



***Projet : Contrôleur  
de rondes***

***Dossier personnel  
Maëva Soulard  
«Le Rondier»***

Membres du groupe :  
Oujidane IMER  
Yoann LETESSIER  
Maëva SOULARD  
Damien CABARET

Professeur référent :  
Philippe CRUCHET  
Commanditaire du projet :  
Laurent DIGUER

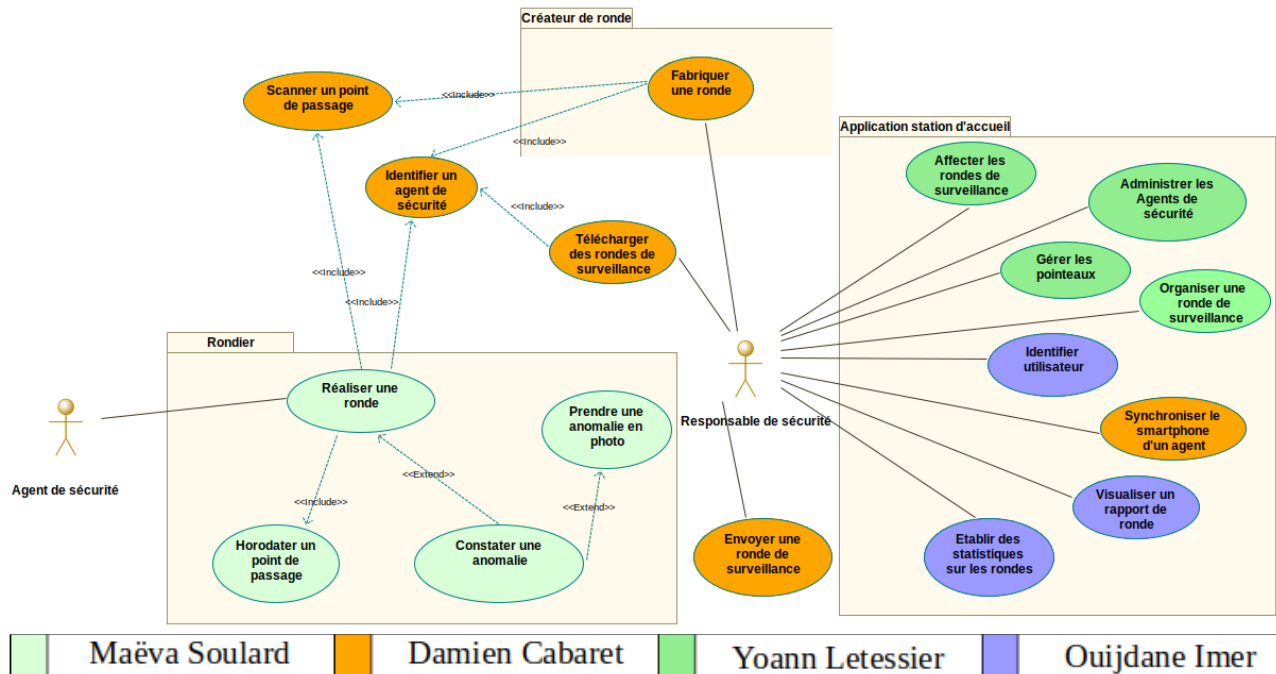
## **Sommaire**

I. Situation dans le projet.....	1
II. Cas d'utilisation «Réaliser une ronde».....	1
A) Description de la tâche.....	1
B) Conception détaillée.....	1
C) Test unitaire.....	3
D) Réalisation.....	5
1) Échange entre QML et C++.....	5
2) Fonctionnement d'une ListView en QML.....	6
E) Compte rendu du test unitaire.....	8
III. Bilan personnel.....	8
IV. Annexes.....	9
A) Annexe 1.....	9
B) Annexe 2.....	10
C) Annexe 3.....	12

## I. Situation dans le projet

L'entreprise BARBE-DEVAUX SAS a fait une demande auprès de notre section pour avoir des applications leur permettant de pouvoir réaliser leur ronde plus aisément et de pouvoir vérifier plus facilement le bon travail des agents de sécurité. De ce fait, le diagramme de cas d'utilisation ci-dessous a été effectué pour répondre au mieux à leur besoin.

Si nous prenons Damien CABARET, Yoann LETESSIER et Ouijdane IMER, ils doivent réaliser



une application pour un ordinateur sous Windows qui se nomme «Station d'accueil». Cette application doit servir à gérer les rondes de surveillance à effectuer ou déjà effectuées. Cependant, Damien doit aussi faire une application, le «Créateur de ronde», pour smartphone sous Android avec un lecteur NFC qui permettra au responsable des agents de sécurité de créer les rondes de surveillance tout en les faisant à pied. En outre, moi-même (Maëva Soulard), je m'occupe d'une application, «Rondier» pour un smartphone sous Android avec également un lecteur NFC. Cette application servira aux agents de sécurité, elle les aidera à effectuer une ronde de surveillance plus aisément.

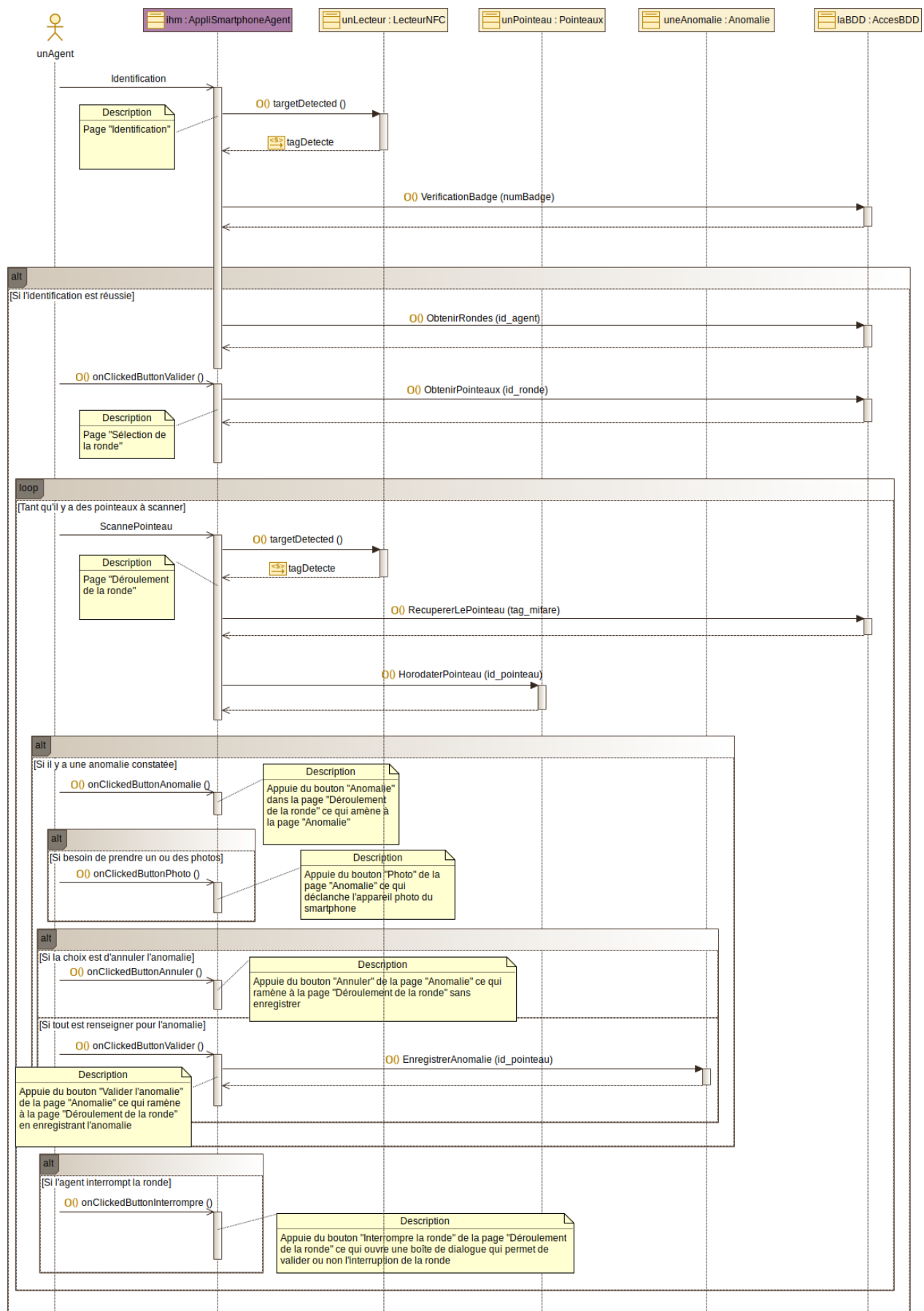
## II. Cas d'utilisation «Réaliser une ronde»

### A) Description de la tâche

Comme dit précédemment, cette partie consiste à créer l'application «Rondier». En effet, cette application sert à réaliser une ronde de surveillance. Pour cela, l'application permet d'horodater un point de passage ou un pointeau c'est-à-dire lorsque l'agent de sécurité valide un pointeau cela met à jour dans la base de données l'heure à laquelle le pointeau a été validé. L'application permet aussi de déclarer une anomalie et de la prendre en photo.

### B) Conception détaillée

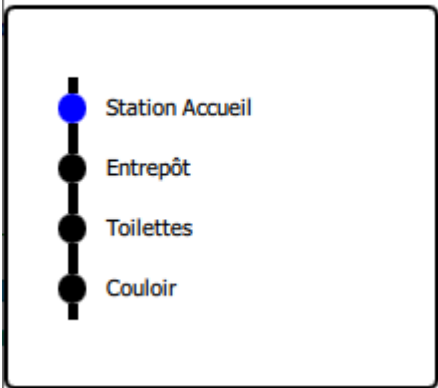
Pour réaliser ce cas d'utilisation, j'ai effectué un diagramme de séquence ci-dessous :



Tout d'abord, l'agent de sécurité commence par s'identifier à l'aide de son badge et de son smartphone avec le lecteur NFC et l'application sur la page «Identification». Après son identification, une requête SQL est effectuée pour récupérer les rondes qui sont affectées à l'agent. L'agent de sécurité arrive sur la page «Sélection de la ronde». Il doit choisir la ronde qu'il va effectuer à l'aide du menu déroulant. Pour valider son choix, l'agent de sécurité doit appuyer sur le bouton «Valider» qui va lui afficher la page «Déroulement de la ronde» et effectuer une requête SQL qui permet de récupérer tous les pointeaux qui constituent la ronde choisie. L'agent de sécurité va pouvoir effectuer sa ronde de surveillance en scannant tous les pointeaux indiqués. S'il découvre une anomalie au niveau d'un pointeau, il peut la déclarer à l'aide du bouton «Anomalie» qui l'envoie sur la page «Anomalie». Sur cette page, il peut décrire l'anomalie, prendre une photo. S'il choisit de prendre une photo, cela l'envoie sur le page qui ouvre l'appareil photo. De plus, sur la page «Anomalie», l'agent de sécurité peut choisir de valider son anomalie ce qui l'enregistre dans la base de données avec ou sans photo et qui le renvoie sur la page «Déroulement de la ronde» ou il peut aussi l'annuler ce qui le renvoie sur la page «Déroulement de la ronde». Sur la page «Déroulement de la ronde», l'agent de sécurité peut interrompre la ronde ce qui ferme l'application.

### C) Test unitaire

Fiche de test unitaire			
Nature :	Fonctionnel	Référence :	F.1.1
Module :	Réaliser une ronde (cas d'utilisation)		
Objectif :	Vérification de la mise à jour de la BDD lors du déroulement de la ronde		
Condition du test			
État initial du module		Environnement du test	
Être sur l'IHM «Sélection de la ronde»		Smartphone sous Android avec lecteur NFC, application «Rondier», tag, ordinateur avec DB Browser	
Procédure de test			
Pour tous pointeaux scannés ou toute anomalie enregistrée, il faut noter l'heure, le pointeau, l'ordre de passage, la description de l'anomalie.			
Conditions initiales		Résultats attendus	
Vérifier l'état initial de la BDD avec DB Browser et le fichier «Rondier_BDDRemplie.db».		Les tables «aEteScannePar», «aEteEffectueePar» et «photos» doivent être vides tandis que dans la table «anomalies», il doit y avoir une seule ligne avec comme «id» 1 et comme description «Défaut d'ordre». L'Annexe 2 montre le contenu de départ de chaque table explicitement. L'Annexe 1 montre la structure de la BDD.	
Faire la procédure ADB pour transférer la BDD sur smartphone (Annexe 3).		La BDD est accessible depuis l'application sur smartphone.	
Repère	Opérations		Résultats attendus
1	Choisir la ronde se nommant «Lycée Touchard-Washington» à		Passage à la page du déroulement de la ronde. Affichage du nom de la ronde «Lycée Touchard-

	l'aide du menu déroulant puis la valider à l'aide du bouton «Valider».	Washington» et de la liste des pointeaux : <u>Lycée Touchard-Washington</u> 
2	Scanner premier pointeau «Station Accueil».	Pointeau «Station Accueil» devient vert, pointeau suivant «Entrepôt» devient bleu. Horodatage du pointeau dans la BDD.
3	Scanner le pointeau «Toilettes».	Enregistrement de l'anomalie : défaut d'ordre sur le pointeau «Toilettes» avec horodatage dans la BDD. Pointeau «Toilettes» devient orange et le pointeau «Entrepôt» reste bleu.
4	Scanner le pointeau «Entrepôt».	Pointeau «Entrepôt» devient vert, pointeau suivant «Toilettes» devient bleu. Horodatage du pointeau dans la BDD.
5	Scanner le pointeau «Toilettes».	Pointeau «Toilettes» devient vert, pointeau suivant «Couloir» devient bleu. Horodatage du pointeau dans la BDD.
6	Scanner le pointeau «Couloir», mais il est défectueux donc appuyer sur «Ignorer le pointeau».	Affichage de la page «Anomalie» avec description préremplie «Pointeau ignoré». Horodatage du pointeau dans la BDD.
7	Appuyer sur le bouton «Valider» puis confirmer.	Enregistrement d'une anomalie : pointeau ignoré, car tag défectueux. Renvoi à la page «Déroulement de la ronde». Pointeau «Couloir» devient rouge et le pointeau suivant «Bureau» devient bleu.
8	Scanner le pointeau «Bureau»	Pointeau «Bureau» devient vert, pointeau suivant «Cuisine» devient bleu. Horodatage du pointeau dans la BDD.
9	Remarquer une anomalie : une fenêtre ouverte.	Affichage de la page «Anomalie». Remplir la description : fenêtre ouverte. Prendre une photo de la fenêtre ouverte.
10	Appuyer sur le bouton «Valider» puis confirmer.	Enregistrement de l'anomalie avec la photo. Horodatage de l'anomalie du pointeau dans la BDD. Renvoi à la page «Déroulement de la ronde».

		Pointeau «Bureau» devient rouge et le pointeau suivant «Cuisine» devient bleu.
11	Scanner le pointeau «Cuisine».	Pointeau «Cuisine» devient vert, pointeau suivant «Station Accueil» devient bleu. Horodatage du pointeau dans la BDD.
12	Scanner le pointeau «Station Accueil» ⇒ fin de la ronde	Pointeau «Station Accueil» devient vert. Horodatage du pointeau dans la BDD. Affichage d'un message de fin de ronde.
Condition finale		Résultats attendus
Analyser les données de la BDD après la réalisation de la ronde.		Les tables «aEteScannePar», «aEteEffectueePar», «photos» et «anomalies» doivent être remplies avec les informations notées pour chaque pointeau et anomalie. L'Annexe 1 montre la structure de la BDD.

## D) Réalisation

Pour réaliser l'application, nous avons fait le choix d'utiliser le langage QML et du C++ pour coder celle-ci. Nous avons également utilisé comme environnement de développement Qt Creator.

### 1) Échange entre QML et C++

Pour pouvoir récupérer des informations de la base de données, il faut utiliser des requêtes SQL dans les méthodes en C++. Cependant, il y avait besoin en QML de pouvoir récupérer ces informations. Il faut donc faire le lien entre le C++ et QML. Voici le code dans le main.cpp qui permet de faire cela :

```
QqmlApplicationEngine engine;
//permet de récupérer les valeurs de la liste C++ pour les utiliser en QML
engine.rootContext()->setContextProperty("pointeauxModel", QVariant::fromValue(listeDesignationPointeaux));
//permet de faire le lien entre QML et classe AccesBdd
engine.rootContext()->setContextProperty("bdd", new AccesBdd());
//permet de faire le lien entre QML et classe Pointeau
engine.rootContext()->setContextProperty("pointeau", new Pointeau());
//permet de faire le lien entre QML et la classe Agent
engine.rootContext()->setContextProperty("agent", new Agent());
//permet de faire le lien entre QML et la classe Ronde
engine.rootContext()->setContextProperty("ronde", new Ronde());
//emplacement du fichier QML correspondant
engine.load(QUrl(QStringLiteral("qrc:/main.qml")));
```

Tout d'abord, ligne 1, il faut commencer par déclarer qu'il y a une application en QML avec la variable *engine*. Celle-ci va permettre de faire le lien entre les classes en C++ et l'application en QML.

Ensuite, nous pouvons voir sur les lignes 5, 7, 9 et 11 qu'on déclare un nouveau pointeur sur la classe qui va permettre de récupérer les informations invocables de la classe avec le nom entre guillemets. Cependant, la ligne 3 permet de récupérer en QML la liste des pointeaux déjà remplie auparavant.

La ligne 13 permet d'indiquer où se situe le main.qml dans le projet.

```
Q_INVOKABLE QList<QString> obtenirListeNomsRondes(QString _numBadge);
```

Pour qu'une méthode de la classe en C++ puisse être invocable en QML ou en JavaScript, il suffit de mettre devant «Q\_INVOKABLE».

Par exemple, pour récupérer la liste des noms des rondes, il suffit en JavaScript de faire appel à la

```
function obtenirNomRondes(){  
    var nomRonde = bdd.obtenirListeNomsRondes("1B4DBE53");  
    console.log(Array.from(nomRonde));  
    return Array.from(nomRonde);  
}
```

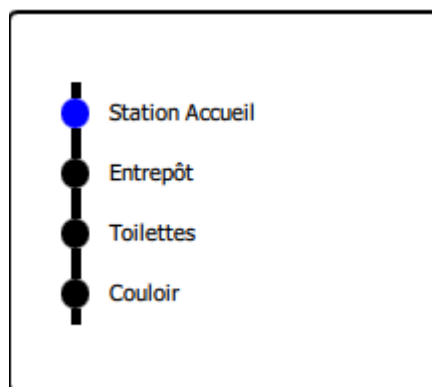
méthode en C++ qui se situe dans la classe «AccesBdd» à l'aide de la variable déclarer dans le main.cpp qui est *bdd*. Cette méthode JavaScript permet de retourner un tableau des noms des rondes pour ensuite pouvoir remplir de comboBox du choix de la ronde.

## ***SELECTION DE LA RONDE***

**Bonjour , veuillez choisir votre ronde**

### **2) Fonctionnement d'une ListView en QML**

Pour pouvoir afficher la liste des points à scanner, il faut utiliser un ListView en QML. Voici ce que cela donne :



Voici le code qui permet de réaliser cet affichage :



```

80 Rectangle {
81     width:217; height: 192
82     border.width: 2
83     radius: 5
84     x:29
85     y:110
86     Layout.rowSpan: 5
87
88 Component {
89     id: representation // Manière dont s'affiche les données
90     Row {
91         height: 30
92         spacing: 10
93         //Déclaration de variables QML qui font le lien avec celles en C++
94         required property string m_couleur
95         required property string m_designation
96         Text { text: " " } // pour ne pas coller la représentation graphique
97         Column {
98             Rectangle // premier rectangle
99             {
100                 x:5
101                 width: 5
102                 height: 8
103                 color: "black"
104             }
105             Rectangle // rond
106             {
107                 id: etatPointeau
108                 height: 15;
109                 width: 14;
110                 radius: 20
111                 color: m_couleur
112             }
113             Rectangle // deuxième rectangle
114             {
115                 x:5
116                 width: 5
117                 height: 8
118                 color: "black"
119             }
120         }
121         Text // texte correspondant
122         {
123             id: designationPointeau
124             y:8
125             text: m_designation
126             MouseArea{
127                 anchors.fill: parent
128                 onClicked: listePointeaux.currentIndex = index
129             }
130         }
131     }
132 }
133 ListView {
134     id: listePointeaux
135     anchors.rightMargin: 13
136     anchors.leftMargin: 14
137     anchors.topMargin: 36
138     anchors.bottomMargin: 40
139     anchors.fill: parent
140     model: pointeauxModel
141     delegate: representation
142 }
143 }

```

Tout d'abord, le rectangle permet de contenir toute la ListView. Le Component permet d'avoir l'affichage des points et de la barre de progression avec les couleurs correspondantes et les désignations des pointeaux correspondants. Ensuite dans la ListView, le model est la liste des pointeaux récupérée en C++ et le delegate est l'affichage de la liste qui reprend le Component.

### **E) Compte rendu du test unitaire**

Lors du test unitaire, nous n'avons pas pu aller plus que l'étape 1, car je n'arrivais pas à afficher la liste des pointeaux. De ce fait, toutes les actions à faire lors de la ronde et la mise à jour de la base de données ne pouvait pas être faite. Par conséquent, nous avons trouvé une solution qui est d'avoir une deuxième application avec laquelle la liste des pointeaux est définie dès le début.

### **III. Bilan personnel**

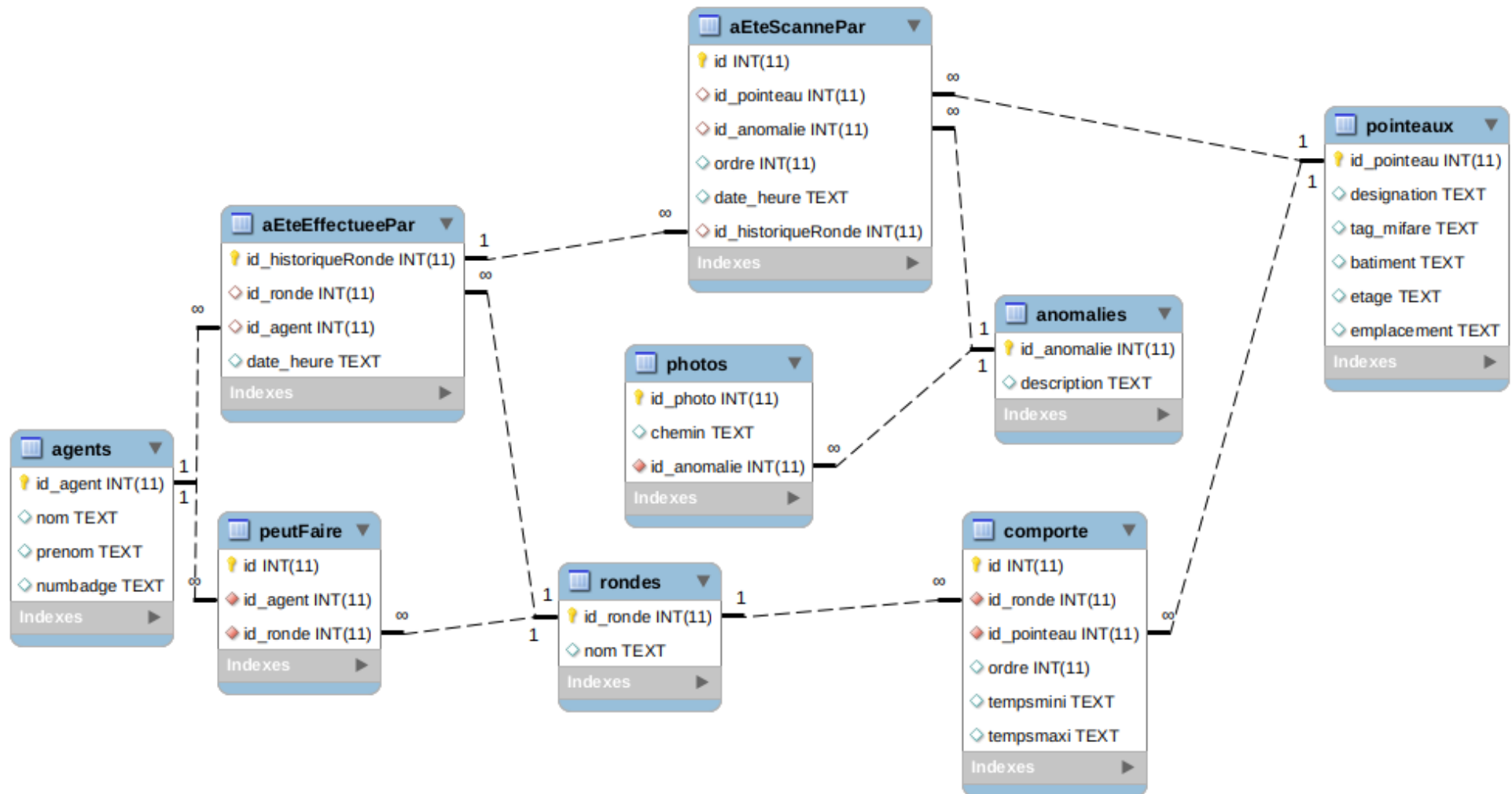
Au moment où je rédige ce dossier personnel, l'application «Rondier» ne fonctionne pas encore comme elle le devrait. C'est-à-dire que je rencontre des problèmes avec l'affichage dynamique de la liste des pointeaux en QML. Je suis obligée de la déclarer dès le début de l'application. Cependant, j'ai pu avancer sur le reste de l'affichage en faisant une deuxième application qui ne permet pas de choisir la ronde à effectuer, mais qui affiche dès le début la liste des pointeaux à effectuer. De plus, je n'ai pas encore pu faire la mise à jour de la base de données.

Dans les prochains jours, j'espère pouvoir finir l'affichage des toutes les données sur l'application et pouvoir faire la mise à jour de la base de données.

Le projet m'a permis d'apprendre un peu plus à travailler en groupe, à se répartir des tâches... De plus, il m'a également permis de découvrir le QML, de pouvoir créer une application pour smartphone, mais aussi de savoir gérer ses tâches et son temps avec les imprévus qu'il peut y avoir.

## IV. Annexes

### A) Annexe 1



## B) Annexe 2

Structure de la Base de Données			
Parcourir les données			
Éditer les Pr			
Table : aEteEffectueePar			
id_historiqueRonde	id_ronde	id_agent	date_heure
Filtre	Filtre	Filtre	Filtre

Structure de la Base de Données					
Parcourir les données					
Éditer les Pragmas					
Exécuter le SQL					
Table : aEteScannePar					
id_pointeau	id_anomalie	ordre	date_heure	id_historiqueRonde	id
Filtre	Filtre	Filtre	Filtre	Filtre	Filtre

Structure de la Base de Données				
Parcourir les données				
Table : agents				
id_agent	nom	prenom	numbadge	
Filtre	Filtre	Filtre	Filtre	
1	5	Lambert	Martin	1B4DBE53

Structure de la Base de Données	
Parcourir les données	
Table : anomalies	
id_anomalie	description
Filtre	Filtre
1	1 Defaut d'ordre

Structure de la Base de Données			
Parcourir les données			
Table : peutFaire			
id_agent	id_ronde	id	
Filtre	Filtre	Filtre	
1	5	4	22
2	5	1	23

Structure de la Base de Données		
Parcourir les données		
Table : photos		
id_photo	chemin	id_anomalie
Filtre	Filtre	Filtre

Structure de la Base de Données	
Parcourir les données	
Table : rondes	
id_ronde	nom
Filtre	Filtre
1	1 Lycée Touchard-Washington
2	2 Carrefour
3	3 Auchan
4	4 Jacobins

Table : comporte

	id_ronde	id_pointeau	ordre	tempsmini	tempsmaxi	id
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	1	1	1	00:00:00.0000	00:00:00.0000	1
2	1	5	2	00:05:00.0000	00:10:00.0000	2
3	1	8	3	00:07:00.0000	00:14:00.0000	3
4	1	7	4	00:06:00.0000	00:12:00.0000	4
5	1	9	5	00:08:00.0000	00:16:00.0000	5
6	1	11	6	00:03:00.0000	00:09:00.0000	6
7	1	1	7	00:08:00.0000	00:16:00.0000	7
8	2	2	8	00:10:00.0000	00:15:00.0000	8
9	2	3	9	00:16:00.0000	00:18:00.0000	9
10	2	4	10	00:05:00.0000	00:10:00.0000	10
11	3	1	11	00:08:00.0000	00:13:00.0000	11
12	3	2	13	00:17:00.0000	00:22:00.0000	12
13	3	3	14	00:08:00.0000	00:15:00.0000	13
14	3	5	15	00:14:14.0000	00:15:00.0000	14
15	3	4	16	00:06:00.0000	00:12:00.0000	15
16	4	6	17	00:14:00.0000	00:15:00.0000	16
17	4	7	18	00:08:00.0000	00:09:00.0000	17
18	4	9	19	00:08:00.0000	00:09:00.0000	18
19	2	1	7	00:08:00.0000	00:16:00.0000	19

Table : pointeaux

	id_pointeau	designation	tag_mifare	batiment	etage	emplacement
	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
1	1	Station Accueil	FD24DE4C	B	Rez-de-chaussé	B004
2	5	Entrepôt	838B87B1	A	1	a104
3	7	Couloir	SFDGH5451D	C	Sous-sol	Porte A
4	8	Toilettes	DBAF2854	D	2	D205
5	9	Bureau	AB3C2554	Eglise	Clochet	Cloche
6	11	Cuisine	BBC92954	Dortoir	2	Chambre 209

## C) Annexe 3

- Procédure ADB sous Linux

- De l'ordinateur vers le smartphone

1. Installer l'application sur le téléphone.
2. Ouvrir un terminal sur l'ordinateur.
3. Faire la commande «adb push 'chemin de la BDD sur l'ordinateur' 'chemin provisoire sur le smartphone'» ce qui correspond à la commande «**adb push /home/USERS/ELEVES/SNIR2019/msoulard/ContrôleurDeRondes2021/ContrôleurDeRondes/BDDs/Rondier/Rondier\_BDDRemplie.db /sdcard/Download/**» en utilisation ma session et mon smartphone.
4. Faire la commande «**adb shell**» qui permet d'aller sur le smartphone.
5. Faire la commande «**run-as org.qtproject.testRondier**» qui permet d'aller dans les dossiers de l'application.
6. Si c'est la première qu'on exécute ces commandes, il faut ensuite faire la commande «**mkdir db**» qui permet de créer le dossier où l'on va copier la BDD.
7. Faire la commande «**cp /sdcard/Download/Rondier\_BDDRemplie.db /data/data/org.qtproject.testRondier/db/**» qui permet de copier la BDD du dossier provisoire vers le dossier de l'application.

- Du smartphone vers l'ordinateur

1. Ouvrir un terminal sur l'ordinateur.
2. Faire la commande «**adb shell**» qui permet d'aller sur le smartphone.
3. Faire la commande «**run-as org.qtproject.testRondier**» qui permet d'aller dans les dossiers de l'application.
4. Faire la commande «**cp /data/data/org.qtproject.testRondier/db/Rondier\_BDDRemplie.db /sdcard/Download/**» qui permet de copier la BDD du dossier de l'application vers le dossier provisoire.
5. Faire la commande «**exit**» pour sortir du dossier de l'application.
6. Faire la commande «**exit**» pour sortir du téléphone.
7. Faire la commande «adb pull 'chemin provisoire de la BDD' 'chemin de la BDD sur l'ordinateur'» ce qui correspond «**adb pull /BDDs/Rondier/Rondier\_BDDRemplie.db /sdcard/Download/Rondier\_BDDRemplie.db /home/USERS/ELEVES/SNIR2019/msoulard/ContrôleurDeRondes2021/ContrôleurDeRondes/BDDs/**» en utilisation ma session et mon smartphone.

- Procédure ADB sous Windows

- De l'ordinateur vers le smartphone

1. Installer l'application sur le téléphone.
2. Ouvrir un terminal sur l'ordinateur.
3. Faire la commande «**cd 'chemin vers le dossier d'installation d'ADB'**» pour que l'ordinateur connaisse l'emplacement des fichiers ADB.
4. Faire la commande «adb push 'chemin de la BDD sur l'ordinateur' 'chemin provisoire sur le smartphone'» ce qui correspond à la commande «**adb push /home/USERS/ELEVES/SNIR2019/msoulard/ContrôleurDeRondes2021/ContrôleurDeRondes/BDDs/Rondier/Rondier\_BDDRemplie.db /sdcard/Download/**» en utilisation ma session et mon smartphone.
5. Faire la commande «**adb shell**» qui permet d'aller sur le smartphone.

6. Faire la commande «**run-as org.qtproject.testRondier**» qui permet d'aller dans les dossiers de l'application.
7. Si c'est la première qu'on exécute ces commandes, il faut ensuite faire la commande «**mkdir db**» qui permet de créer le dossier où l'on va copier la BDD.
8. Faire la commande «**cp /sdcard/Download/Rondier\_BDDRemplie.db /data/data/org.qtproject.testRondier/db/**» qui permet de copier la BDD du dossier provisoire vers le dossier de l'application.
  - **Du smartphone vers l'ordinateur**
1. Ouvrir un terminal sur l'ordinateur.
2. Faire la commande «**cd 'chemin vers le dossier d'installation d'ADB'**» pour que l'ordinateur connaisse l'emplacement des fichiers ADB.
3. Faire la commande «**adb shell**» qui permet d'aller sur le smartphone.
4. Faire la commande «**run-as org.qtproject.testRondier**» qui permet d'aller dans les dossiers de l'application.
5. Faire la commande «**cp /data/data/org.qtproject.testRondier/db/Rondier\_BDDRemplie.db /sdcard/Download/**» qui permet de copier la BDD du dossier de l'application vers le dossier provisoire.
6. Faire la commande «**exit**» pour sortir du dossier de l'application.
7. Faire la commande «**exit**» pour sortir du téléphone.
8. Faire la commande «**adb pull 'chemin provisoire de la BDD' 'chemin de la BDD sur l'ordinateur'**» ce qui correspond «**adb pull /BDDs/Rondier/Rondier\_BDDRemplie.db /sdcard/Download/Rondier\_BDDRemplie.db /home/USERS/ELEVES/SNIR2019/msoulard/ControleurDeRondes2021/ControleurDeRondes/BDDs/**» en utilisation ma session et mon smartphone.