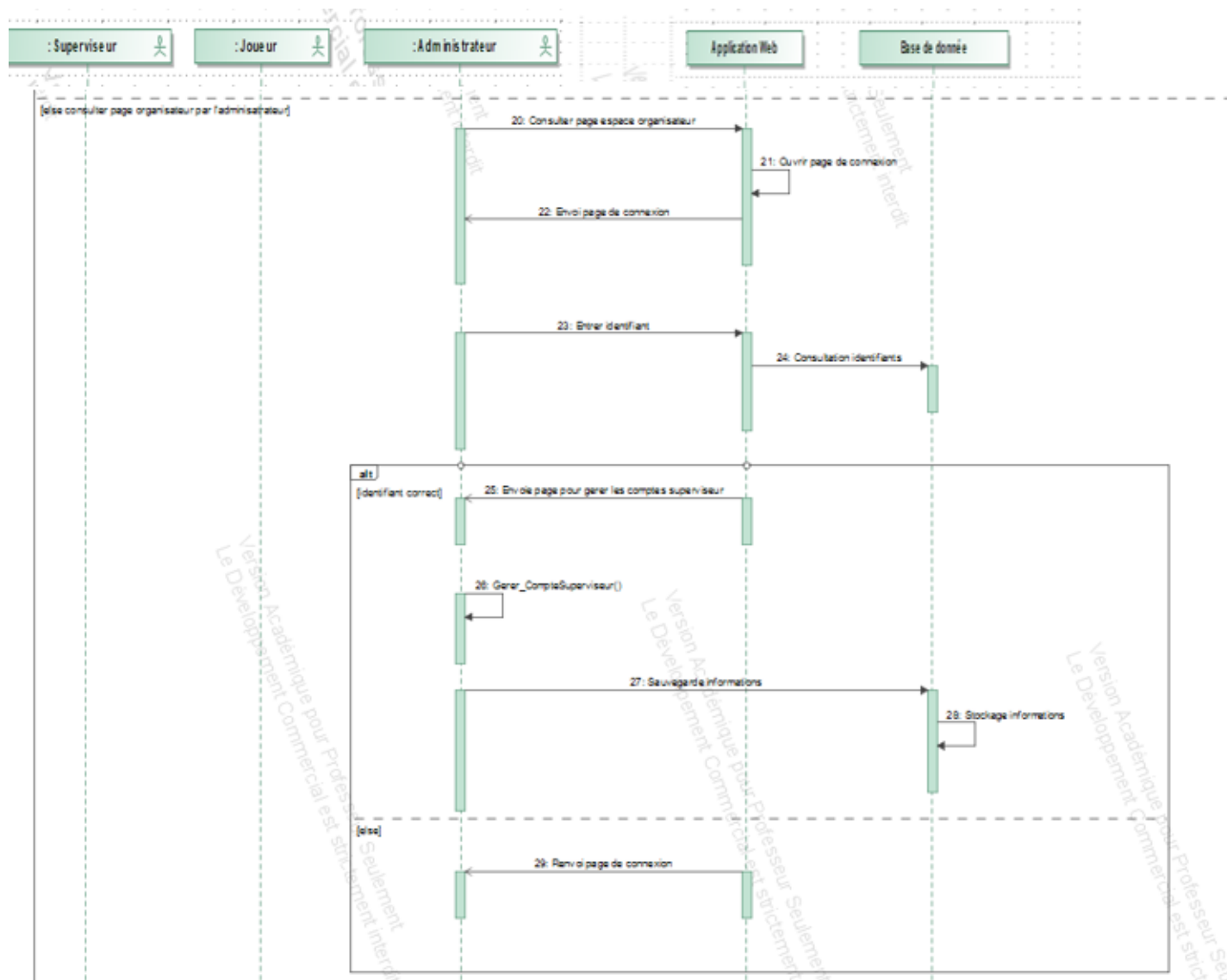


Image 10 : Diagramme de séquence de la gestion d'un compte superviseur

Mis en situation 6 : Enigme musique

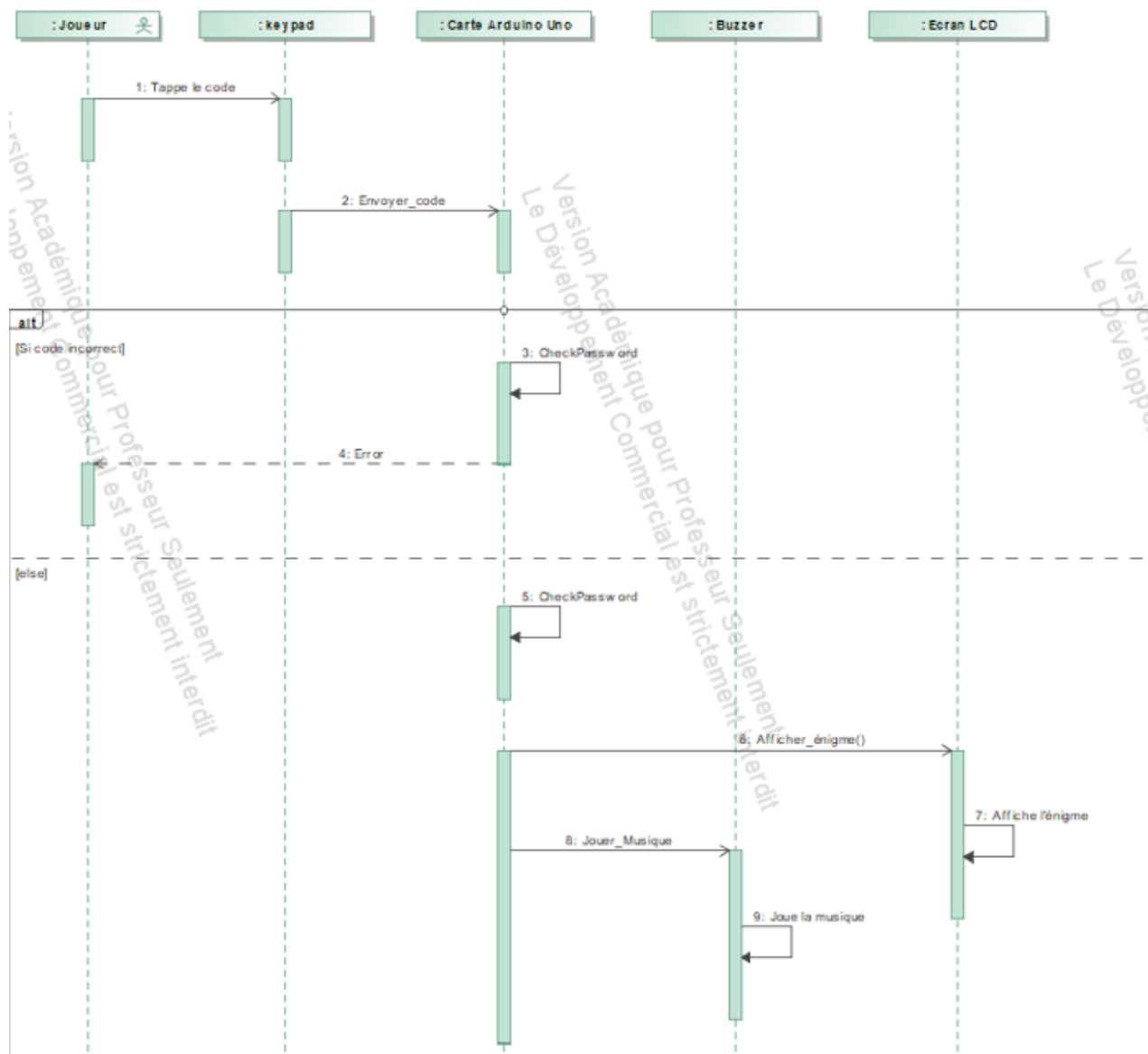


Image 11 : Diagramme de séquence de la gestion d'une énigme musique

Étudiant 3 :

 Mise en situation 7 : Lancement d'une partie

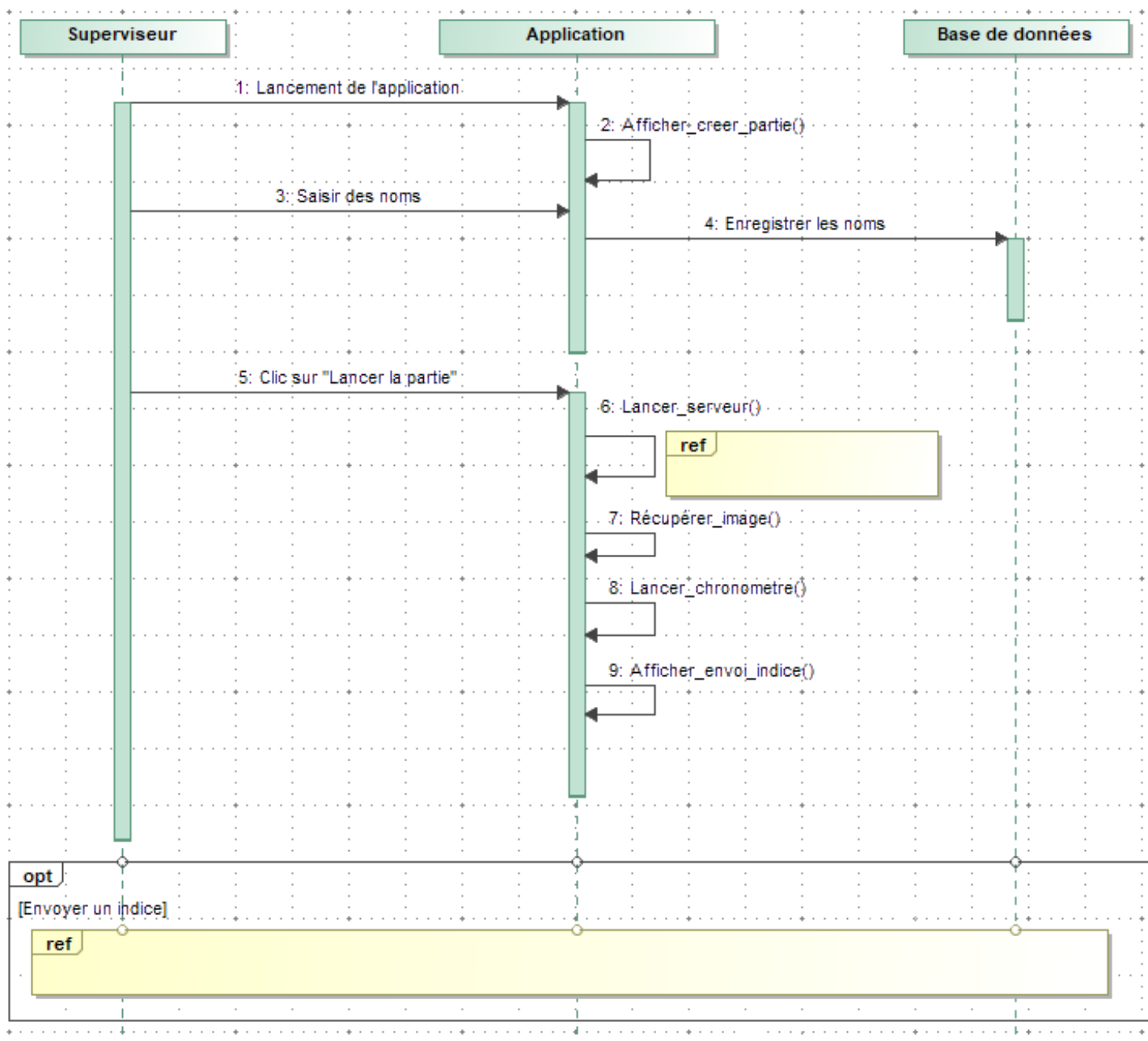


Image 12 : Diagramme de séquence du lancement d'une nouvelle partie

Mise en situation 8 : Envoi d'un message à l'afficheur

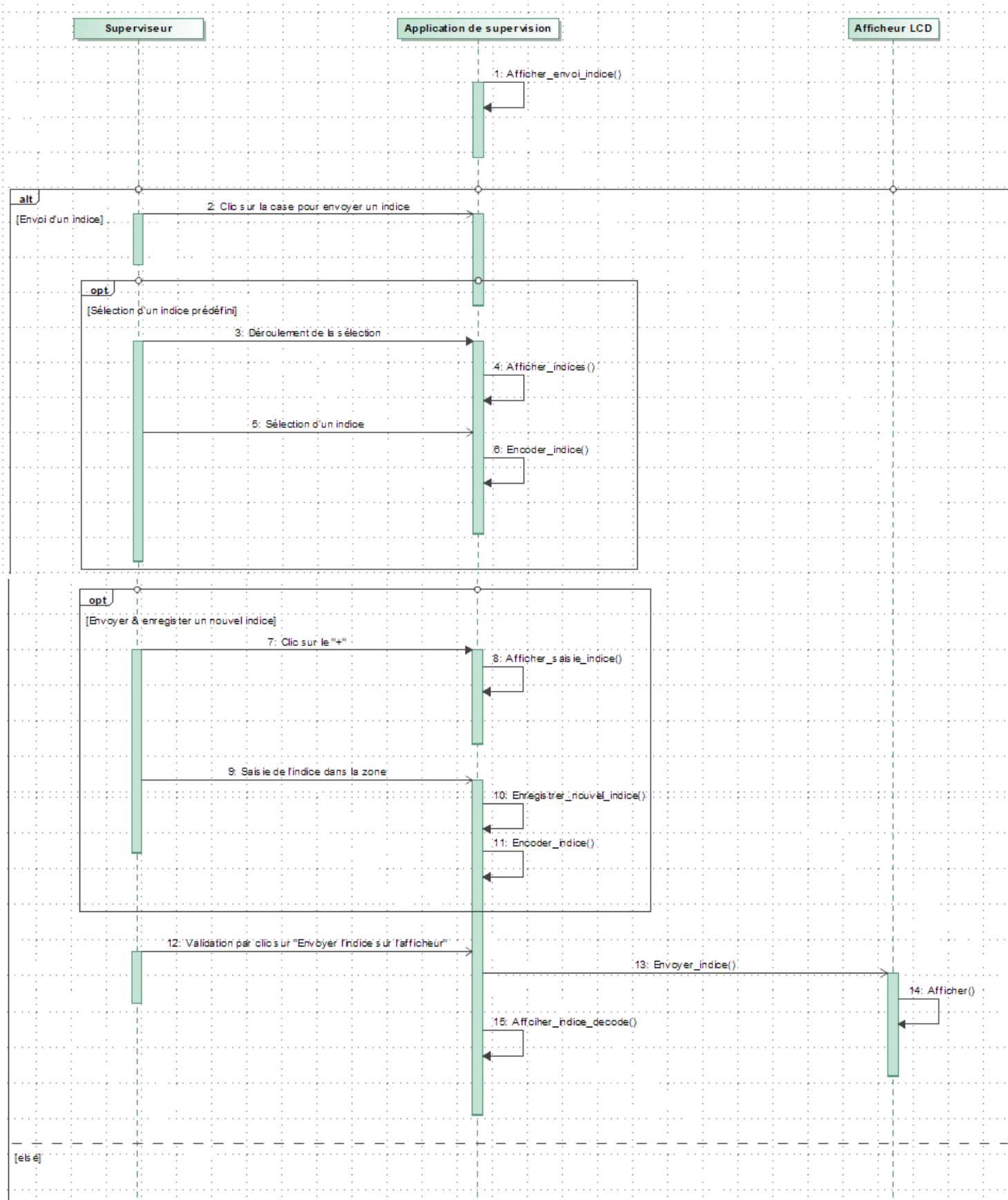
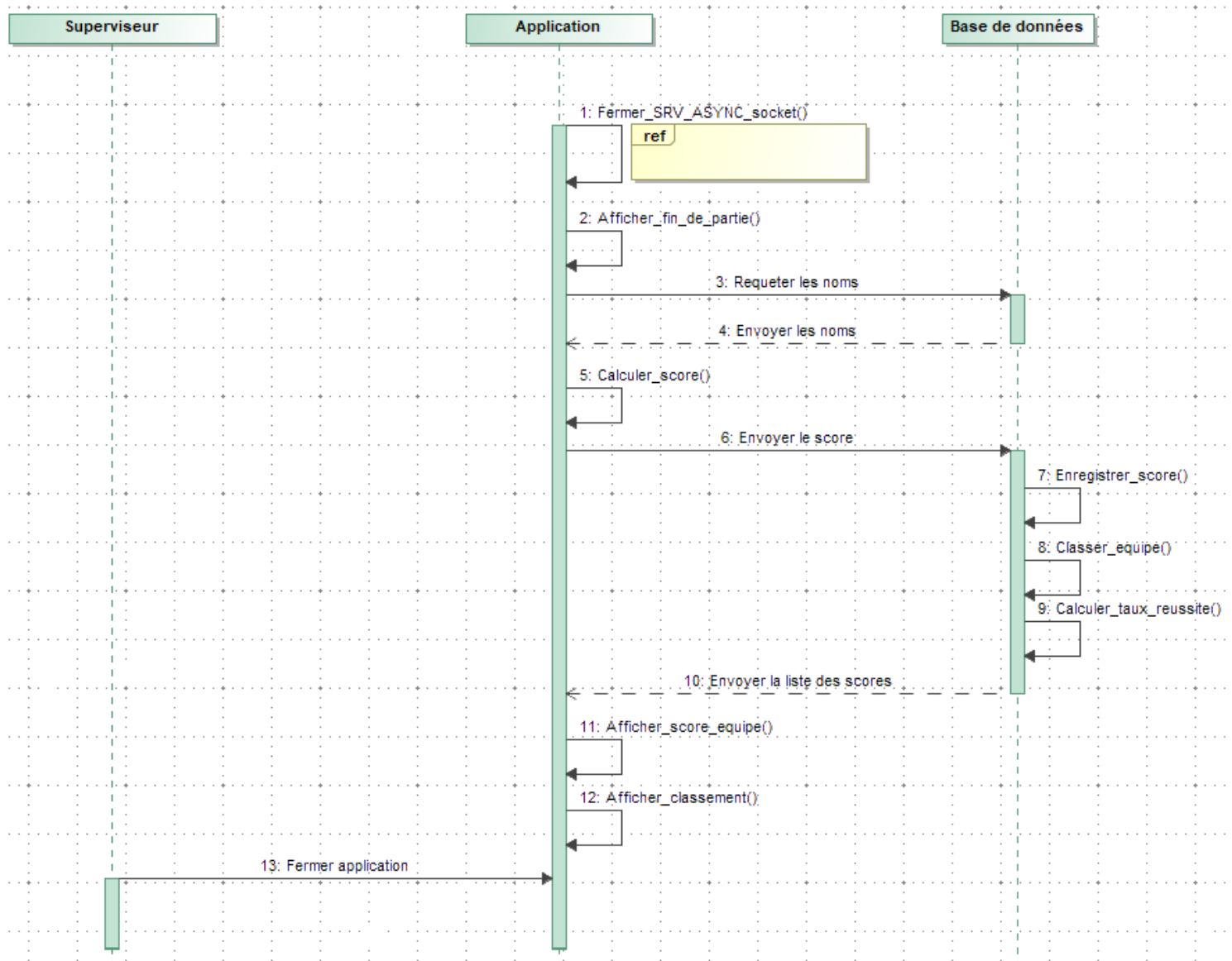


Image 13 : Diagramme de séquence de l'affichage d'un indice en salle

Mise en situation 9 : Fin de la partie*Image 14 : Diagramme de séquence de la fin d'une partie*

e) Diagrammes de classes :

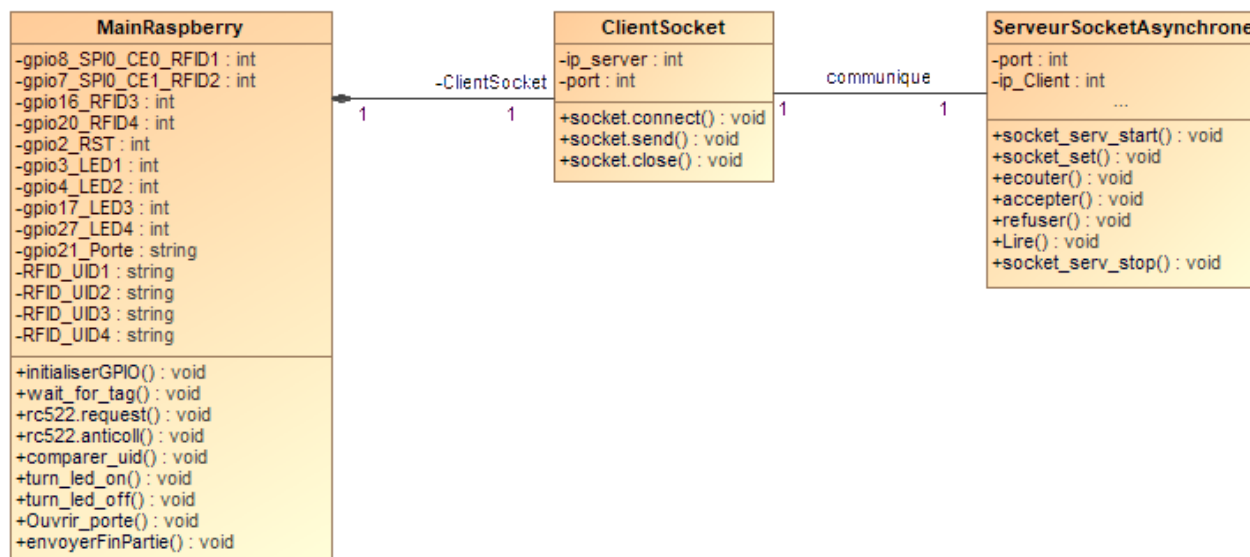
Sous-système médailles :

Image 15 : Diagramme de classes du sous-système médailles

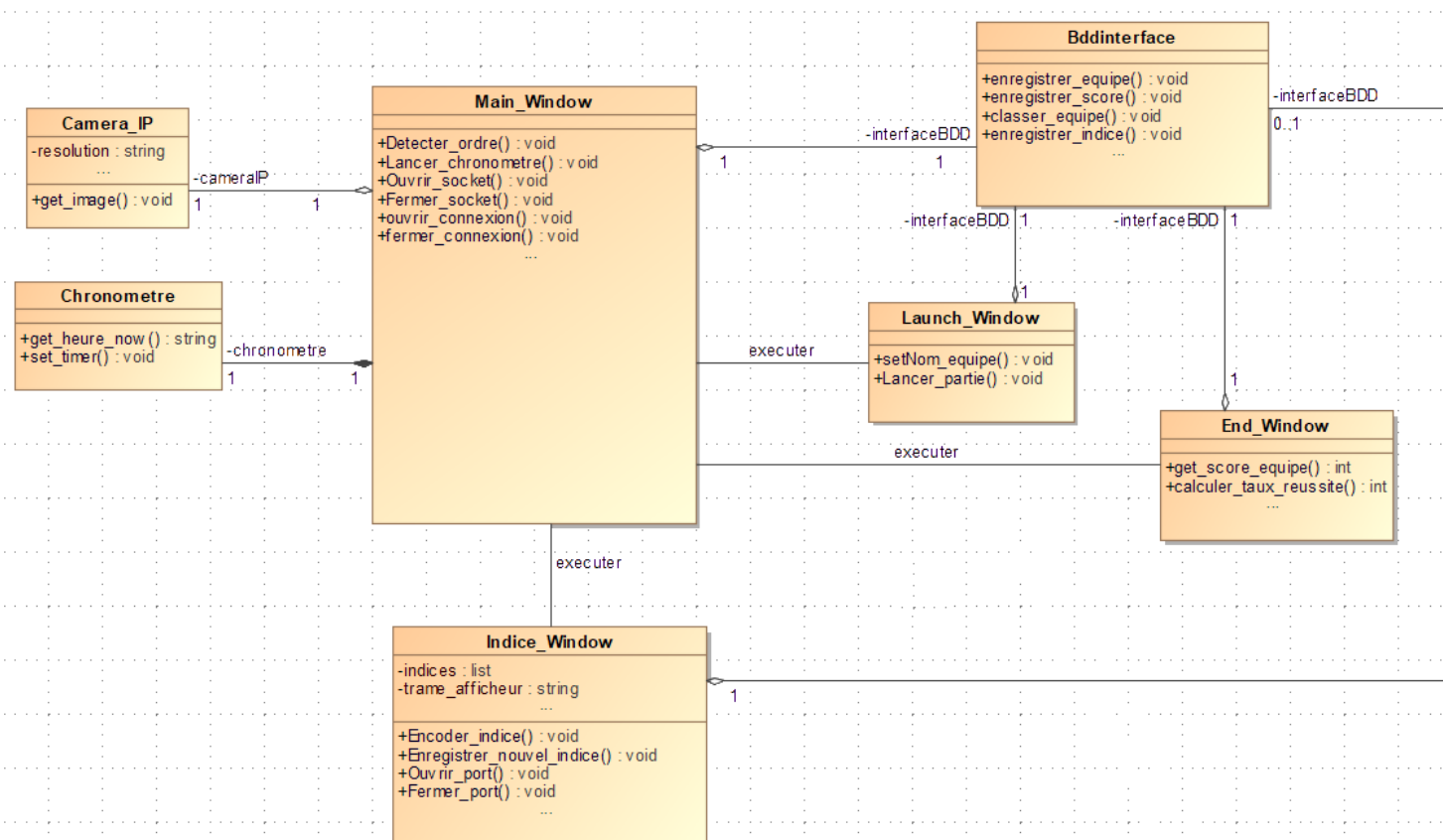
Application de supervision :

Image 16 : Diagramme de classes de l'application de supervision

Sous-système énigme musique :

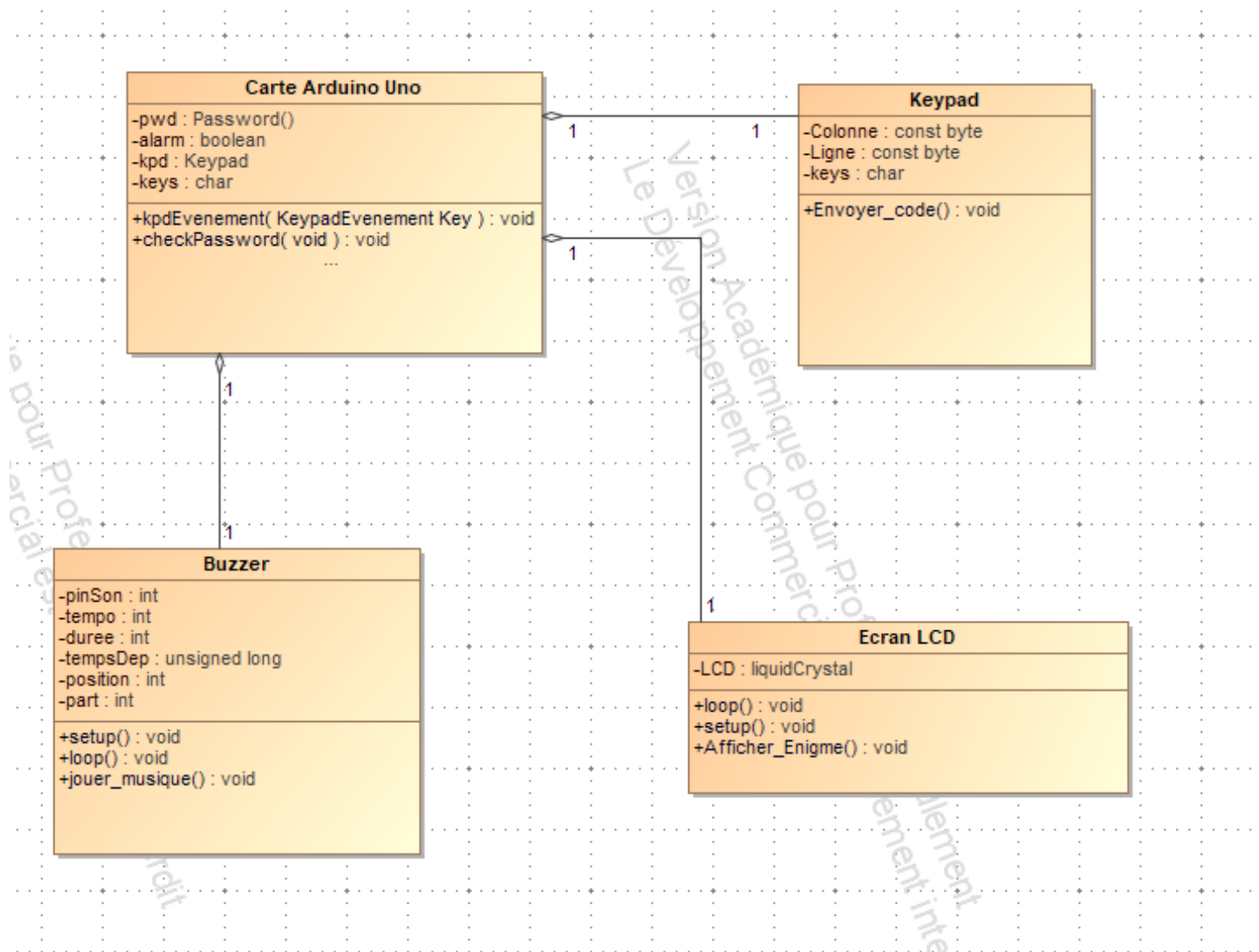


Image 17 : Diagramme de classes du sous-système énigme musique

f) Modèle relationnel de la base de données de l'application Web :



Image 18 : Modèle relationnel de la base de données de l'application web