

Brevet de Technicien Supérieur

Systeme Numérique : Informatique et Réseau

Dossier Personnel du projet de casier

Charly BOURGOUIN

Sommaire

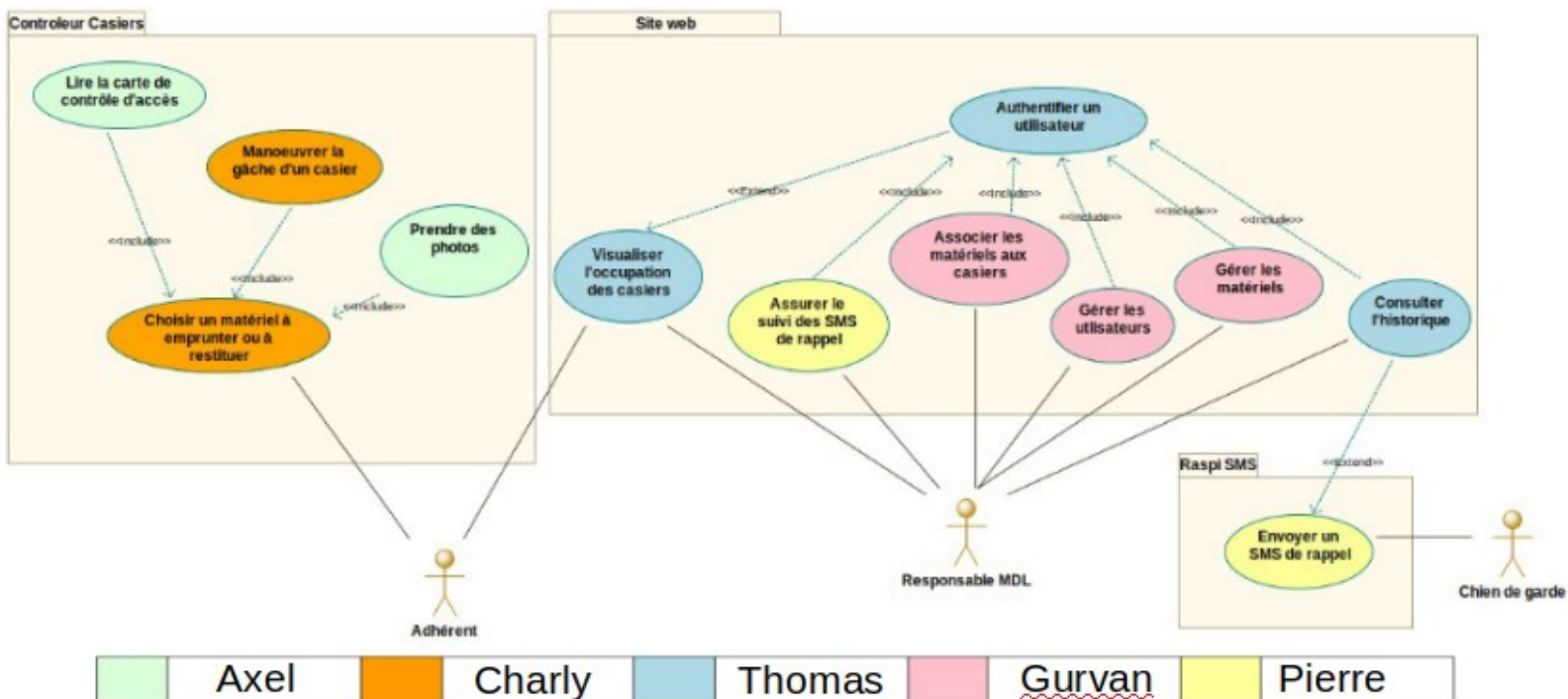
| | |
|--|-----------|
| 1 - Situation dans le projet..... | 3 |
| 1.1 – Synoptique de la réalisation..... | 3 |
| 1.2 – Prototype d’interface..... | 4 |
| 2 – cas d’utilisation « Choisir le matériel a emprunter ou a restituer »..... | 6 |
| 2.1 – Description de la tache..... | 6 |
| 2.2 – Conception détaillé..... | 6 |
| 3 - cas d’utilisation « Manœuvrer la gâche d’un casier »..... | 9 |
| 3.1 – Description de la tache..... | 9 |
| 3.2 – Conception détaillé..... | 9 |
| 4 – Tests Unitaires..... | 12 |
| 4.1 – Fiche de test unitaire..... | 12 |
| 4.2 – Résumé du test unitaire..... | 13 |
| 5 – Bilan de réalisation personnel..... | 13 |

1 - Situation dans le projet

1.1 – Synoptique de la réalisation

Pendant la phase de développement, j'ai du utilisé et m'intéresser à l'utilisation de

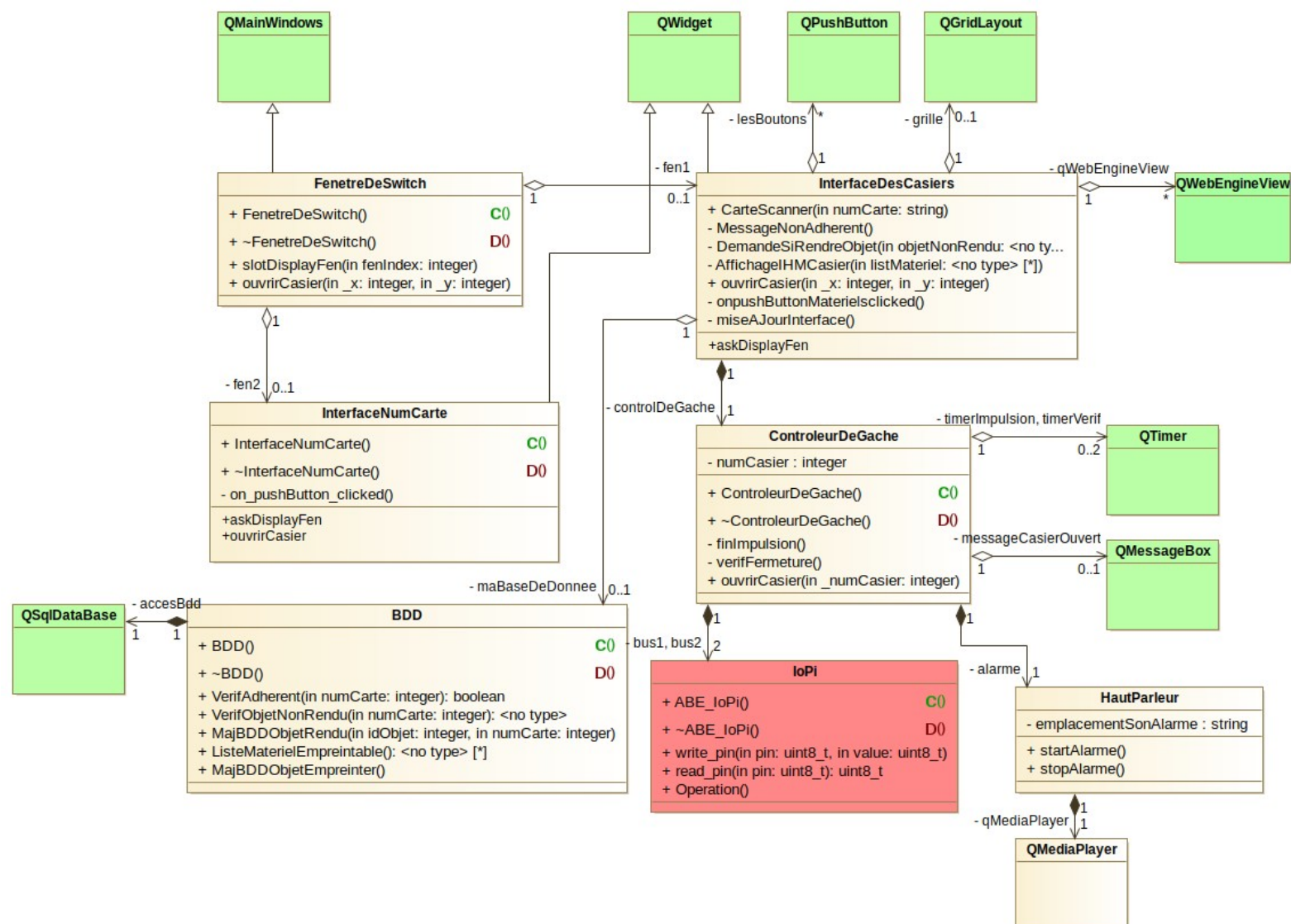
Diagramme de cas d'utilisation du projet :



Puisque, étant l'étudiant numéro 2, je me suis occupé de 2 cas d'utilisation que l'on peut distinguer avec la couleur Orange :

- **Choisir le matériel a emprunter ou a restituer** qui permet d'identifier l'utilisateur et d'automatiser au maximum l'utilisation du casier.
- **Manoeuvrer la gâche d'un casier** qui permet l'ouverture et la détection de fermeture d'un casier précis défini par l'objet sélectionner par l'utilisateur.

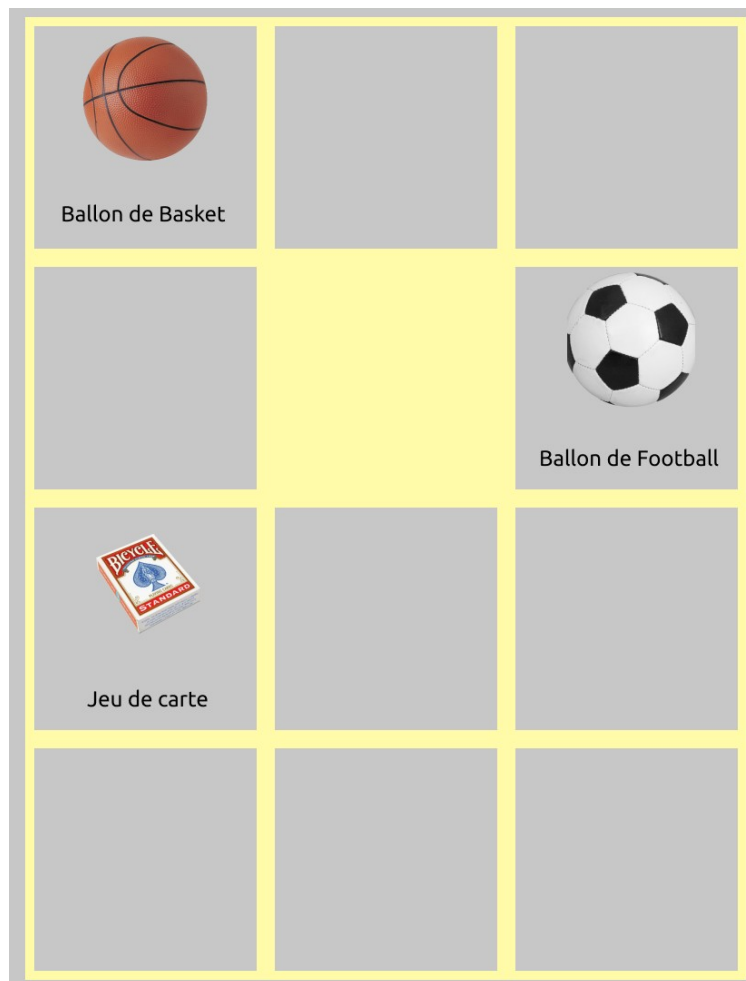
Voici le diagramme de classe qui m'a servi de base pour la conception des programmes de mes cas d'utilisation :



- **En vert** il s'agit des bibliothèques comprises dans QT.
- **En Rouge** il s'agit de la classe importée externe servant à faire fonctionner la carte d'extension

1.2 – Prototype d'interface

Après consultation du client, voici ce à quoi l'interface pourrait ressembler :



Ici il s'agit de l'interface de sélection des objets, l'interface avant d'être connecter sera un message demandant de bien vouloir passer sa carte, l'interface de sélection des objets sera afficher seulement si l'adhérent n'as pas déjà empreinté un objet, sinon il lui sera demandé si il veut rendre l'objet oui ou non.

L'interface est une représentation des casiers. Ils pourront être régler grâce a un fichier de configuration en « .ini ».

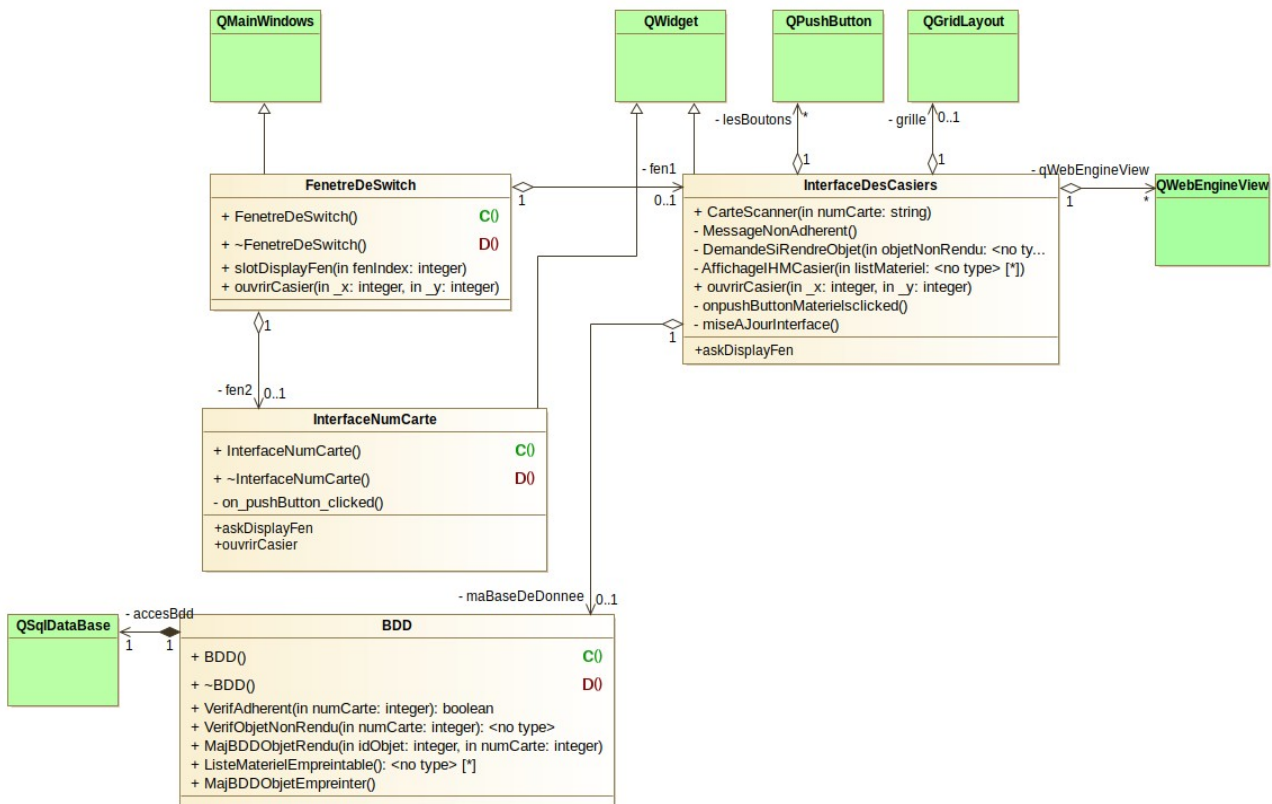
2 – cas d'utilisation « Choisir le matériel a emprunter ou a restituer »

2.1 – Description de la tâche

- Ce cas d'utilisation servira a l'affichage de l'interface et la sélection de l'objet.
- Une vérification de l'utilisateur sera faite a partir de son numéro de carte pour savoir si il est adhérent ou non et donc savoir si il est en droit d'utiliser le casier pour remprunter un objet.
- Par la suite il sera vérifier si l'adhérent a déjà emprunté un objet, si un objet est emprunté, alors il sera demandé a l'adhérent si il veut rendre l'objet. Dans le cas contraire, il sera redirigé vers l'interface d'empreint.
- L'interface d'empreint sera mis a jour a partir de la disponibilité des objet : il seront afficher sur l'interface si l'objet est présent dans le casier.

2.2 – Conception détaillé

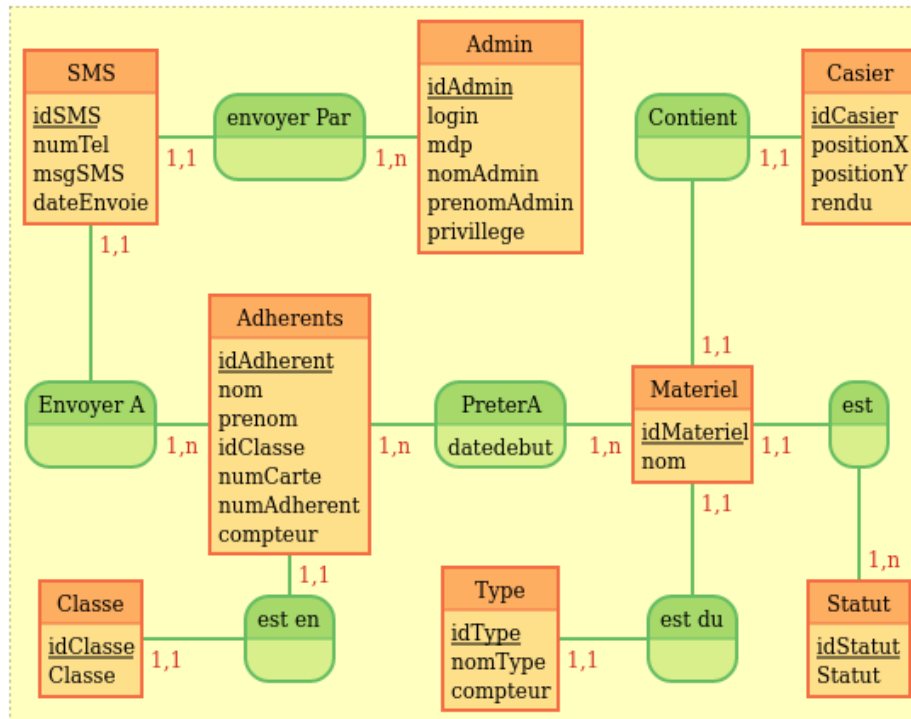
Voici les classes utilisé pour ce cas d'utilisation :



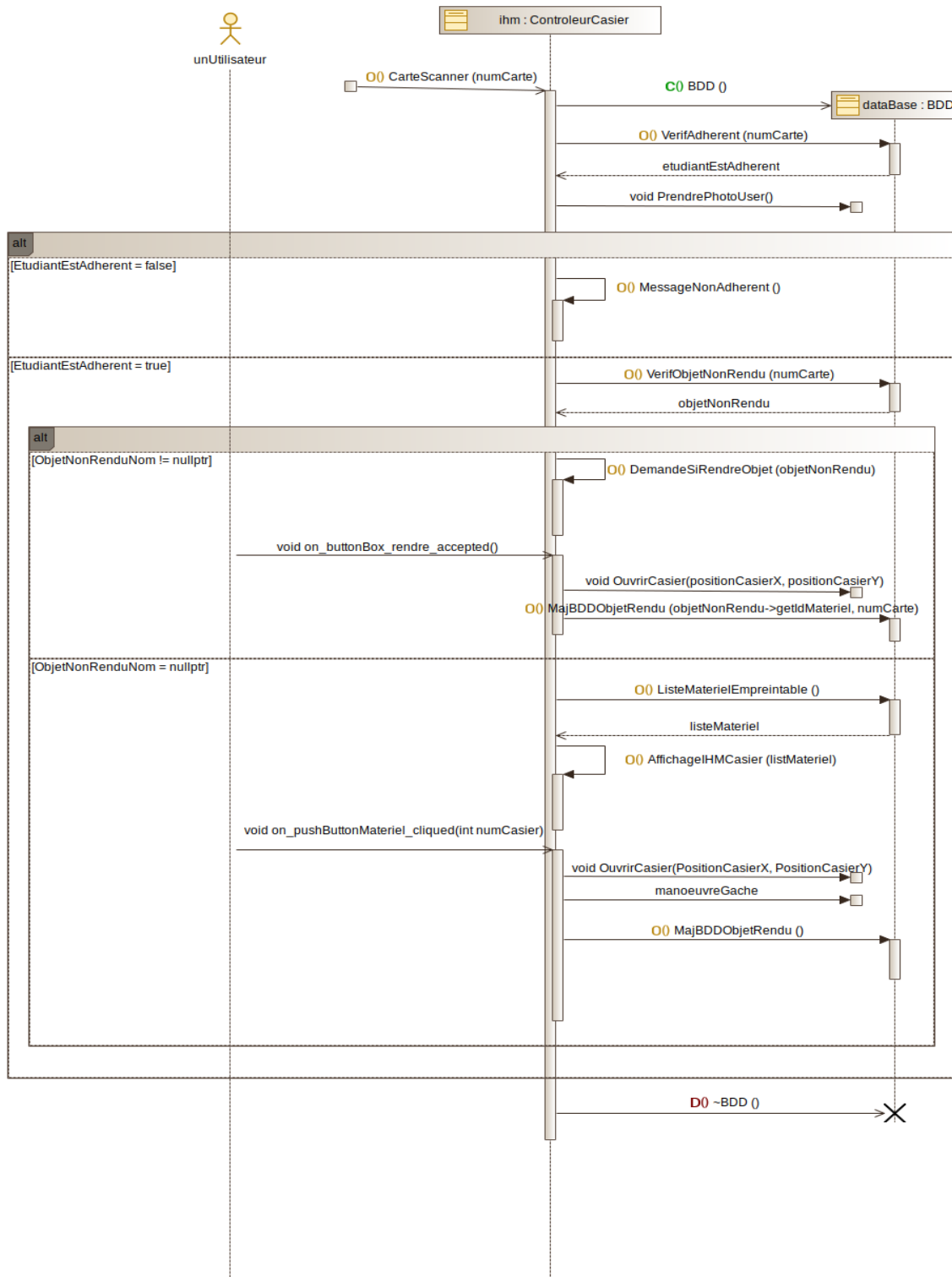
- **interfaceDeSwitch** : classes héritant de QMainWindow qui sera l'affichage principe, sert a pouvoir naviguer à travers les différentes interfaces.
- **interfaceNumCarte** : interface permettant d'attendre que l'utilisateur passe sa carte, elle sert aussi à identifier celui-ci et à savoir si il a déjà emprunter un objet.
- **interfaceDesCasiers** : interface affichant l'écran de sélection d'un objet.
- **BDD** : classe permettant de modifier et de lire la base de données.

Afin d'afficher l'interface, la Raspberry sera branché un un écran tactile. L'écran tactile 7" de Raspberry Pi donne aux utilisateurs la capacité de créer des projets intégrés tout-en-un, tels que les tablettes, les systèmes d'info-divertissement et les projets intégrés. L'écran 800 x 480 se connecte via une carte d'adaptation qui s'occupe de la conversion de la puissance et du signal. Seuls deux des connexions avec la Raspberry sont nécessaires ; l'alimentation du port GPIO et un câble ruban qui se connecte au port DSI présent sur tous les Raspberry Pi. Les pilotes d'écran tactile avec la prise en charge de 10 toucher simultanée et d'un clavier à l'écran sera intégré dans le dernier OS Raspbian pour une fonctionnalité sans clavier ni souris physique.

Voici la base de donnée avec ses différente table :



Voici le diagramme de séquence du cas d'utilisation :



Il s'agit du diagramme de séquence présenté lors de la conception préliminaire, depuis plusieurs changements on été fait mais le fonctionnement global reste le même

- L'IHM est séparé en trois classes d'instinct, "interfaceDeSwitch" qui permet de pouvoir passer d'un QWidget à l'autre, "interfaceNumCarte" qui est l'interface afficher avant le passage de la carte de l'utilisateur et "interfaceDesCasiers" qui est l'affiche de sélection de l'objet que l'on veut emprunter. "interfaceDeSwitch" permet de passer d'une interface à l'autre grâce à la classe "QStackWidget" et la méthode "". Donc pour afficher l'interface des casiers, un signal et envoyé afin de changer le QWidget afficher
- Quand l'adhérent a déjà emprunter un objet et qu'il souhaite le rendre, un signal est émis par "interfaceNumCarte" avec la position x et y du casier jusqu'à "interfaceDesCasiers" pour pouvoir récupérer le numéro de casier à partir de sa position

Pour cette partie du projet, un fichier de configuration en « .ini » a été utilisé :

[BDD]

identifiant=snir

ip=172.18.58.7

motDePasse=snir

nomBDD=projet_casier

[CASIER]

nbcasierx=3

nbcasiery=5

positionsystemex=1

positionsystemey=1

nbcasier=15

- La partie [BDD] est la partie ou sont enregistrer les données nécessaires à la connexion à la base de donnée :
 - **identifiant** : identifiant de l'utilisateur
 - **ip** : adresse ip du serveur contenant la base de donnée
 - **motDePasse** : mot de passe de l'utilisateur possédant l'identifiant donné
 - **nomBDD** : nom de la base de donnée ciblée
- La partie [CASSIER] quand a elle permet de paramétrer l'affichage des casier dans la classe "interfaceDesCasiers" :

- **nbcasierx** : contient le nombre de colonne de l'armoire.
- **nbcasiery** : contient le nombre de ligne de casiers de l'armoire.
- **positionsystemex** : désigne la colonne ou se trouve le casier servant a contenir le système. Il commence de 0 et va de gauche à droite.
- **positionsystemey** : désigne la ligne ou se trouve le casier servant a contenir le système. Il commence de 0 et va de haut en bas.
- **Nbcasier** : contient le nombre de casier

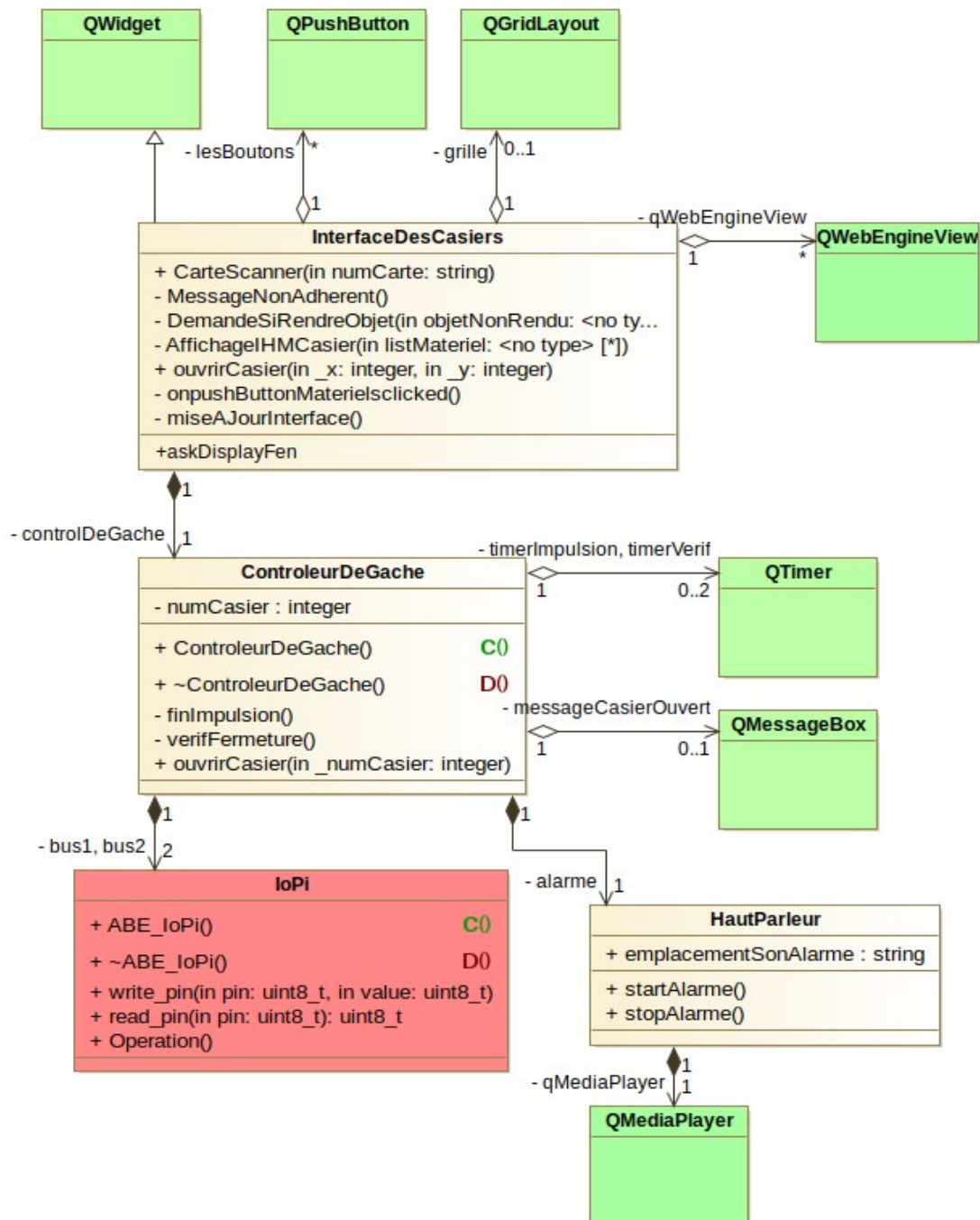
3 - cas d'utilisation « Manœuvrer la gâche d'un casier »

3.1 – Description de la tâche

- Ce cas d'utilisation servira à l'ouverture du casier sélectionner
- Le programme permettra de détecter la fermeture du casier
- De l'ouverture à la fermeture du casier, une alarme sera émise par le casier.

3.2 – Conception détaillé

Voici les classes utilisé pour ce cas d'utilisation :

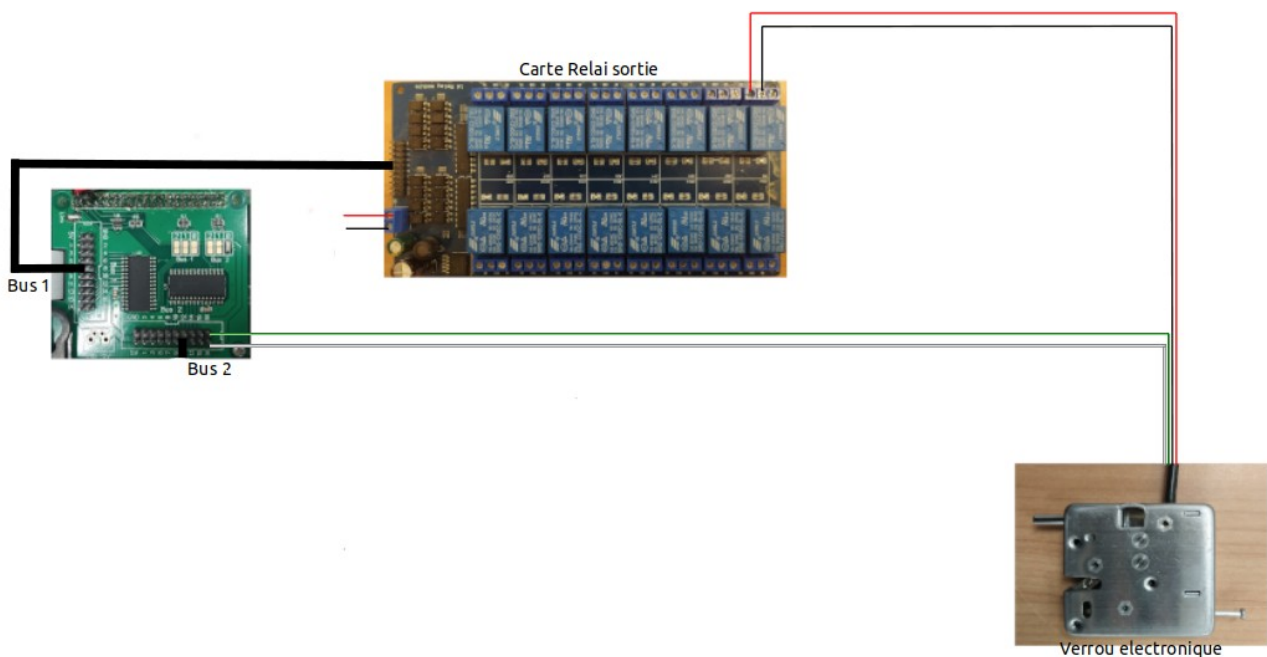


- **ControleurDeGache** : il s'agit de la classe permettant le lancement de la procédure d'ouverture
- **HautParleur** : classe permettant l'utilisation de l'alarme
- **IoPi** : est une classe qui permet d'écrire ou de lire grâce à la carte d'extension

Ce programme servira a la gestion de la partie matériel, le matériel est composé de :

- **1 Carte d'extension** : La carte d'extension est un module de la Raspberry pi qui permet d'augmenter le nombre des entrées/sorties numériques a 32.
- **1 Carte relai** : La carte relai est une carte permettant une alimentation dispenser par une alimentation extérieur pilotable a l'aide d'un contrôleur. Elle possède 16 entrée/sortie pilotable indépendamment des autres, elle peut fournir une tension jusqu'à 12 V
- **de 1 a 16 Gâches** : La gâche est un système de verrou électrique avec un système de détection d'ouverture. Elle est alimenté en 12 VDC et . Il marche par émission de courant (Verrouillé lorsque alimenté : normalement fermé) et possède une détection d'ouverture. La gâche possède aussi un « push » qui permet, quand le verrou est fermé, de donner une impulsion a la porte. Il possède de petite dimension (67 x 59 x 13) le rendant facile à disposer sans un cassier.

Ces différent composant son connecté entre eux de la manière suivante :

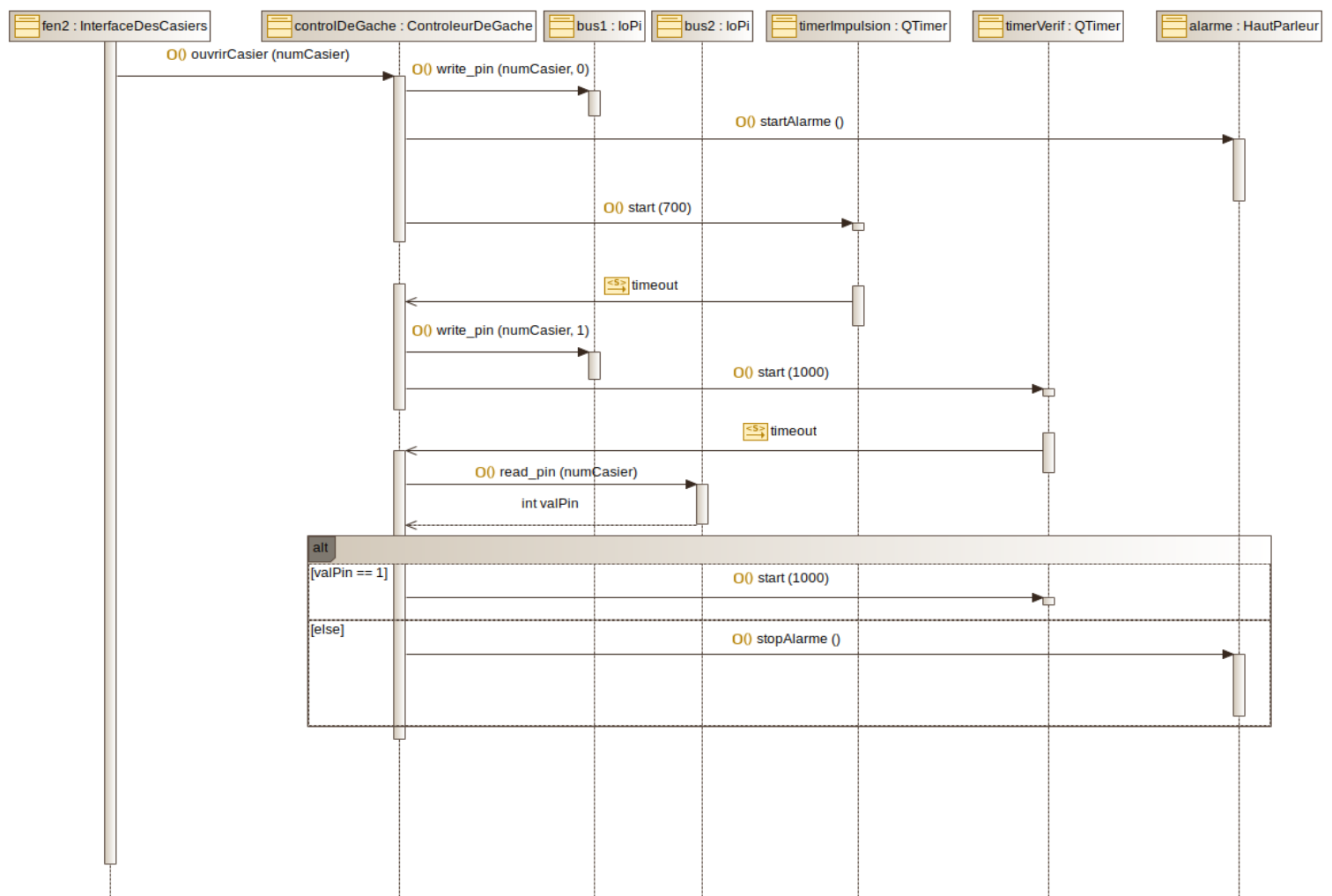


| Bus 1 | Carte Relai |
|-------|-------------|
| +5v | +5v |
| GND | GND |
| 1 | 1 |
| 2 | 2 |
| ... | |
| 16 | 16 |

| Carte Relai | Gache |
|-------------|-------|
| + | Rouge |
| - | Noir |

| Gâche | Bus2 |
|-------|------------|
| Vert | +5v |
| Blanc | Pin 1 à 16 |

Voici le diagramme de séquence de ce cas d'utilisation :



Pour l'utilisation de la carte d'extension, l'utilisation des bibliothèque du constructeur est indispensable,

Pour l'utilisation faite de la carte d'extension, nous avons eue besoin de 4 méthode de la classe IoPi :

`bus1(0x20)` Ici il s'agit du constructeur de la classe, pour utiliser les bus, il fallait préciser les adresses de ceux-ci, il s'agit ici des valeurs « 0x20 » et « 0x21 ».

```
bus1.set_bus_direction(0x0000);
bus2.set_bus_direction(0xFFFF);
```

la méthode « set_bus_direction » permet de d'indiquer si le bus sera une entrée ou une sortie

- **0x0000** désigne le bus en tant que **sortie**
- **0xFFFF** désigne le bus en tant qu'**entrée**

`bus1.write_bus(0xFFFF);` cette méthode permet d'écrire sur l'entièreté du bus. Ici on réinitialise sur le bus1 l'entièreté des pins

`bus1.write_pin(numCasier, 0);` méthode qui permet d'écrire sur un pin ciblé, ici on active le pin se trouvant à l'index « numCasier »

`bus2.read_pin(numCasier)` permet de lire un pin défini, ici sur le bus2, on lit le pin correspondant à numCasier

4 – Tests Unitaires

4.1 – Fiche de test unitaire

| Fiche de tests | | | |
|---|--|---|-------|
| Nature : | Fonctionnel | Référence : | F.2.1 |
| Module : | Cas d'utilisation : Choisir un matériel à emprunter ou à restituer | | |
| Objectif : | Vérifier le bon fonctionnement des gâches et de l'accès a la base de données | | |
| Condition du test | | | |
| État initial du module | | Environnement du test | |
| Programme | Projet_Casier | - Raspberry PI avec écran tactile - - | |
| Conditions initiales | | | |
| Présence de au moins deux adhérents et un objet dans la base de données et de l'objet dans le casier (Annexe BDD) | | | |
| Procédure de test | | | |

| Repère | Opérations | Résultats attendus |
|--------|--|---|
| 1 | Se connecter en tant que non adhérent | → Affichage de message de refus d'accès si l'utilisateur n'est pas adhérent |
| 2 | Se connecter en tant que adhérent qui n'a pas emprunter un objet | → l'interface d'emprunt doit s'afficher |
| 3 | Affichage de l'interface d'emprunt | → Si le casier ciblé contient un objet dans la base de données, activation du bouton et affichage de l'image de l'objet ainsi que sont nom. |
| 4 | | → Si le casier ciblé ne contient d'objet dans la base de donnée, désactivation du boutons |
| 5 | Sélection d'un objet sur l'interface d'emprunt | → Ouverture du casier sélectionner → début de l'émission de l'alarme |
| 6 | Fermeture du casier ouvert | → Arrêt de l'alarme → MaJ de la ligne « rendue » de la table « CASIER » → Ajout d'une ligne dans la table « PRETERA » <div> <p>Accès a la base de donnée :</p> <p>phpMyAdmin : http://172.18.58.7/phpmyadmin/</p> <p>identifiant : snir</p> <p>mot de passe : snir</p> </div> |
| 7 | Se connecter en tant que adhérent qui a emprunter un objet | → l'interface de restitution doit s'afficher |
| 8 | Valider le rendu | → Ouverture du casier sélectionner → début de l'émission de l'alarme |
| 9 | Fermeture du casier ouvert | → Arrêt de l'alarme → MaJ de la ligne « rendue » de la table CASIER <div> <p>Accès a la base de donnée :</p> <p>phpMyAdmin : http://172.18.58.7/phpmyadmin/</p> <p>identifiant : snir</p> <p>mot de passe : snir</p> </div> |

4.2 – Résumé du test unitaire

A cause de erreur du au « connect » qui ne voulais pas fonctionner ainsi que le manque de temp, je ne pus pas validé les tests 5, 6, 8 et 9 :

Pour remplacer la validation par carte, j'ai ajouté une zone de test pour taper a la main le numéro de carte, un fois le numéro de carte rentrer , il reconnu si le numéro appartenait a un adhérent ou non.

Par la suite l'interface s'affichait correctement en fonction des objet présent dans la base de données.

Lorsque un adhérent se connecte alors qu'ils a déjà remprunter un objet, le programme lui propose de le rendre.

5 – Bilan de réalisation personnel

Actuellement, beaucoup de chose on été modifié et le programme est extrêmement différent de ce qui est dit dans ce dossier personnel, mais manquant de temps et éprouvant certaine difficulté en modélisation, je les ai laissé comme telle . Le problème des « connect » n'a pas encore été résolue.

Après la résolution de ces problèmes, il ne restera plus que la gestion des alarmes ainsi que l'affichage des image en icône sur las boutons de l'interface.