



Rapport de Mi-Projet Smart Picking

Sommaire :

Sommaire

Introduction

1) Contexte du projet

2) Cahier des charges

3) Travail réalisé

a. Système de détection

b. Programme Arduino

c. Application Android

d. Application Web et Base de Données

4) Planning pour la suite

a. Liaison série

b. Localisation par balise BLE

c. Chemin optimal

Conclusion

Introduction :

Dans le cadre de notre projet de fin d'étude, voici un premier rapport présentant le contexte de projet ce projet, réalisé pour la Pharmacie du CHRU de Lille, le cahier des charges, le travail réalisé durant ces premiers mois ainsi que le planning pour la suite du projet.

1) Contexte du projet

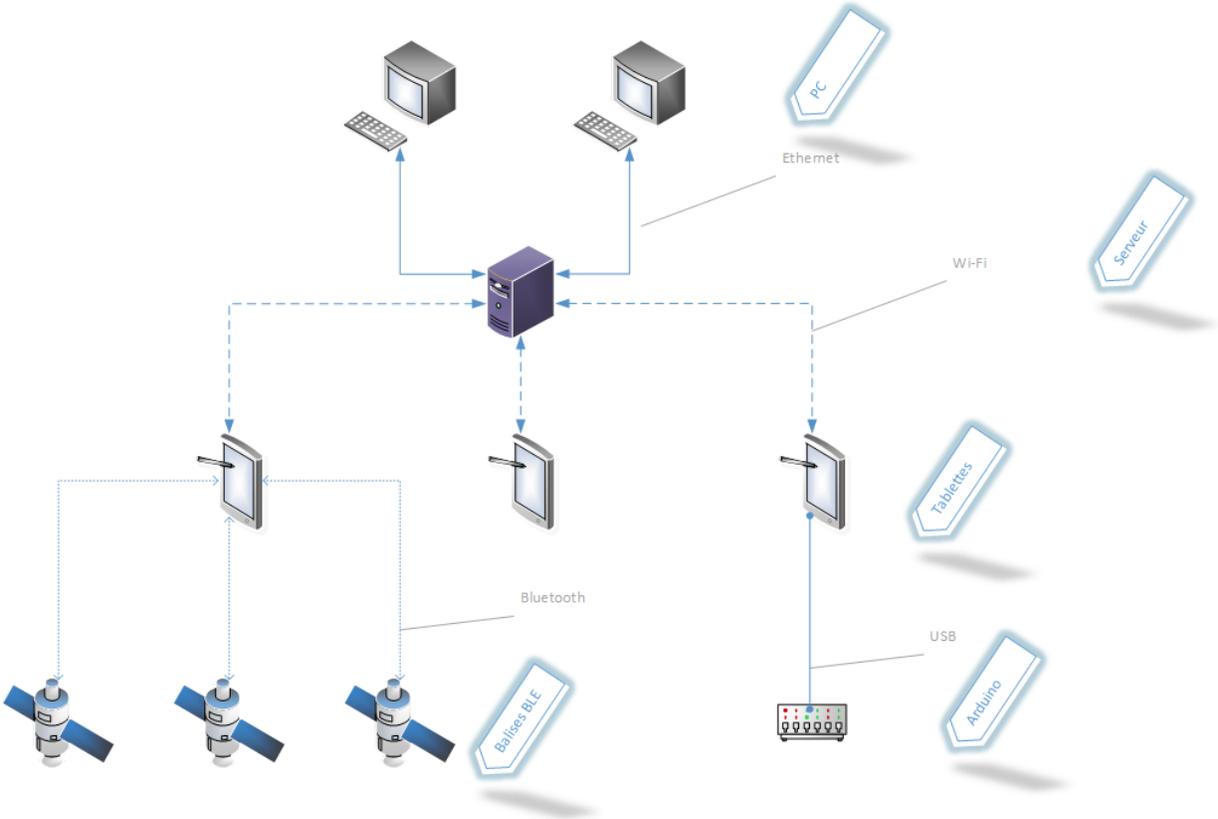
Le "picking" ou ramassage des médicaments dans une pharmacie de grande taille, ici celle du CHRU de Lille qui compte plus de 3000 références de médicaments, stockés sur 3 étages, est une tâche répétitive et non optimisée. Des erreurs peuvent se produire lors de la préparation des commandes et un temps important est perdu pour l'orientation des préparateurs de commande dans les différents rayons de l'entrepôt, notamment en formation, et même pour un préparateur régulier il est impossible de retenir l'emplacement de tous les médicaments. Nous proposons de mettre en place une méthode "Smart picking" permettant l'agrégation de 6 commandes différentes ainsi que la localisation et l'assistance au déplacement des différents préparateurs en leur proposant un chemin optimal permettant de récupérer les médicaments commandés dans un temps inférieur à celui actuellement tout en évitant de croiser un autre chariots dans des rayons parfois un peu étroits. La solution passe par la mise au point d'un chariot intelligent à 6 bacs (un par commande) permettant : la localisation dans l'entrepôt par l'utilisation de balises BLE (Bluetooth Low Energy), l'affichage d'un parcours optimisé pour le ramassage des médicaments et la vérification de l'intégrité de la commande, c'est à dire vérifier que les médicaments ont été placés dans le bon bac. Le chariot embarquera une tablette sous Android.

2) Cahier des charges

Pour mener à bien notre projet, nous devons mettre en relation plusieurs entités différentes :

- Une tablette sous Android faisant office d'interface de contrôle et d'orientation pour le préparateur, grâce à une application simple d'utilisation, mais complète. Cette application donnera la liste des médicaments à récupérer pour chaque bac, les contenus des bacs à tout moment, le plan de l'entrepôt avec l'itinéraire à parcourir, la localisation de la tablette donc du chariot et du préparateur.
- Un système pour détecter le nombre de médicaments déposés dans tel ou tel bac. Celui-ci utilisera des émetteurs et capteurs infrarouges ainsi qu'un Arduino qui devra envoyer les informations à la tablette.
- Des balises BLE permettant la localisation de la tablette dans l'entrepôt.
- Une base de données MySQL contenant la liste des médicaments avec les différentes informations associées comme l'emplacement et le conditionnement, les commandes, les plans et les objets associés (balises, "murs", emplacements...) et également l'historique d'utilisation des différents dispositifs.
- Une interface Web permettant de créer les plans, de consulter l'historique, d'ajouter les médicaments avec leur localisation sur les plans, de voir la localisation de tous les chariots en activité, et de rentrer les commandes. Elle servira aussi d'intermédiaire entre la tablette et la base de données.
- Une LED à côté de chaque bac qui indiquera, en s'allumant, pour chaque médicament, s'il faut (encore) en mettre dans ce bac.

Schéma général



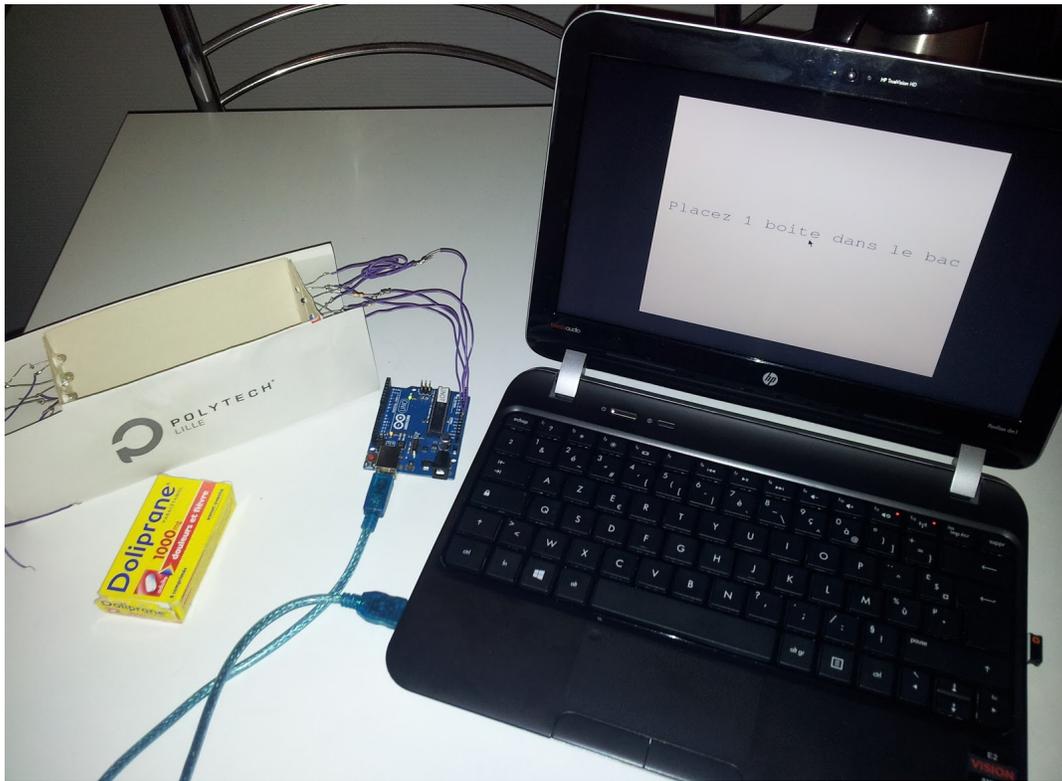
3) Travail réalisé

a. Système de détection

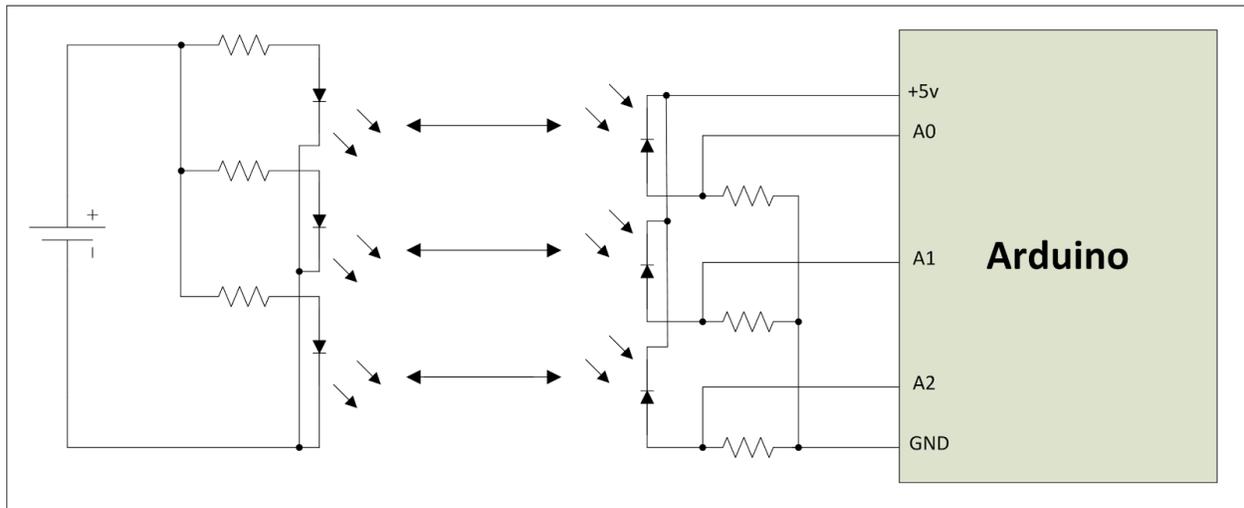
La première étape de notre projet aura été de construire un prototype du système permettant de détecter la pose des différents médicaments dans les bacs. Celui-ci devra se trouver sur chacun des 6 bacs, pour donner des informations à la tablette sur le nombre de médicaments récupéré et le(s) bac(s) dans le(s)quel(s) ils ont été mis, cela dans le but d'éviter les erreurs et simplifier la procédure. Nous avons donc réalisé un prototype, pour le moment sur un seul bac, qui détecte lorsqu'une boîte est placé dans le bac.

Ce prototype utilise un système de détection infrarouge, avec d'un côté une rangée de LEDs infrarouges et de l'autre une rangée de phototransistor qui renvoie une certaine tension en fonction de la quantité de rayons infrarouges reçus. Cette tension est mesurée par un Arduino, ainsi lorsque une boîte passe entre les émetteurs et les récepteurs la tension chute, l'Arduino transmet alors l'information à l'ordinateur qu'une boîte a été mise dans le bac.

Ci-dessous une photographie du dispositif :



Et ci-dessous le schéma de la partie Arduino : Lorsqu'un phototransistor est éclairé (par une diode infrarouge), il est passant, et la tension aux bornes de la résistance en série avec celui-ci est élevée ($\sim 5V$). Lorsqu'un objet passe entre la diode et le phototransistor, ce dernier se bloque et fait chuter l'intensité dans la résistance, et donc la tension à ses bornes ($\sim 0V$). C'est donc la tension aux bornes de ces résistances que l'on mesure pour détecter les objets (ci-dessous les tension entre A1, A2, A3 et la masse).



Une vidéo de démonstration est disponible sur la page wiki du projet.

b. Programme Arduino

```
#include <avr/io.h>           // for the input/output register
#include <util/delay.h>

// For the serial port

#define CPU_FREQ      16000000L // Assume a CPU frequency of 16Mhz

void init_serial(int speed)
{
  /* Set baud rate */
  UBR0 = CPU_FREQ/(((unsigned long int)speed)<<4)-1;

  /* Enable transmitter & receiver */
  UCSRB = (1<<TXEN0 | 1<<RXEN0);

  /* Set 8 bits character and 1 stop bit */
  UCSRC = (1<<UCSZ01 | 1<<UCSZ00);
}

void send_serial(unsigned char c)
{
  //loop_until_bit_is_set(UCSR0A, UDRE0);
  while(!(UCSR0A & (1 << UDRE0)))
  {}
  UDR0 = c;
}

unsigned char get_serial(void) {
  loop_until_bit_is_set(UCSR0A, RXC0);
  return UDR0;
}

// For the AD converter
void ad_init(unsigned char channel)
{
  ADCSRA|=(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);
  ADMUX|=(1<<REFS0)|(1<<ADLAR);
  ADMUX=(ADMUX&0xf0)|channel;
  ADCSRA|=(1<<ADEN);
}
```

```

unsigned int ad_sample(void){
    ADCSRA|=(1<<ADSC);
    while(bit_is_set(ADCSRA, ADSC));
    return ADCH;
}

void input_init(void){
    DDRB = 0xFF;
    PORTB = 0xFF;
}

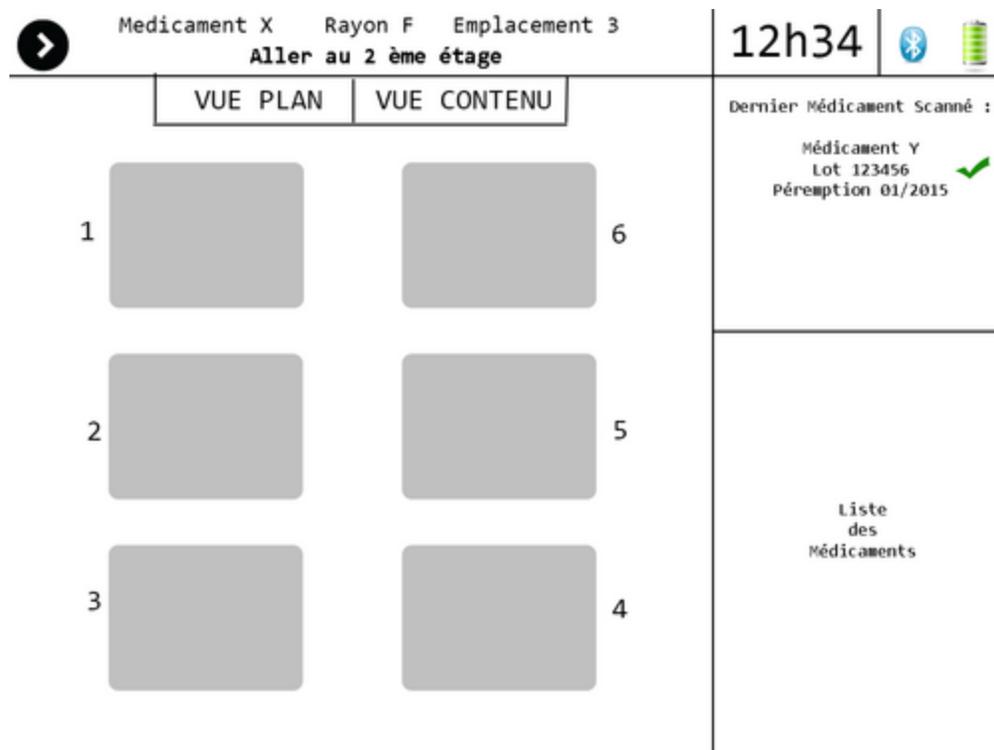
//Return 1 if something is detected, or return 0 if there is nothing
int detection(void){
    int t=0;
    ad_init(0x00);
    if(ad_sample()<50)
        t=1;
    ad_init(0x01);
    if(ad_sample()<50)
        t=1;
    ad_init(0x02);
    if(ad_sample()<50)
        t=1;
    return t;
}

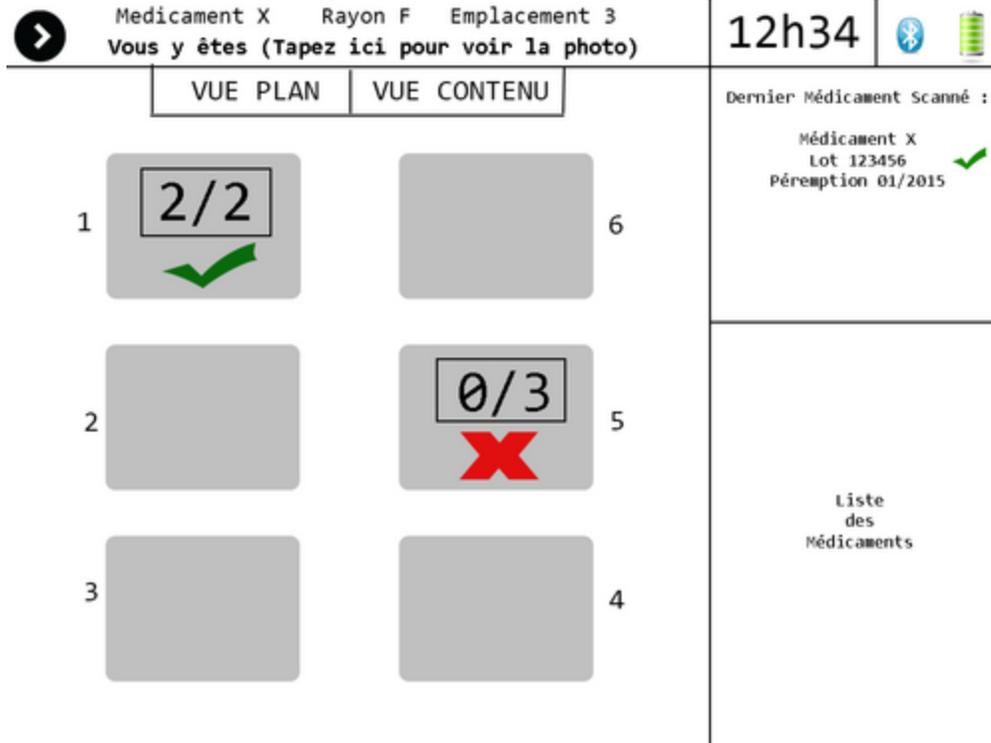
int main(void){
    init_serial(9600);
    int detected = 0;
    input_init();
    while(1){
        detected=detection();
        _delay_ms(200);
        if(detected==1){
            send_serial(0x31);//Medicine detected
            detected=0;
        }
        else
            send_serial(0x30);//Nothing detected
    }
    return 0;
}

```

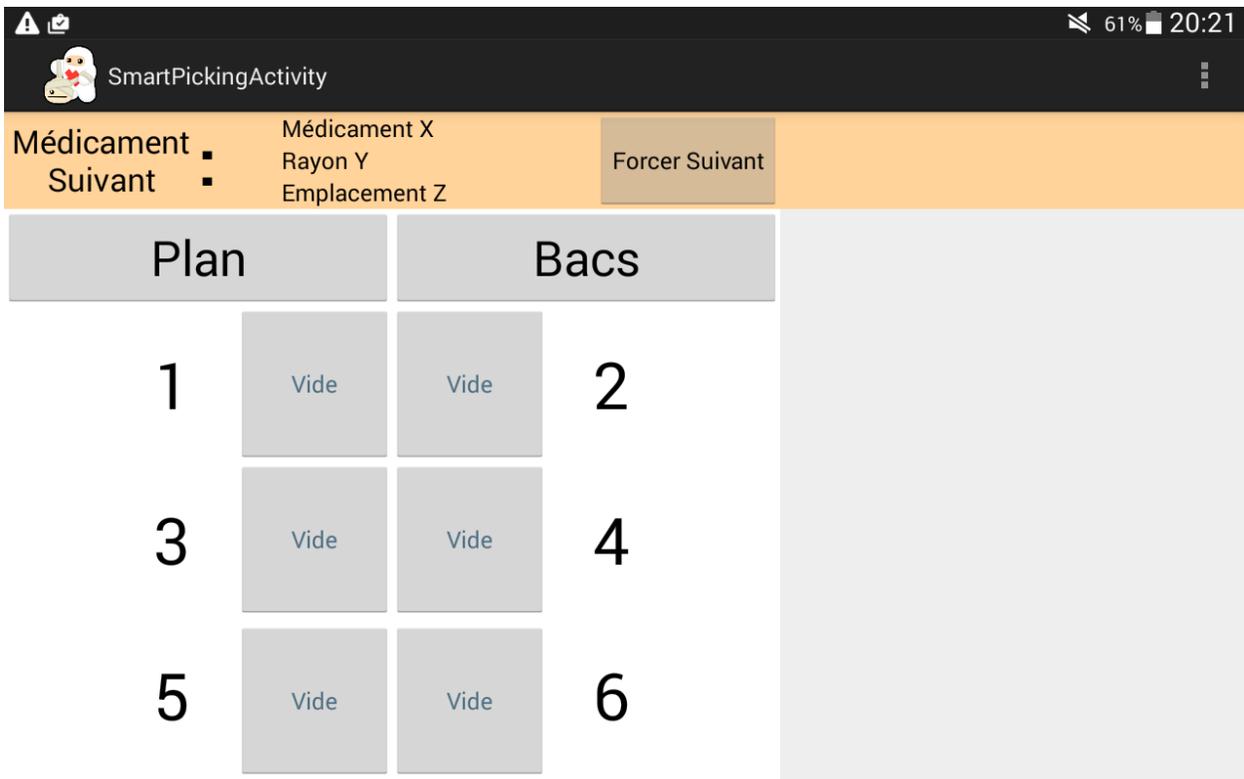
c. Application Android

Nous avons également commencé la réalisation de l'application Android, qui équipera les tablettes, et qui permettra au préparateur de se déplacer entre les différents médicaments, de savoir dans quel bac les mettre et en quelle quantité, mais aussi de vérifier qu'il n'a pas fait d'erreur. Nous utilisons pour cela le logiciel Android Studio, qui a été spécialement conçu pour créer des applications Android, ce qui permet de faciliter quelque peu le développement. Pour le moment, nous avons surtout travaillé sur l'interface graphique de l'application. Celle-ci tend à ressembler aux schémas ci-dessous, proposés lors de notre visite de la pharmacie, mais nous avons finalement dû faire quelques ajustements.



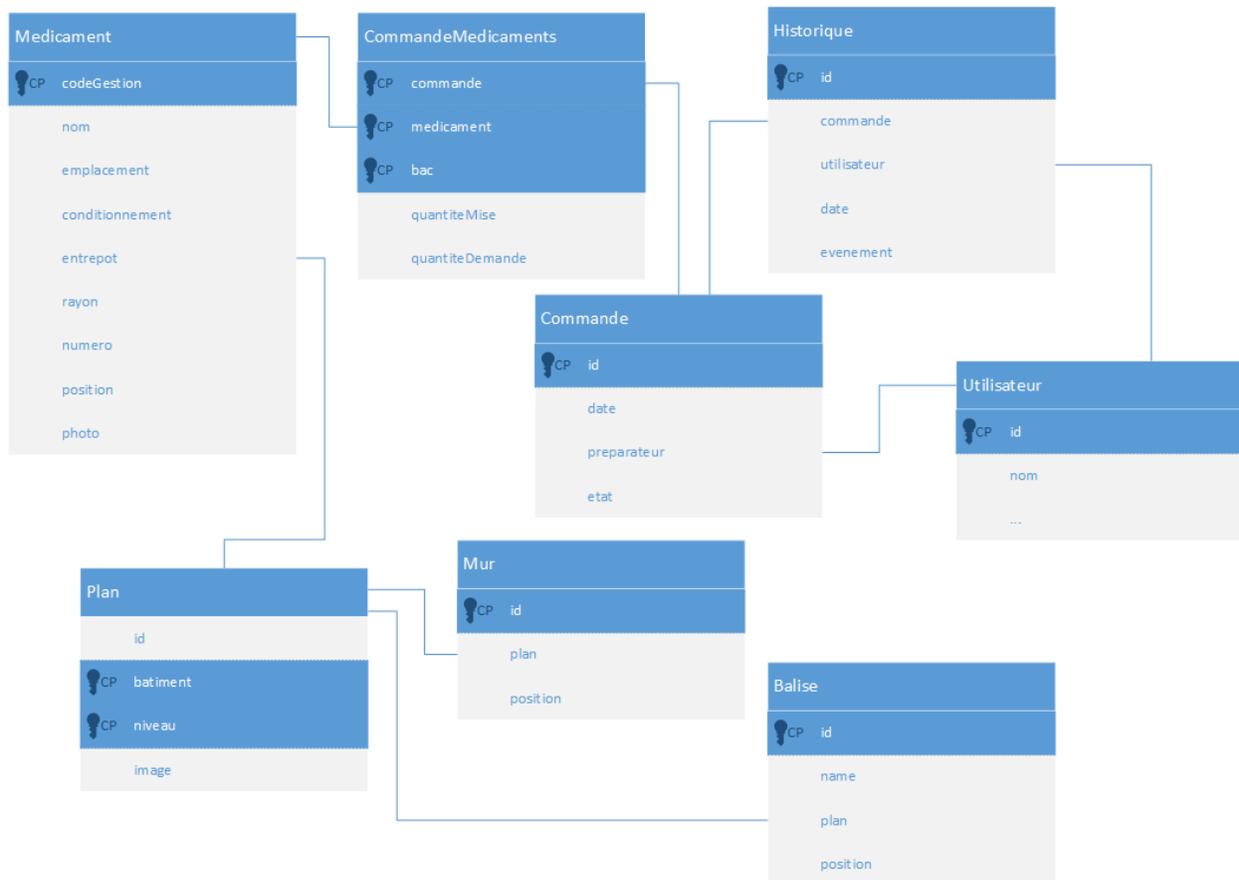


Ci-dessous le résultat temporaire de l'interface Android



d. Application Web et Base de Données

Nous avons choisi de réaliser l'application web et la base de donnée à l'aide du framework PHP Symfony2 qui est l'un des plus utilisés actuellement. Nous avons commencé par réfléchir aux différentes entités qui seront présentes sur l'application et nous avons ainsi obtenu le diagramme UML suivant :



Pour le moment les fonctionnalités réalisées sont l'enregistrement et la connexion d'utilisateurs, l'affichage de la liste des commandes et des médicaments, la création, la modification et l'affichage d'une commande, ainsi que l'ajout des médicaments. Vous pouvez voir ci-dessous la page listant toutes les commandes à

partir de laquelle les commandes seront assignées à un préparateur, et il est également possible de voir le contenu de la commande.

The screenshot displays the Smartpicking web application interface. At the top, there is a dark navigation bar with the logo 'Smartpicking', a search input field labeled 'Rechercher', and links for 'Connecté en tant que admin', 'Déconnexion', and 'Aide'. On the left side, a vertical sidebar contains a menu with the following items: 'Vue d'ensemble', 'Mes Commandes', 'Mon Compte', 'Mes Paramètres', 'Utilisateurs', 'Médicaments', 'Commandes', 'Plans', 'Tablettes', and 'Historique'. The main content area is titled 'Liste des Commandes' and features a table with the following data:

Etat	Id	Date	Préparateur	Voir	Supprimer
En Attente	1	07/12/2014 00h00	Non Assigné		
Vide	2	14/12/2014 17h15	Non Assigné		
Vide	3	15/12/2014 12h50	Non Assigné		
Vide	4	16/12/2014 16h29	Non Assigné		

Below the table, there is a blue button labeled 'Nouvelle Commande'. In the bottom right corner of the page, there is a small circular logo with the letters 'sy'.

4) Planning pour la suite

a. Liaison série

Le prototype proposait une liaison série entre l'Arduino et un ordinateur, mais ce qui nous intéresse dans le projet est la liaison entre la tablette et l'Arduino. Android ne proposant pas cette fonctionnalité de base, nous allons donc avoir recours à une librairie.

b. Localisation par balises BLE

Le but du projet étant de guider le préparateur de la commande à travers la pharmacie, il est important pouvoir localiser le chariot. La solution retenue est l'utilisation de balises BLE, facilement implémentables dans une application Android, et nécessitant peu d'énergie, ce qui facilitera l'installation car les balises n'ont pas besoin d'une source d'énergie autre qu'une cellule photovoltaïque récupérant l'énergie de l'éclairage de la pharmacie, ou sinon une simple pile qui pourrait durer des mois. Ensuite l'idée retenue pour la localisation est de mesurer la puissance du signal des balises les plus proches et de déterminer la position par trilatération, la puissance du signal donnant en théorie une idée de la distance par rapport à la balise, mais des tests seront à effectuer pour déterminer la viabilité de cette solution, notamment à cause des interférences, le signal n'arrivant pas forcément en ligne droite selon la configuration des lieux.

c. Chemin optimal

L'avantage du guidage est bien entendu de ne pas toujours suivre le même circuit, donc de déterminer le chemin optimal en tenant compte de la commande, ainsi que des obstacles éventuels tels que les autres chariots, ce qui nécessitera une communication entre chariots, ainsi que l'utilisation d'un algorithme de recherche de chemin. Une phase de tests sera aussi nécessaire ici afin de déterminer des temps moyens pour aller d'un point A à un point B, ce pour prédire à quel moment un chariot s'engagera dans tel rayon pour pouvoir l'éviter à tel moment du circuit, et déterminer le chemin optimal.

Conclusion

Pour conclure, même s'il est loin d'être terminé, nous pouvons voir que ce projet permet de mettre en oeuvre différentes technologies, différents langages de programmation, au sein d'un même système. De plus, l'objectif a une réelle importance et le résultat de ce projet pourra à terme éventuellement être utilisé quotidiennement dans la pharmacie du CHRU s'il est fonctionnel, et peut-être d'autres par la suite si le système est efficace. Il reste encore des défis technologiques à surmonter, mais nous espérons avoir la possibilité de mener ce projet à bien jusqu'à son terme.