Gestion Convecteur Crous

Table des matières

[Partie 1 : Présentation commune du projet 2](#_Toc72140632)

[Présentation du projet 2](#_Toc72140633)

[Le but du projet 3](#_Toc72140634)

[Principe de réalisation du projet 3](#_Toc72140635)

[Synoptique simplifié du système 3](#_Toc72140636)

[Synoptique simplifié du boitier de régulation autonome 4](#_Toc72140637)

[Diagramme de cas d’utilisationsimplifié 4](#_Toc72140638)

[Diagramme d’exigence 5](#_Toc72140639)

[Diagramme de classe 6](#_Toc72140640)

[Diagramme de déploiement 6](#_Toc72140641)

[MCD 7](#_Toc72140642)

[Organisation du projet 7](#_Toc72140643)

[Partage des ressources et méthodes du projet 7](#_Toc72140644)

[Gantt prévisionnel 8](#_Toc72140645)

[Gantt réel 8](#_Toc72140646)

[Répartition des taches 9](#_Toc72140647)

[Partie 2 : Partie individuel de valentin Bouet 10](#_Toc72140648)

[analyse personnel 10](#_Toc72140649)

[Cas d’utilisations 10](#_Toc72140650)

[organisation 10](#_Toc72140651)

[Suivie du projet 10](#_Toc72140652)

[Matériel 10](#_Toc72140653)

[Fonctions du système 10](#_Toc72140654)

[Partie 3 : Partie individuel de WANTELEZ FLORIAN 11](#_Toc72140655)

[ORGANISATION 11](#_Toc72140656)

[*Compte rendu d’activité* 11](#_Toc72140657)

[Cahier de bord 11](#_Toc72140658)

[*Vie du projet* 12](#_Toc72140659)

[Fonctionnalités du système 13](#_Toc72140660)

[Use case 13](#_Toc72140661)

[Cas d’utilisation 13](#_Toc72140662)

[Scenario 14](#_Toc72140663)

[Exigence 14](#_Toc72140664)

[Code 14](#_Toc72140665)

[Sequence 16](#_Toc72140666)

[Classe 17](#_Toc72140667)

[Test 18](#_Toc72140668)

# Partie 1 : Présentation commune du projet

## Présentation du projet

Le but de notre projet est de permettre au CROUS de pouvoir gérer les radiateurs électriques présents dans les appartements à distance.

Les deux grandes exigences de notre client sont de rendre autonome les radiateurs et de centraliser leurs gestions. Actuellement, de nombreux logements étudiants possèdent des radiateurs à convection électrique. Pour répondre aux besoins exprimés par le client, nous allons créer un boîtier connecté qui pourra se greffer aux radiateurs et gérera leurs utilisations en leurs imposant des consignes de chauffes.



## Le but du projet

Le but de notre projet est donc de répondre aux deux besoins principaux de notre client qui sont :

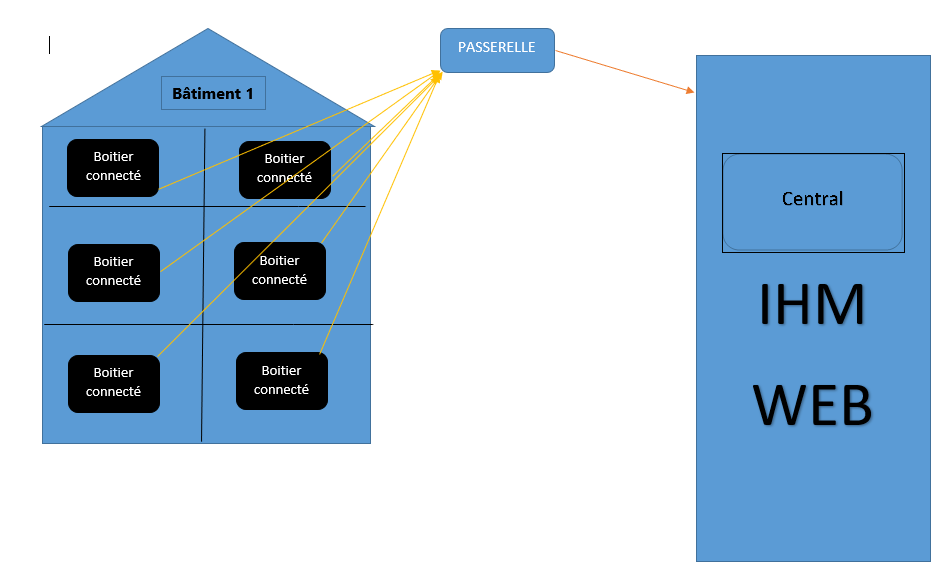
**1) Rendre autonome les radiateurs**

**2) Centraliser la gestion des radiateurs**

L’objectif final du projet est de réduire l’importante part de la consommation électrique que représente le chauffage de l’ensemble de ces logements étudiants en permettant au Crous de réguler leur utilisation.

## Principe de réalisation du projet

### Synoptique simplifié du système

****

- Chaque radiateur se verra équipé d’un boîtier qui pourra contrôler le radiateur.

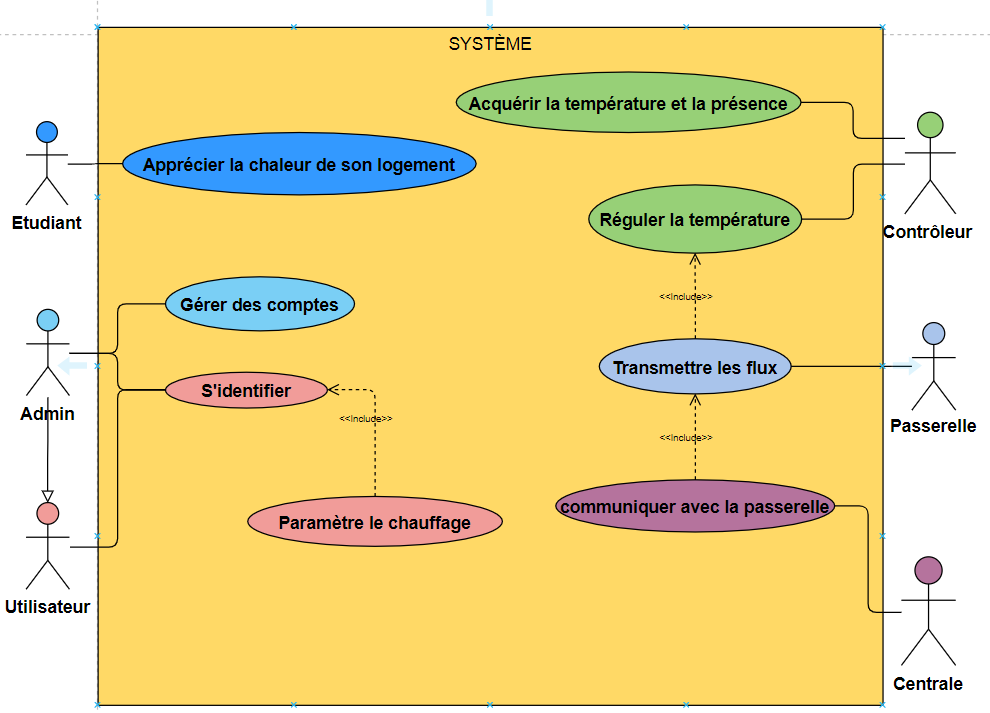
- Les boîtiers pourront communiquer avec la centrale présente aux bâtiments du CROUS via une passerelle de communication qui sera propre à chaque bâtiment de logement étudiant.

- Les opérateurs du Crous pourront voir et gérer chaque radiateur équipé d’un boîtier via un site web.

### Synoptique simplifié du boitier de régulation autonome



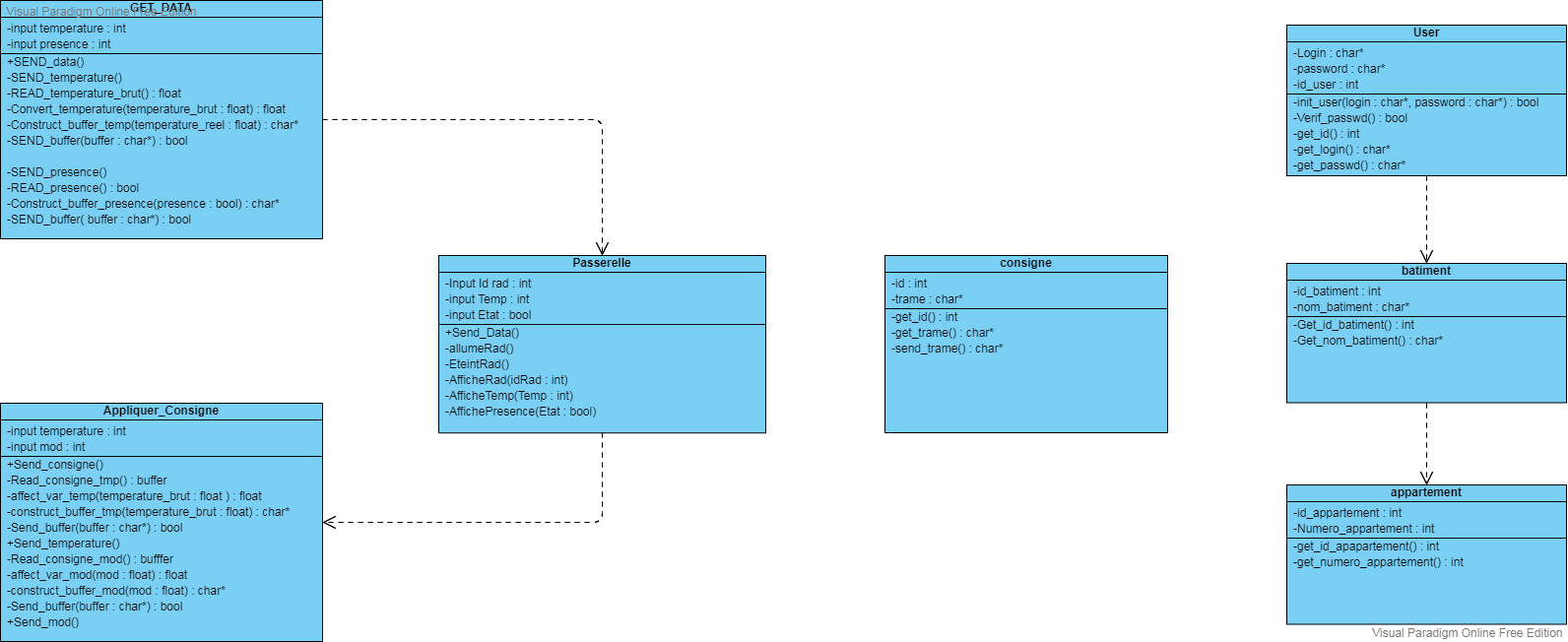
### Diagramme de cas d’utilisationsimplifié

**

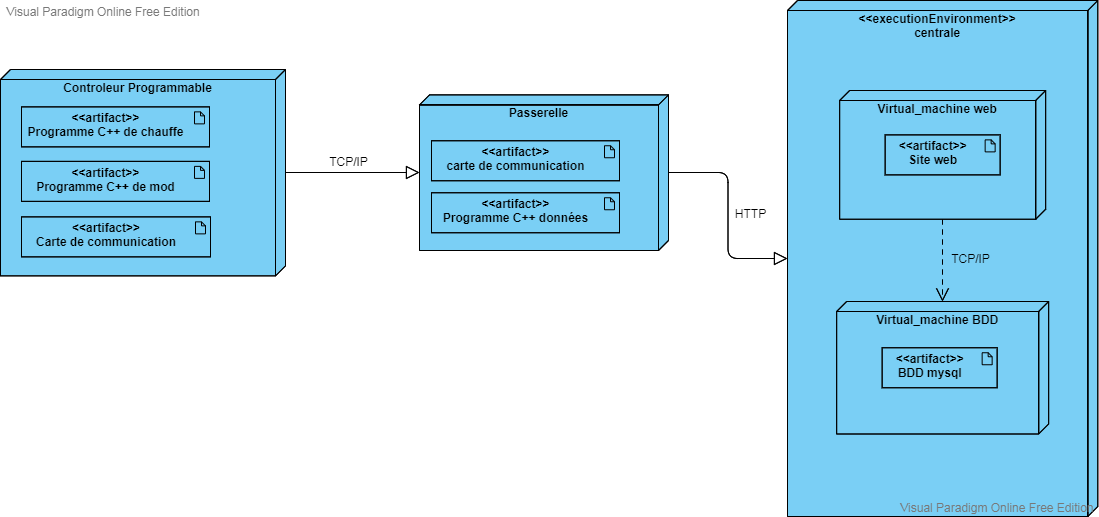
### Diagramme d’exigence



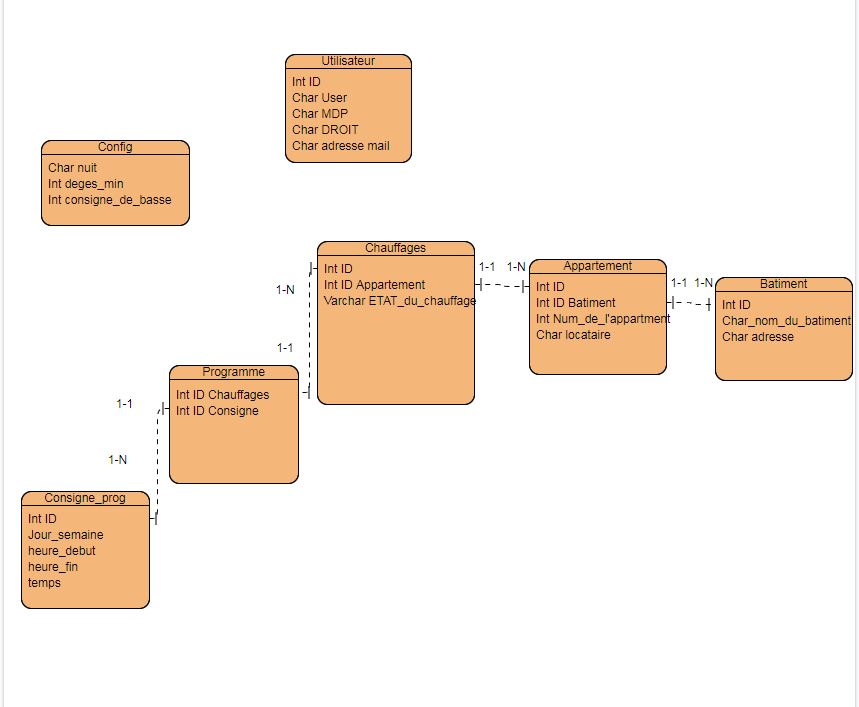
### Diagramme de classe



### Diagramme de déploiement

**

### MCD

**

## Organisation du projet

### Partage des ressources et méthodes du projet

Une Virtual Machine (ip : 192.168.65.113) a été mise en place pour partager les ressources, diagrammes SYSML, code C++ et web lié au projet.

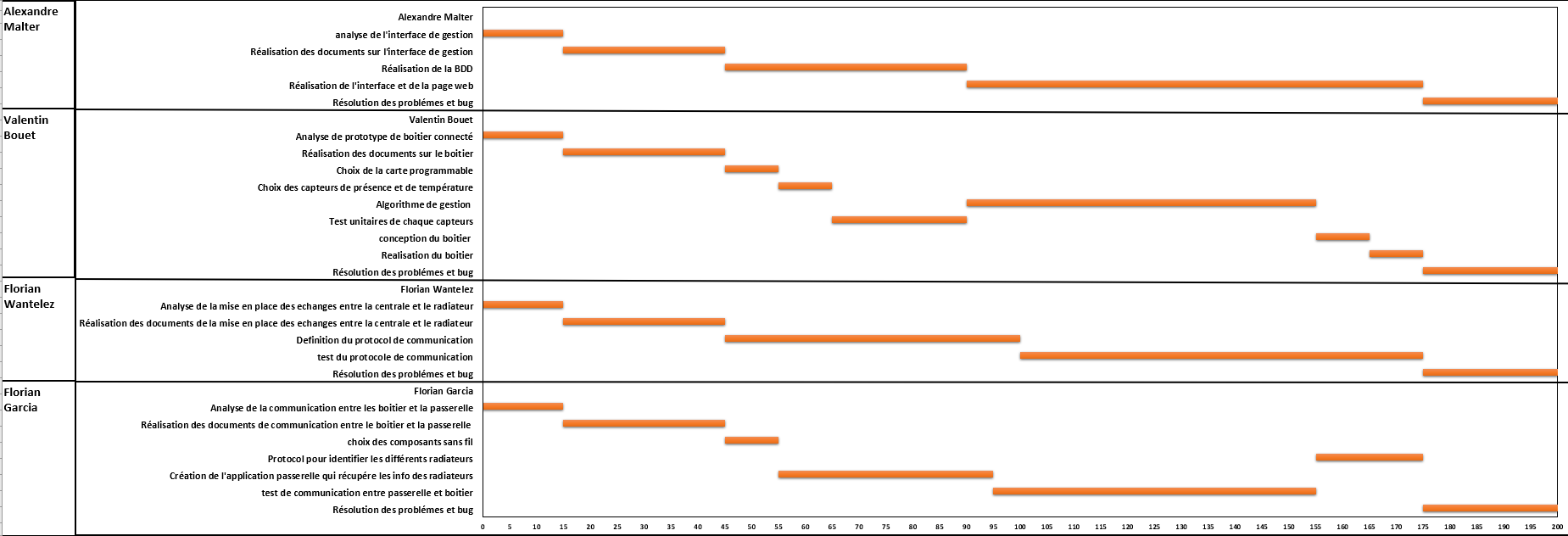
Deux Virtual machines permettent d’héberger la Base de Données sous mariaDB (ip : 192.168.64.155 et le site web (ip : 192.168.65.4).

Un dossier Google drive mets en commun tous les fichiers sources et SYSML du projet afin de pouvoir les récupérer à notre domicile.

Enfin, un répertoire GITHUB est mis en place pour partager les codes web et c++ et pour nous permettre de les récupérer sur n’importe quel ordinateur.



### Gantt prévisionnel



### Gantt réel

### Répartition des taches

Voici la répartition des tâches. Cette dernière nous est imposée par le sujet de BTS SN.

Etudiant 1 : Création de l’interface de gestion de pilotage centralisé en PHP/MYSQL

- Interface CRUD de Gestion des utilisateurs

- Interface CRUD de Gestion des radiateurs par Passerelles

- Interface CRUD de Gestion des Maisons (Passerelles)

- Création du MCD

- Page de Connexion

- Page d’affichage de toutes les informations des radiateurs

- Page de Paramétrage d’un Radiateurs

- Page de Paramétrage d’une Passerelle

Etudiant 2 : Création d’un prototype de boitier connecté autonome avec les composants. Régulation de température.

- Choix de la carte programmable

- Choix des capteurs de présence, de température, d’alimentation

- Tests unitaires de chaque Capteurs

- Algorithme de Gestion Autonome Reprenant les exigences de chauffes.

Etudiant 3 : Communication entre le boitier connecté et la passerelle (Sans fil) Communication entre la passerelle et l’interface de gestion (Via TCP/IP)

- Choix du composant sans fil de communication entre Boitier connecté et la passerelle

- Test de communications Bidirectionnel entre la passerelle et le boitier

- Mise en place d’un protocole pour identifier les différents radiateurs

- Création de l’application Passerelle qui récupère les informations des radiateurs pour les envoyer à l’application web de Centralisation.

- Test d’installation d’un Boitier et d’une passerelle en respectant les exigences.

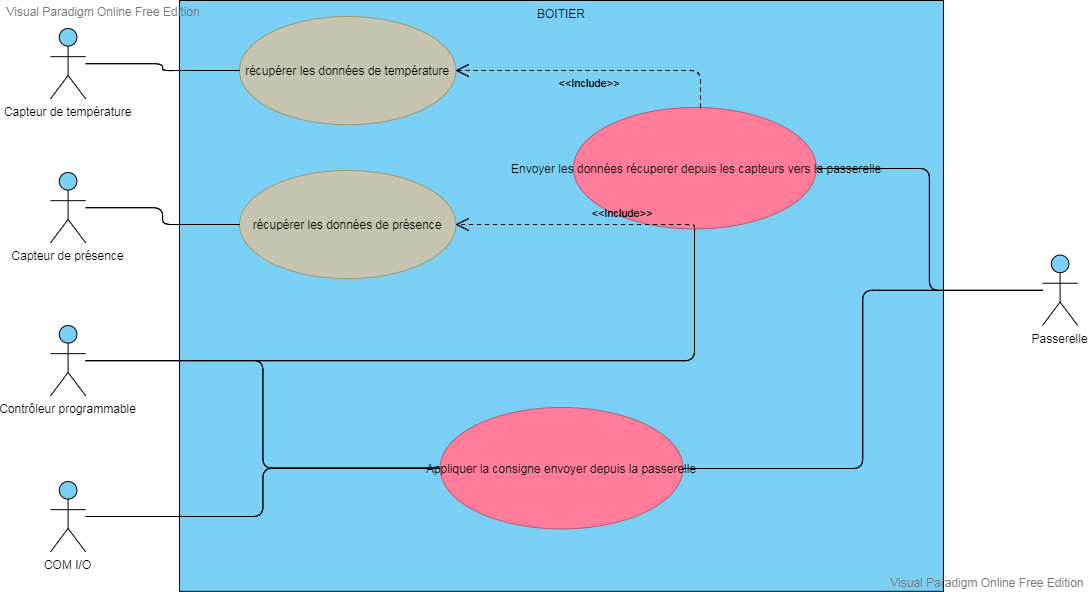
Etudiant 4 : Mise en place des échanges de commande entre la central et le radiateur. Mise en place du protocole de communication.

- Mise en place des échanges de messages entre Boitier et Central.

# Partie 2 : Partie individuel de valentin Bouet

## analyse personnel

### Cas d’utilisations

****

## organisation

### Suivie du projet

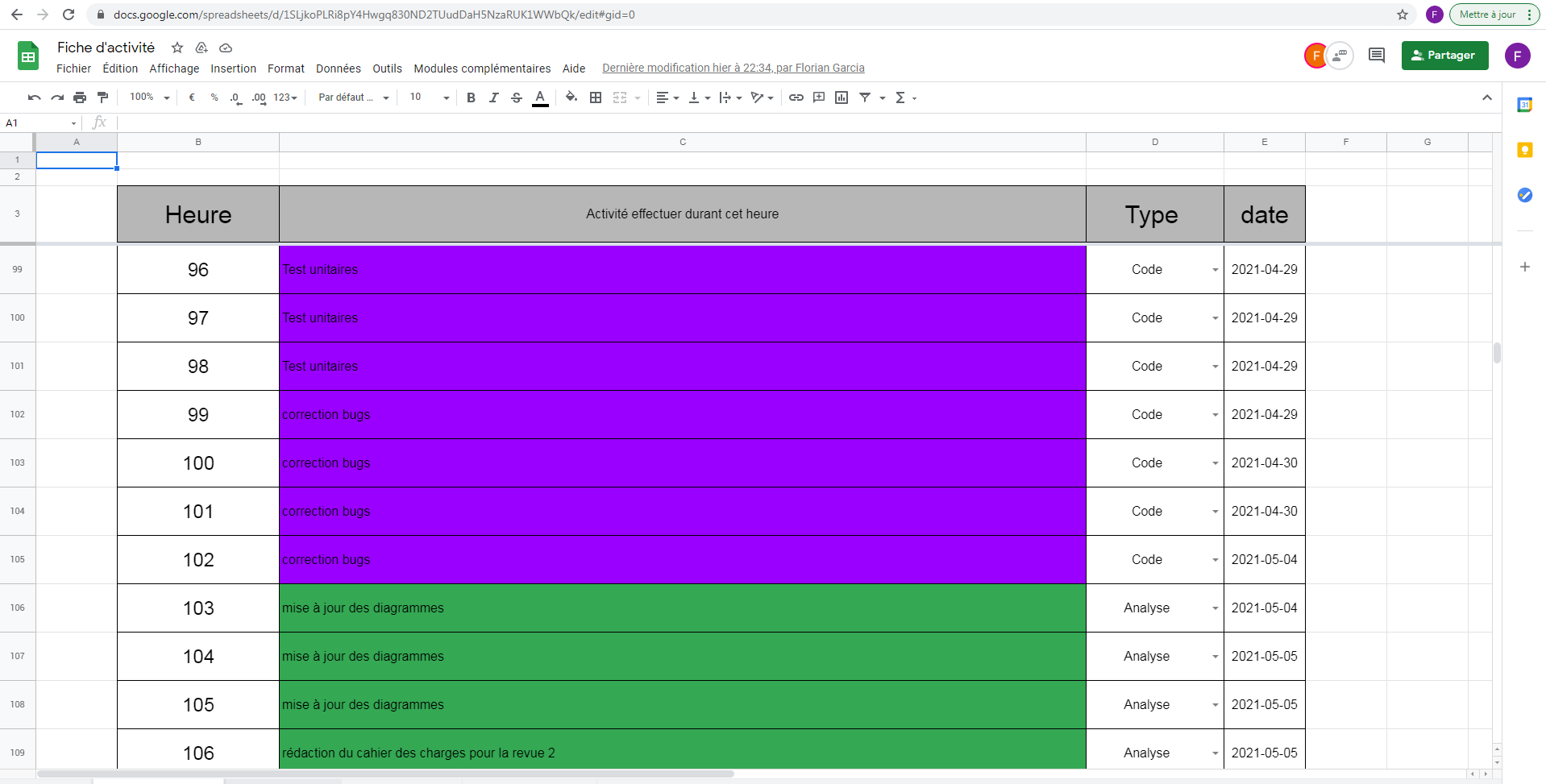
## Matériel

## Fonctions du système

# Partie 3 : Partie individuel de WANTELEZ FLORIAN

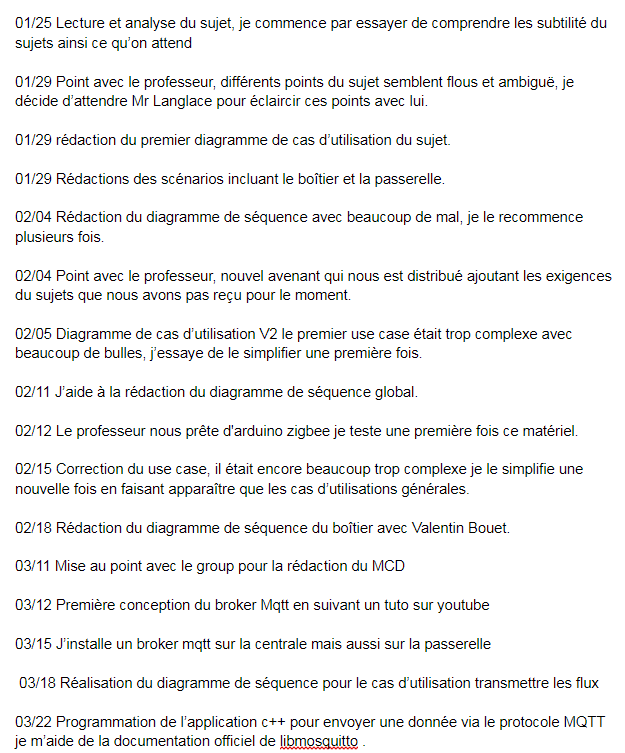
## ORGANISATION

### *Compte rendu d’activité*



Le compte rendu d’activité permet un suivi horaire de chaque tâche du projet, qui sont détaillés en 5 types : conception, code, analyse, prototypage, matériel avec un code couleur correspondant.

### Cahier de bord



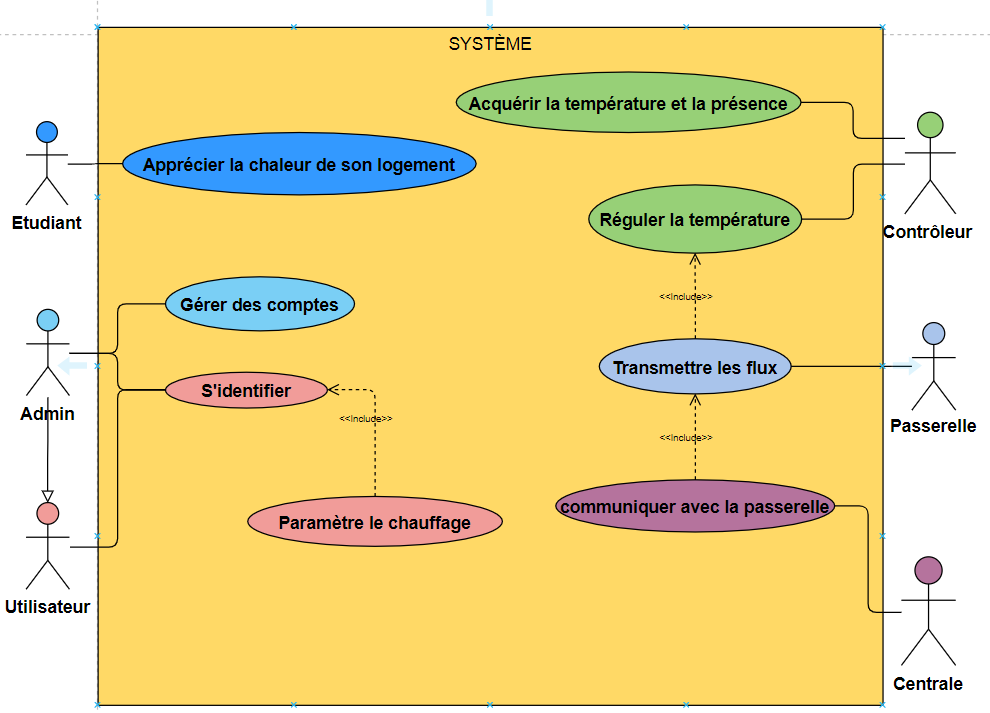
Pour le projet a été mis en place un cahier de bord personnel permettant de suivre les différentes étapes du projet, les problèmes rencontrés ainsi que les phases du développement.

### *Vie du projet*

La vie en projet s’est décomposé en … avec …. Pour …

## Fonctionnalités du système

### Use case



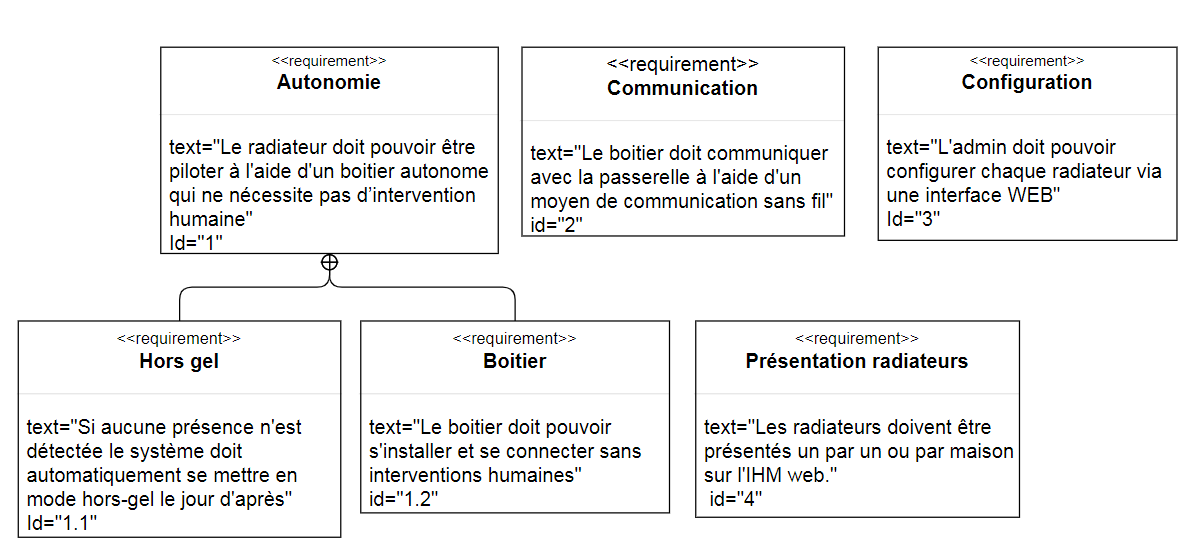
Mon diagramme de cas d’utilisation est composé de la fonctionnalité blablabla qui blablabla une fois qu’on a blablabla

### Cas d’utilisation

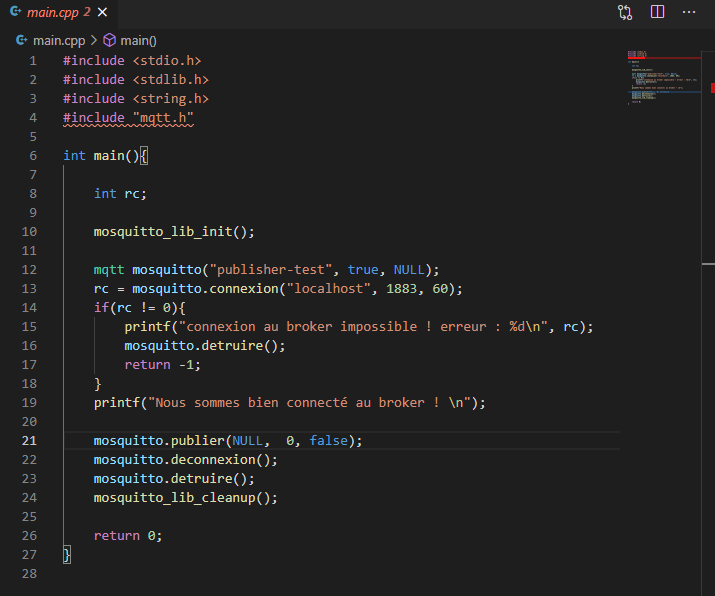
Cette fonctionnalité inclue 2 partie, une partie réception des messages et l’autre envoie des messages.

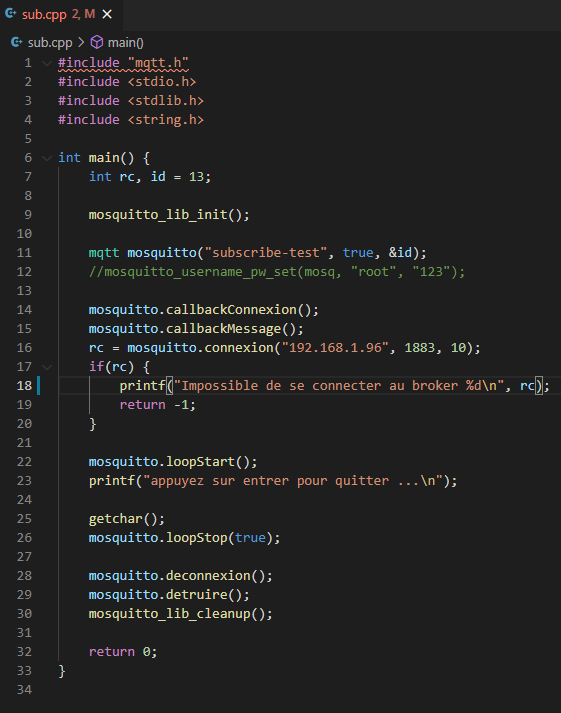
### Scenario

### Exigence



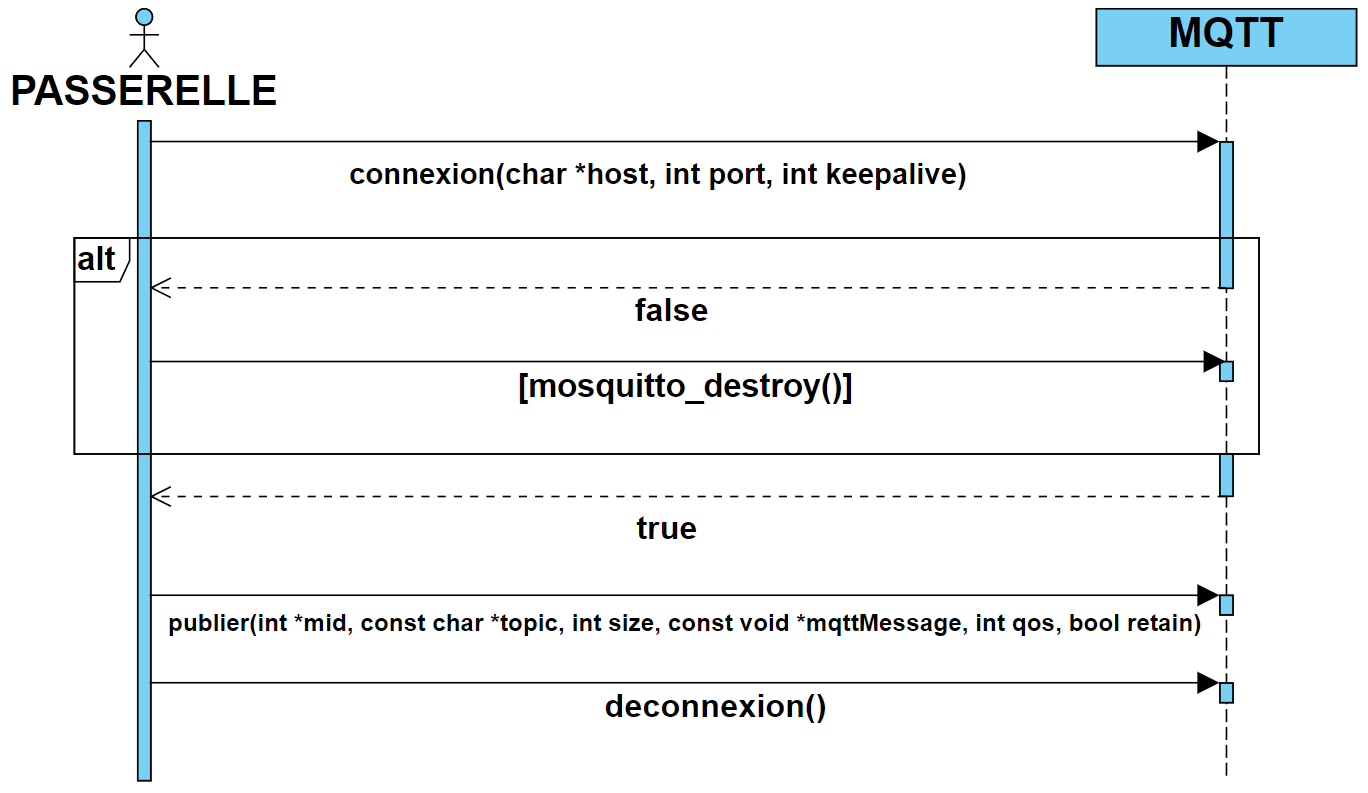
### Code

****

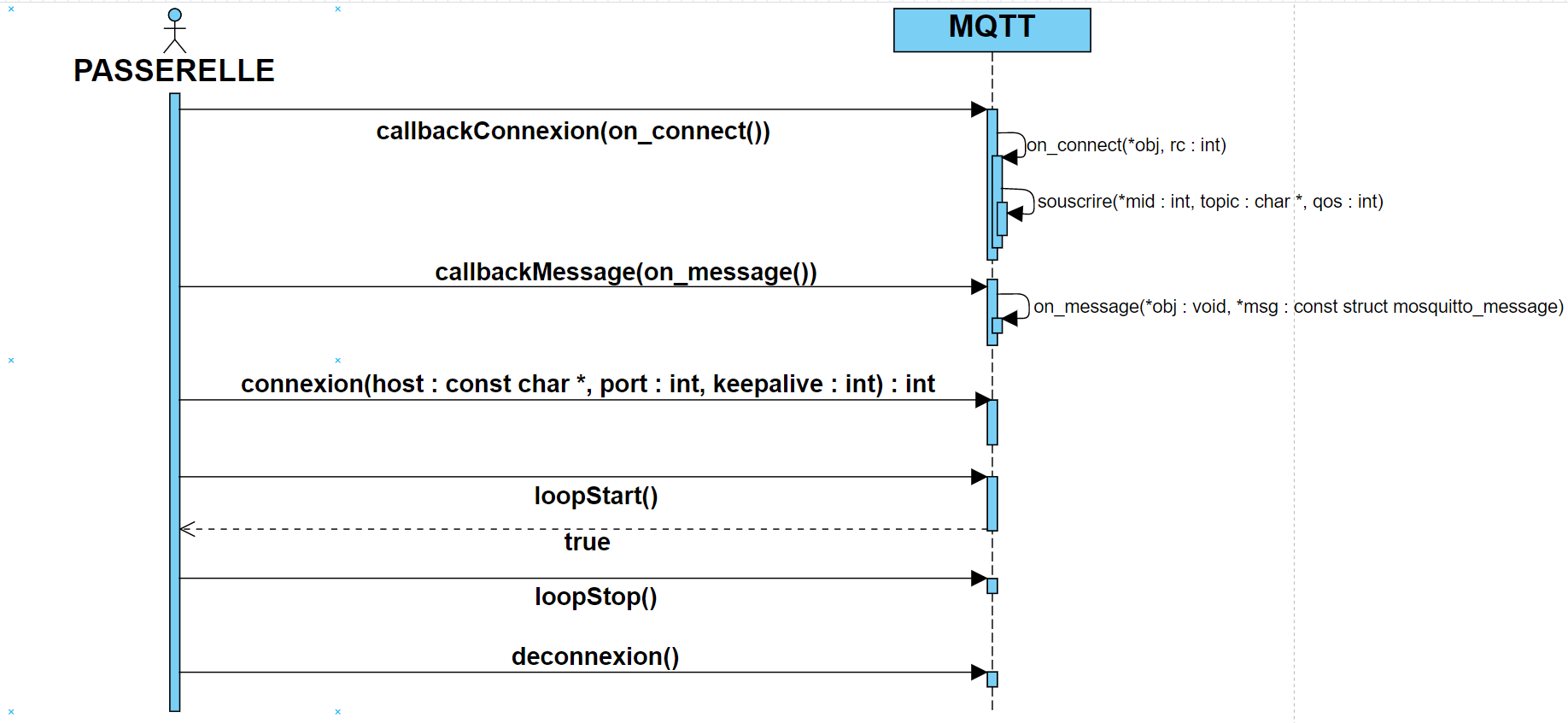


### Sequence

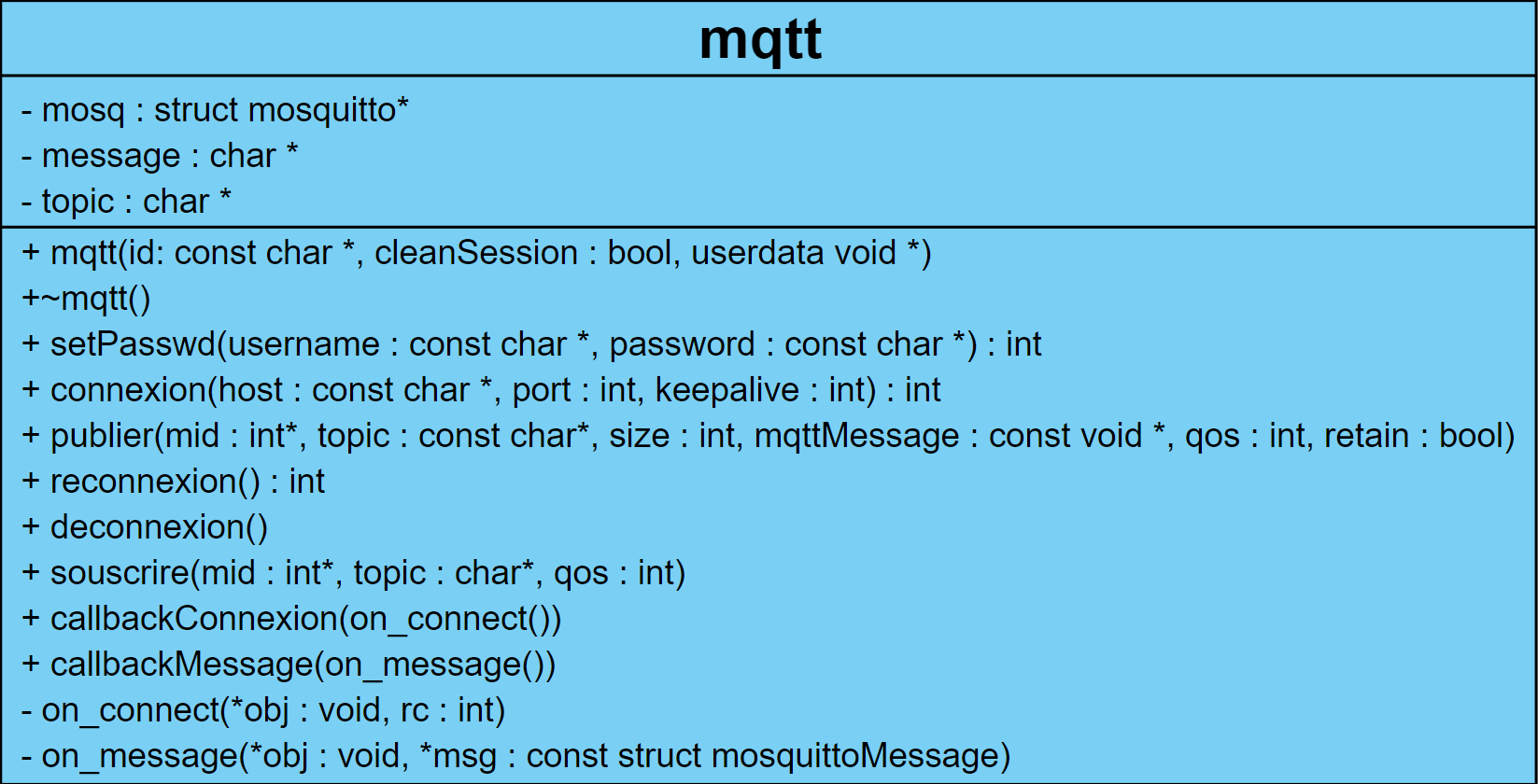
Envoyer un message à la centrale



Recevoir un message de la centrale

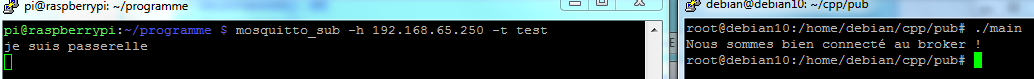


### Classe

****

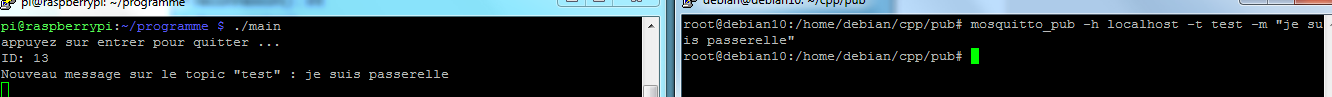
### Test

Envoyer un message à la centrale

****

Pour ce test j’utilise la fonction mosquitto\_sub disponible dans le paquet mosquitto (screen de gauche) pour écouter les messages MQTT reçus sur le topic « test » du broker puis je lance mon programme main.cpp avec la commande ./main qui va s’occuper d’envoyer lui-même le message prédéfinis, on peut voir qu’il arrive belle et bien à se connecter au broker et que le message est bien envoyé puisqu’on le reçoit dans la console de gauche

Recevoir un message de la passerelle

****

Pour ce test j’utilise la fonction mosquitto\_pub disponible dans le paquet mosquitto (screen de gauche) pour envoyer un message en MQTT sur le topic « test » du broker puis je lance mon programme sub.cpp avec la commande ./sub qui va s’occuper d’écouter les messages reçu sur les topic configurés, ici le topic prédéfinis est « test » on peut voir qu’il arrive belle et bien à recevoir le message.