Samg-mouit BTS SN-IR

Vaianu (IR 2)

Projet : CHRONO_NG

Avec Wolff Paul (EC) et Samg-mouit Vaiani (IR 1)

Lycée Couffignal 2018 -2019

SOMMAIRE

INTRODUCTION

Je m'appelle Samg-Mouit Vaianu, je suis originaire de la Polynésie Française et je suis arrivé en France métropolitaine en décembre 2017 pour faire un BTS SN-IR au lycée Louis Couffignal de Strasbourg.
J'aime beaucoup le sport, et ce projet m'a permis de pouvoir faire de la programmation informatique pour le sport. Et donc dans ce rapport je vous présenterais notre projet et mon implication.

PRESENTATION DU PROJET

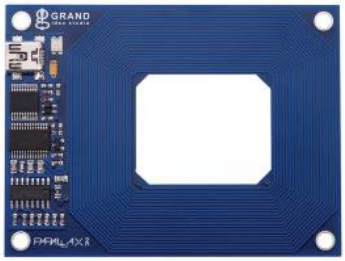
Le nom de notre projet est chrono_ng et il a pour but de pouvoir utiliser la technologie en faveur des organisateurs de courses à pied, que ce soit de la création d'une course à travers une interface homme-machine (application Windows) à l'inscription des participants soit par une interface homme-machine, soit par un site web, puis le jour de la manifestation au chronométrage automatique et à la publication des résultats sur un site web.
Nous sommes trois personnes à faire ce projet, deux informaticiens et un électronicien.
Et donc chacun à son rôle.
Un informaticien s'occupera de la partie création des courses, inscription des participants et publication des résultats.
Tandis que le deuxième informaticien s'occupera de la partie chronométrage automatique avec l'électronicien qui lui mettra en place des stations de chronométrage afin de détecter l'heure de passage de chaque coureur à différents endroits sur le parcours.

MON IMPLICATION DANS CE PROJET

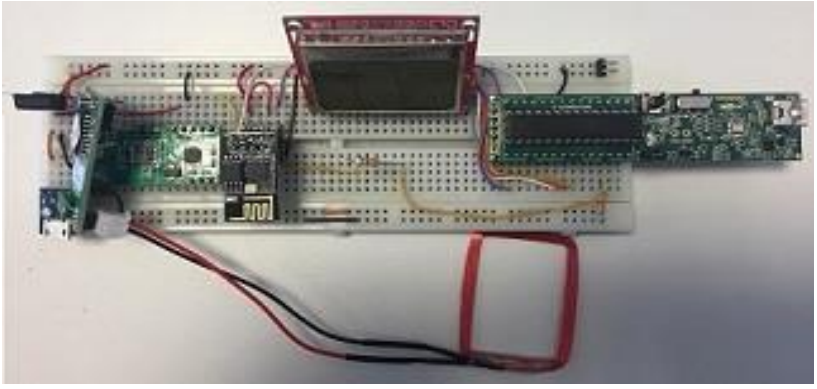
Mon rôle à moi dans ce projet est de m'occuper de la partie informatique du chronométrage automatique. Mais avant de vous montrer la partie technique, je vais vous présenter les matériels nécessaires pour faire du chronométrage automatique.

- Le premier c'est le badge RFID (radio-identification). Chaque coureur le jour de la course recevra un badge RFID unique et donc un numéro unique qui sera associé grâce à une application à son nom, à un dossard et qui va permettre de l'identifier pendant la course au passage devant les stations de chronométrage (temps départ, temps intermédiaire, temps arrivée) mis en place par mon collègue électronicien.

- Le deuxième c'est le lecteur de carte RFID usb paralax, il sera relié le jour de la manifestation au PC par USB. Il va permettre de récupérer et d'afficher les numéros des badges RFID sur l'application pour les associations RFID.

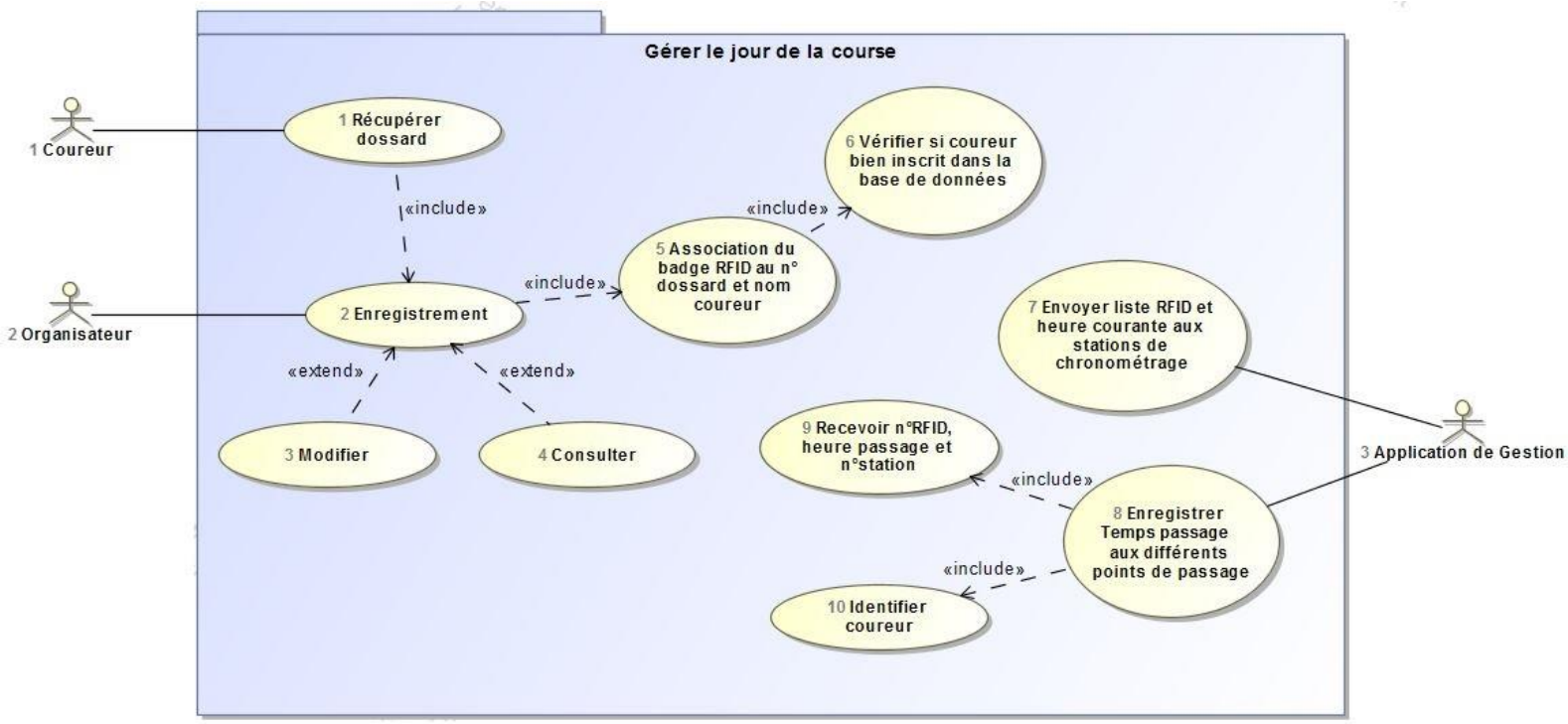


- Le troisième c'est les stations de chronométrage. Ils seront mis à des endroits stratégiques sur le parcours et seront connectées à un point d'accès Wifi émis par un mobile. Leurs fonctions seront de détecter tous les badges RFID (coureur) qui passeront devant eux et de transmettre le numéro RFID, le numéro station et l'heure de passage du coureur à une application de gestion.



Maintenant je vais vous présenter la partie technique. Et pour cela nous allons voir plusieurs diagrammes afin de comprendre le déroulement du chronométrage automatique.

Diagramme de cas d'utilisation



Nous avons trois acteurs ici, le premier **Coureur**, le deuxième **Organisateur** et le troisième **Application de Gestion**.

Avant le départ de la course :

L'acteur **Coureur** a comme cas d'utilisation principal « Récupérer dossard » et pour pouvoir récupérer son dossard il doit obligatoirement passer à « l'enregistrement » qui est un cas d'utilisation principal de l'acteur **Organisateur**. Et donc l'**organisateur** pour pouvoir lui donner son dossard doit faire grâce à une application « l'association du badge RFID au numéro dossard et au nom du coureur » et cela inclus le cas d'utilisation « Vérifier si coureur bien inscrit dans la base de données » afin de vérifier si le coureur s'est bien inscrit à la course.

Ensuite l'acteur **Application de Gestion** lui a deux principaux cas d'utilisation, le premier avant le départ de la course, il doit « Transmettre la liste des numéros RFID et l'heure courante aux stations de chronométrage »

Pendant la course :

Et le deuxième pendant la course **l'Application de Gestion** doit « Enregistrer le temps de passage des coureurs dans la base de données aux différents points de passage (stations de chronométrage) et cela inclus les cas d'utilisation « Recevoir numéro RFID, heure passage et numéro station » et « Identifier coureur ».

Diagramme de séquence (association RFID)

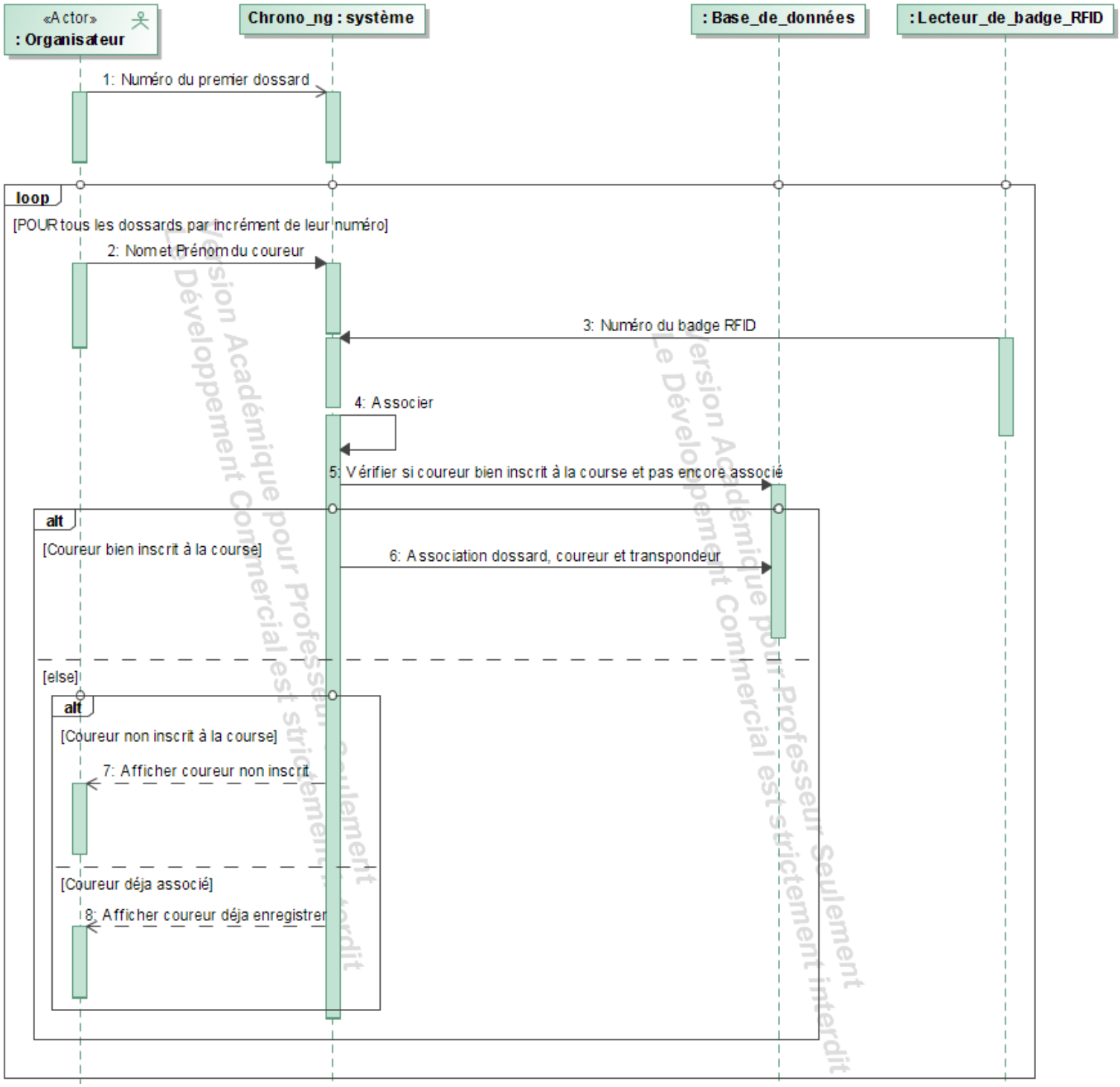


Diagramme d'exigences

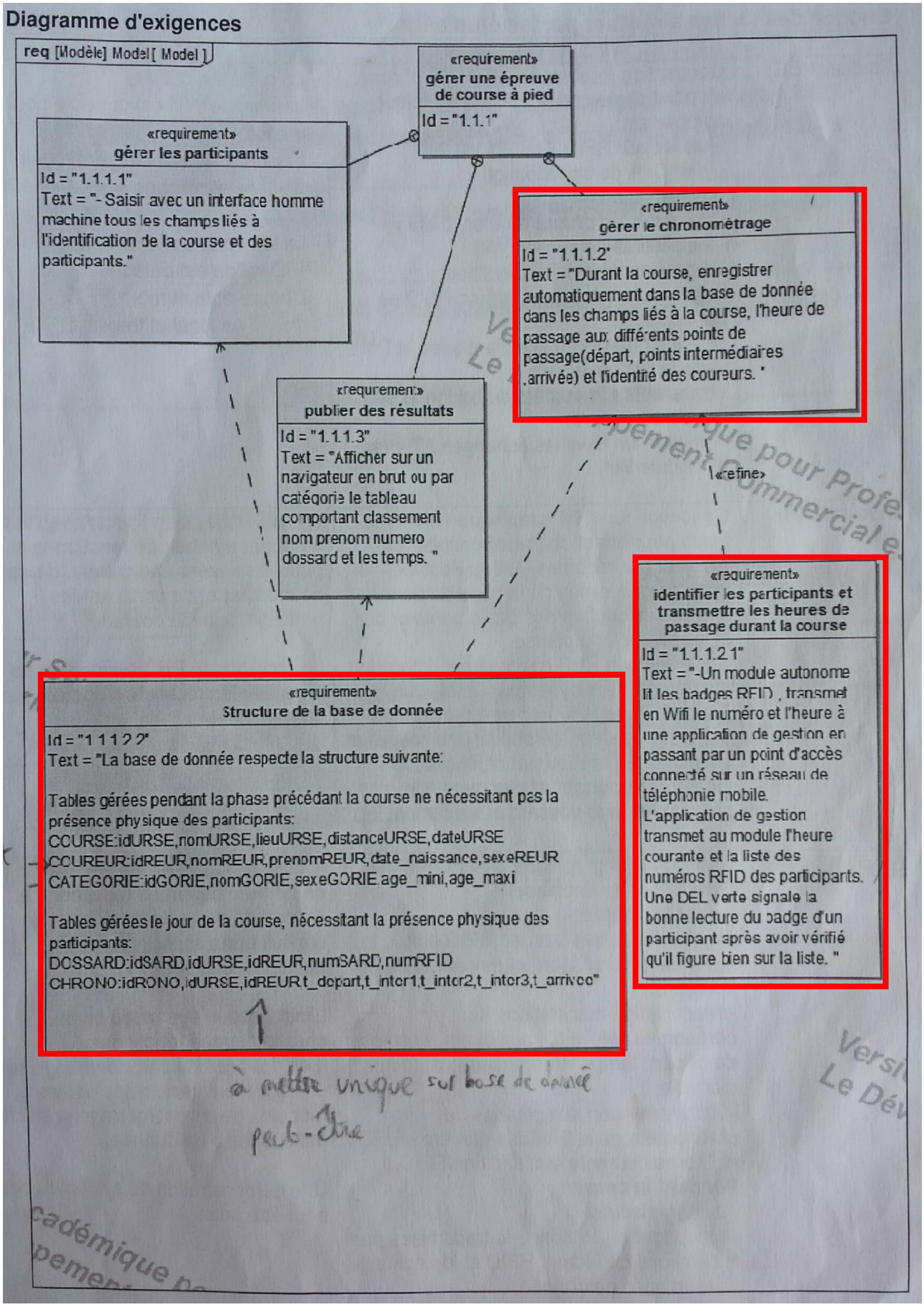
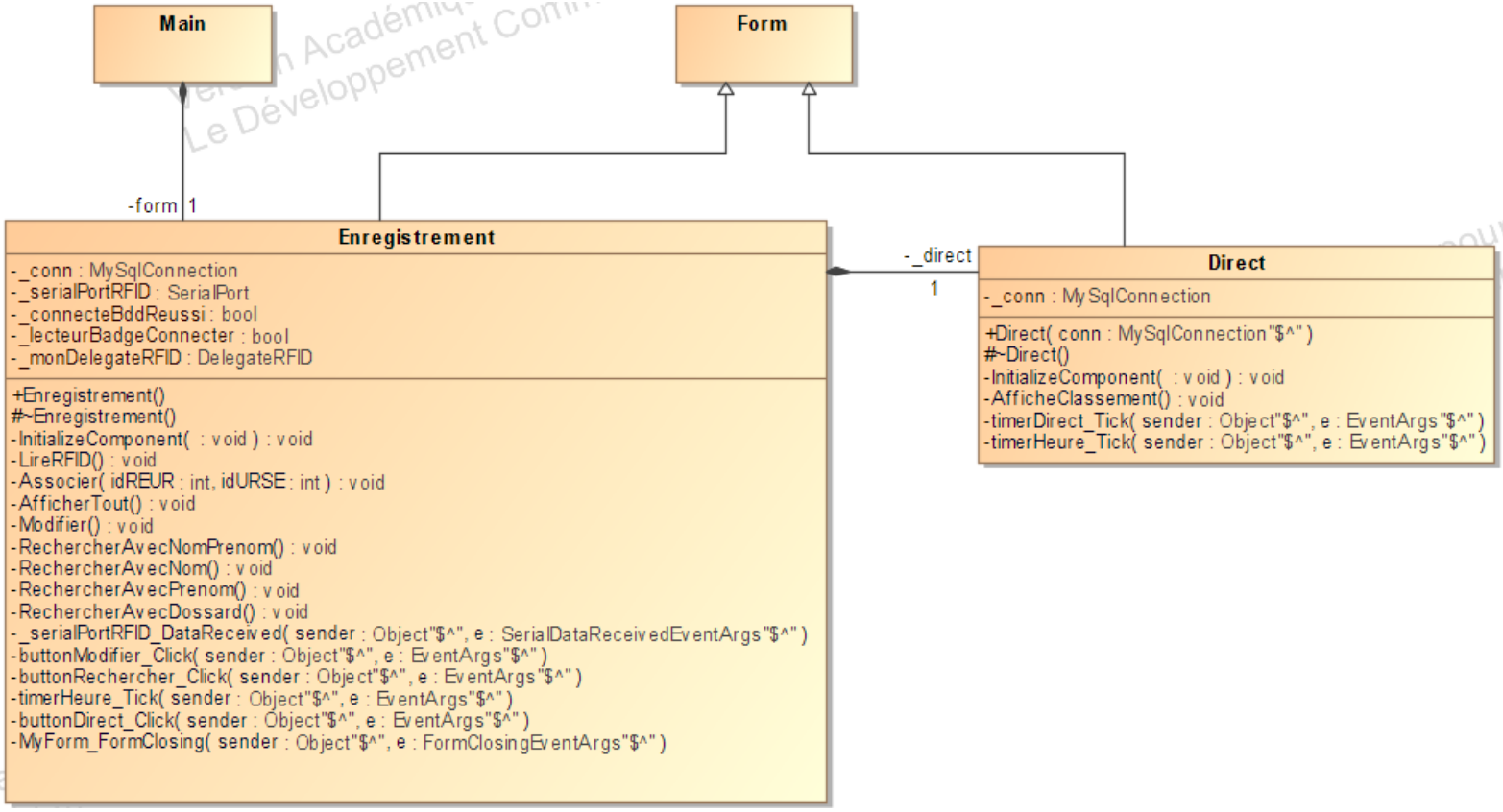
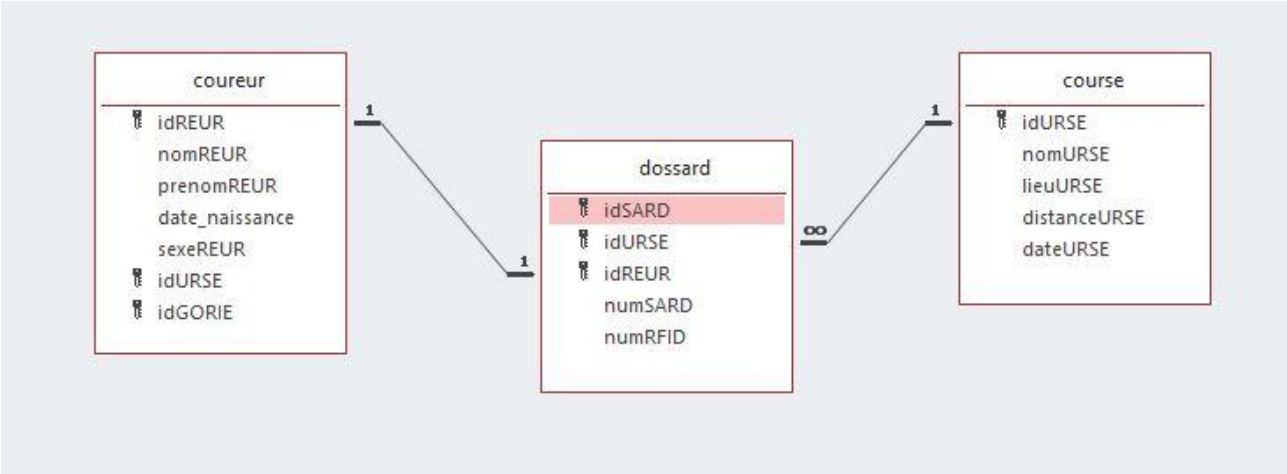


Diagramme de classe



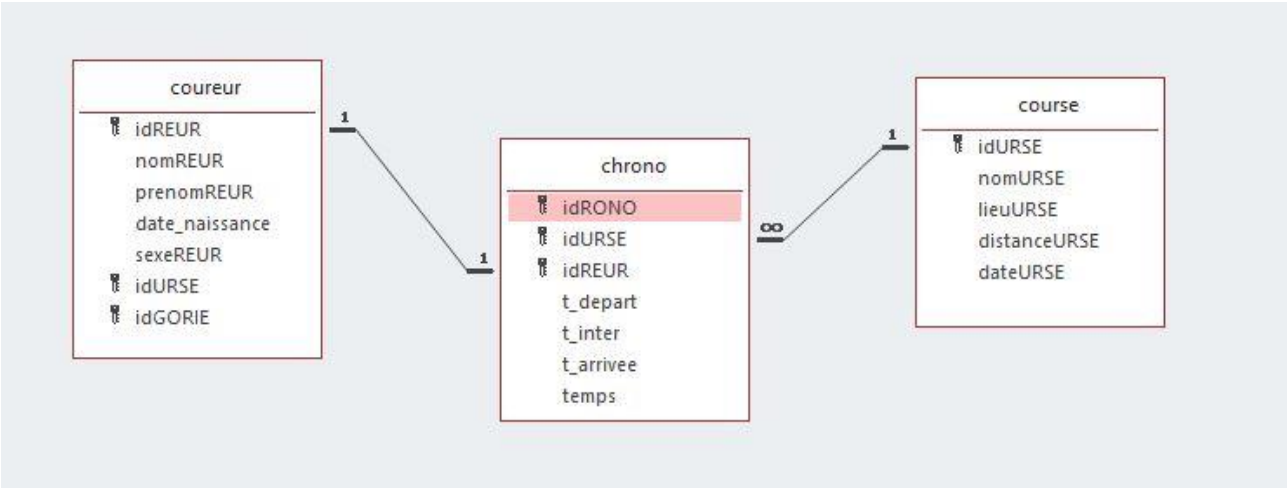
Relations entre les tables de la base de données chrono_ng

Pour pouvoir enregistrer les informations concernant l'association des badges RFID aux coureurs, aux numéros des dossards et l'enregistrement des heures de passage des coureurs à différentes stations de chronométrage, nous avons mis en place une base de données de nom **chrono_ng** qui est hébergé sur un serveur distant.



Un **coureur** a un seul **dossard** et un **dossard** appartient à un seul **coureur**.

Un **dossard** appartient à une seule **course** tandis qu'une **course** peut avoir plusieurs **dossards**.



Un **coureur** a un seul **chrono** et un **chrono** appartient à un seul **coureur**.

Un **chrono** appartient à une seule **course** tandis qu'une **course** peut avoir plusieurs **chronos**.

Avantages du chronométrage automatique

Les avantages du chronométrage automatique :
- Pour les organisateurs :
  - ✓ Economie de temps
  - ✓ Fiabilité
  - ✓ Information en temps réel
  - ✓ Plus besoin d'avoir des personnes pour noter l'heure de passage des coureurs (t_inter, t_arrivée)

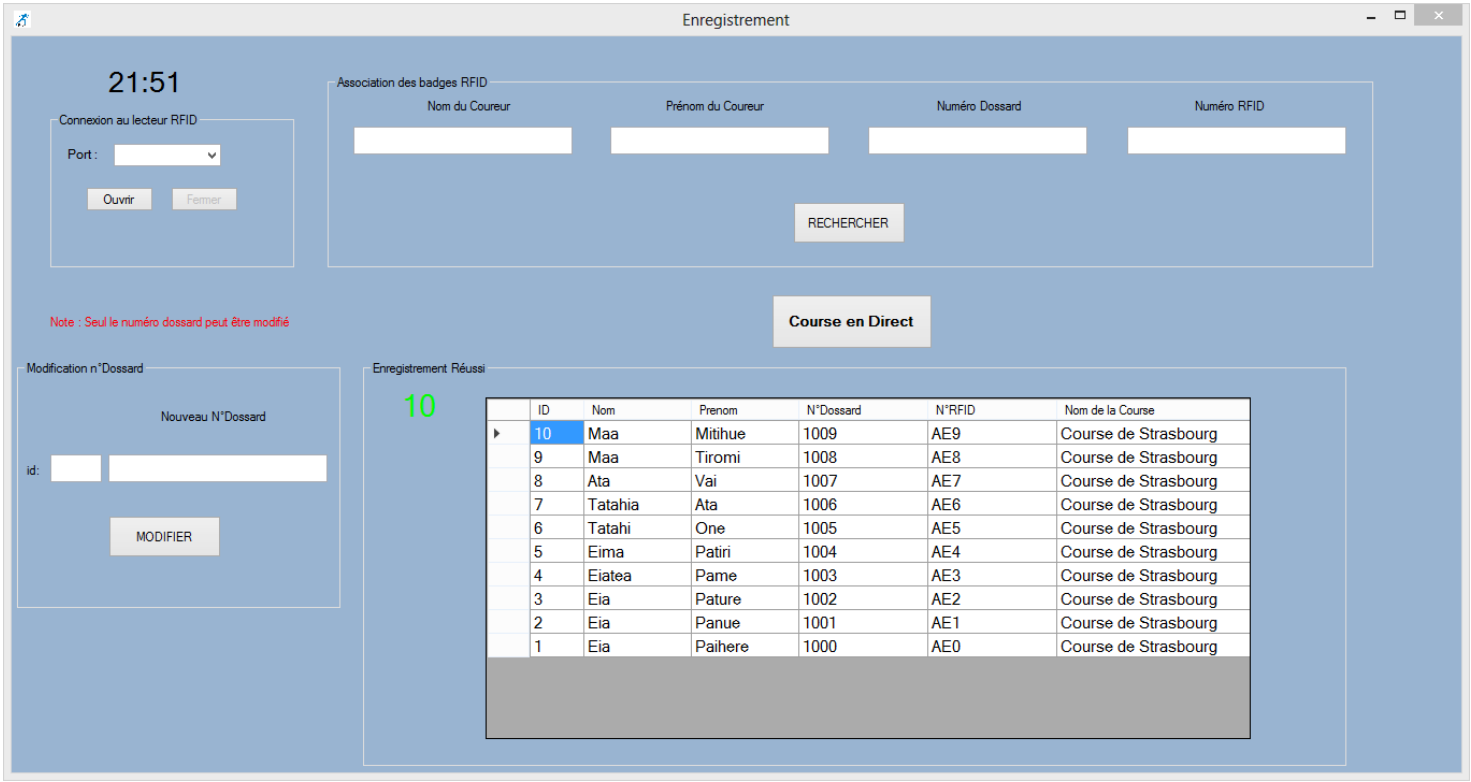
- Pour les coureurs :
  - ✓ Mise à disposition rapide et fiable des classements
  - ✓ Pas de désavantage en partant en dernier peloton car le temps de départ de chaque coureur sera déclenché quand il passera la ligne de départ (station t_départ)

- Pour le public :
  - ✓ Une publication instantanée des résultats

J'avais donc deux applications à créer, une application pour l'association des badges RFID aux dossards, aux noms des coureurs, et une autre application pour transmettre l'heure courante, la liste des numéros RFID associés, à toutes les stations de chronométrage et d'enregistrer automatiquement les temps de passage des coureurs à différentes stations.

L'Application d'association des badges RFID

Voici ci-dessous la première application que j'ai créée pour l'association des badges RFID. Je l'ai conçu avec le langage C++/CLI et c'est une application Windows, donc il ne peut être qu'utilisé que sur des ordinateurs qui ont un système d'exploitation Windows. Cette première application est une interface homme-machine (IHM) et donc l'utilisateur interagit avec l'ordinateur.



Comme sur le diagramme de cas d'utilisation qu'on a vu, un coureur pour pouvoir récupérer son dossard doit passer à l'enregistrement. Et à l'enregistrement l'organisateur aura cette application ouverte. Le coureur donne son nom et son prénom, l'organisateur le rentre sur l'application dans les champs concernés avec le numéro dossard et ensuite il badge le badge RFID sur le lecteur de badge relié au PC par USB pour pouvoir récupérer le numéro RFID sur l'application. La validation est automatique, il n'est pas nécessaire d'appuyer sur un bouton Enregistrer, l'enregistrement se fait quand on badge le badge RFID.

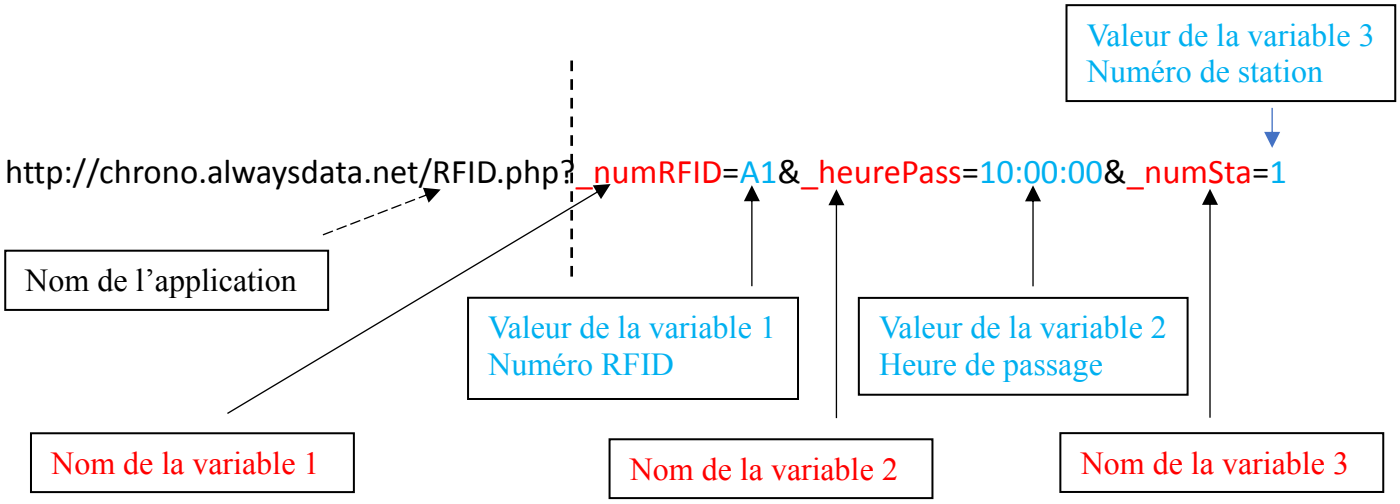
Si le coureur s'est bien inscrit, ces informations seront enregistrées dans la table « ***dossard*** » de la base de données « **chrono_ng** » et s'ajoutera dans le tableau de l'application en couleur jaune, sinon un message indiquant que le coureur ne s'est pas inscrit s'affichera. Et de même pour le numéro dossard, l'organisateur le rentre une seule fois, et par la suite l'application l'incrémentera (additionner par 1), mais dans ce cas il faudra que les dossards soit rangé dans l'ordre croissant de leur numéro dans un bac. Sur l'application l'organisateur pourra aussi rechercher un coureur soit par son nom et prénom ou bien par son dossard. Il pourra aussi modifier le numéro de dossard d'un coureur et suivre en temps réel la course.

L'Application de Gestion

Maintenant je vais vous présenter la deuxième application que j'ai créé avec le langage PHP. Le nom de l'application est « ***RFID.php*** ». Elle est hébergée sur un serveur distant chez Alwaysdata, un hébergeur et est accessible à l'adresse web « http://chrono.alwaysdata.net/RFID.php ». Cette application permet de transmettre aux stations de chronométrage 5 minutes avant le départ de la course l'heure courante pour initialiser l'heure des stations, la liste des numéros RFID associés et pendant la course d'enregistrer automatiquement le temps de passage des coureurs à différentes stations dans la table ***chrono*** dans la base de données.

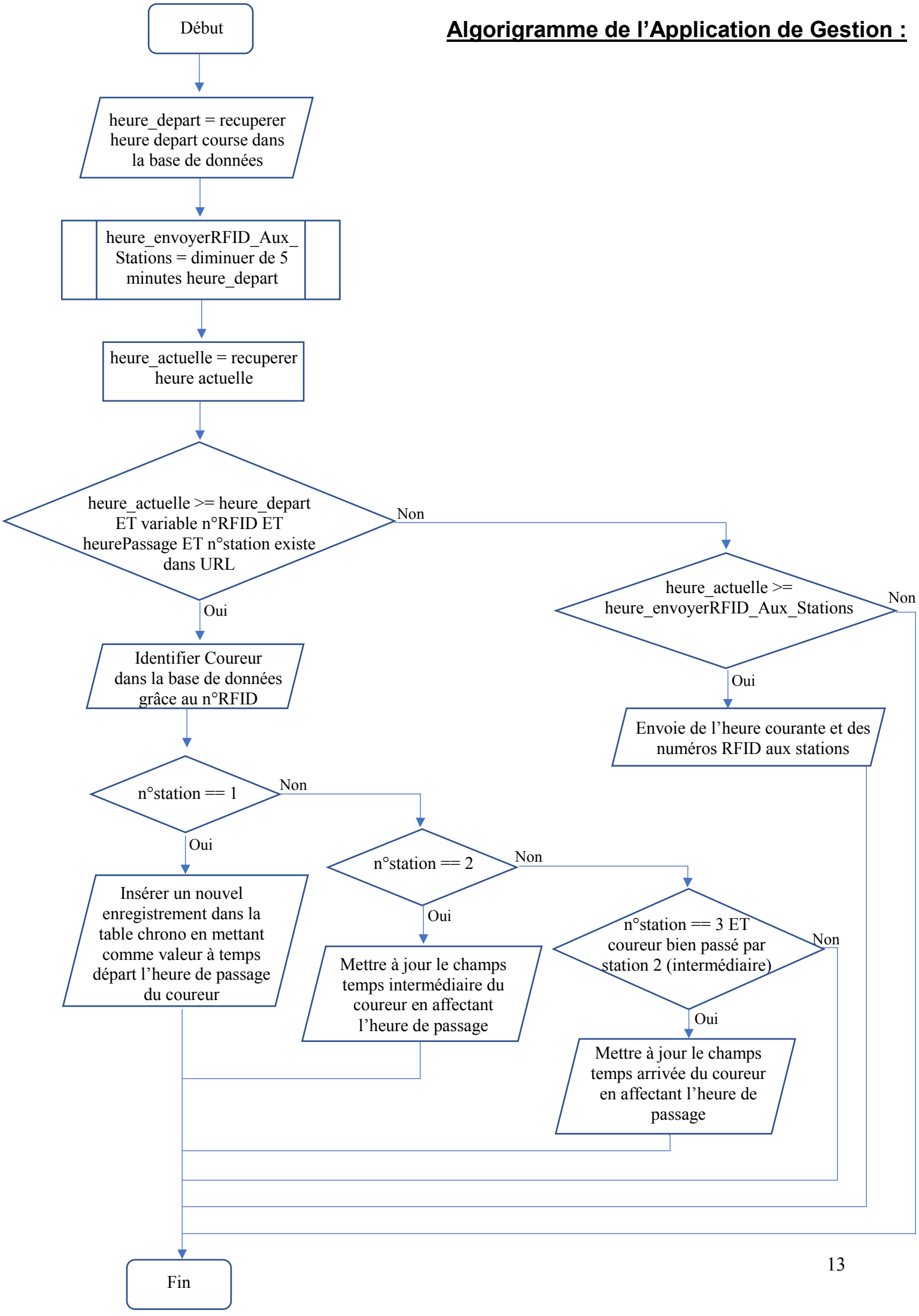
```php
<?php include('connectBDD.php'); //Connexion à la base de données ?>
<?php
$date = date('Y-m-d'); // Récupération de la date d'aujourd'hui
$requete = $bdd->query("SELECT h_depart FROM course where dateURSE='$date'"); // requête pour récupérer l'heure de départ de la course
$recupHeure_depart = $requete->fetch();
$heure_depart = $recupHeure_depart['h_depart'];
// Diminution heure départ de 5 minutes pour envoyer aux stations heure courante et n°RFID 5 minutes avant départ
$heure_envoyerRFID_Aux_Stations = DiminuerDeCinqMinutes($heure_depart);
$heure = date('H:i:s'); // heure courante pour ensuite comparaison avec heure départ et heure diminuer
/*----------------# Enregistrement des temps passage des coureurs dans la base de données #-------------------------*/
// si heure départ atteint et toutes les variables sont présents dans l'URL
if ($heure >= $heure_depart && isset($_GET['_numRFID']) && isset($_GET['_heurePass']) && isset($_GET['_numSta']))
{
    $numRFID = $_GET['_numRFID']; // réception du numRFID transmis par la station
    $requete = $bdd->query("SELECT idURSE, idREUR FROM dossard WHERE numRFID='$numRFID'"); // identification du coureur
    $recup = $requete->fetch();
    $idURSE = $recup['idURSE']; // on récupère clé primaire idURSE du coureur pour clé étrangère dans la table chrono
    $idREUR = $recup['idREUR']; // pareil pour la course
    $heurePassage = $_GET['_heurePass']; // réception de l'heure de passage du coureur
    $numSta = $_GET['_numSta']; // réception du numéro de station
    if ($numSta == 1)// association de la station 1 au champ temps de départ de la table chrono
        $bdd->exec("INSERT INTO chrono(idURSE,idREUR,t_depart) VALUES('$idURSE','$idREUR', '$heurePassage')");
    else if ($numSta == 2)// association de la station 2 au champ temps intermédiaire
        $bdd->exec("UPDATE chrono SET t_inter='$heurePassage' WHERE idREUR='$idREUR'");
    else
    {   // requête pour vérifier que le coureur est  bien passé par la station intermédiaire
        $verif = $bdd->query("SELECT t_inter FROM chrono WHERE idREUR='$idREUR'");
        $recupVerif = $verif->fetch();
        // association de la station 3 au champ temps arrivée et vérification si coureur est bien passé par station intermédiaire
        if ($numSta == 3 && $recupVerif['t_inter'] != '00:00:00')
            $bdd->exec("UPDATE chrono SET t_arrivee='$heurePassage' WHERE idREUR='$idREUR'");
    }
}
/*------------------# Envoie de l'heure courante et des numéros RFID #-------------------------------*/
else if ($heure >= $heure_envoyerRFID_Aux_Stations) // si heure pour envoyer les numéros RFID aux stations est atteint ou dépassé
{
    $requete = $bdd->query("SELECT numRFID FROM dossard, course WHERE course.dateURSE='$date' && dossard.idURSE=course.idURSE");
    echo date('H:i:s').' '; // alors envoie de l'heure courante aux stations
    while ($donnees = $requete->fetch()){
        echo $donnees['numRFID'].' '; // envoie des numéros RFID aux stations
    }
}
/*-----------#  Fonction pour diminuer l'heure de départ de 5 minutes #---------------*/
function DiminuerDeCinqMinutes($heure)
{
    $h = $heure[0].$heure[1];    // on récupère l'heures de départ
    $m = $heure[3].$heure[4];    // on récupère les minutes de départ
    for ($i=0; $i < 5; $i++)
    {
        if ($m == 00) // si la minutes est égal à 00
        {
            $m = 59; // alors on lui affecte 59, puisque -1 n'existe pas comme minute
            $h--; // on soustrait l'heure par 1
            if ($h >= 0 && $h <= 9) // si heure soustrait est entre 0 et 9
                $h = '0'.$h; // alors on ajoute un 0 devant la nouvelle heure pour comparaison
        }                    // car après soustraction on a qu'un seul chiffre
        else // sinon
        {
            $m--; // on soustrait que les minutes
        }
    }
    if ($m >= 0 && $m <= 9) // si minute soustrait est entre 0 et 9
        $m = '0'.$m; // alors on ajoute un 0 devant la nouvelle minute pour comparaison
    return $h.':'.$m.':00';
}
?>
```

La méthode utilisé ici est la méthode GET. Cette méthode permet aux stations de transmettre des données à l'application avec l'URL. Donc chaque station aura cette URL dans son programme.

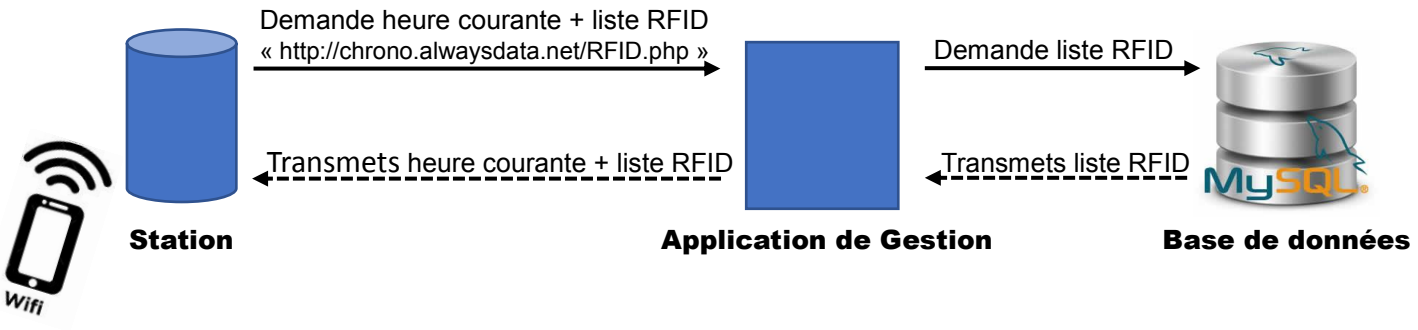


Les informations après le point d'interrogation seront des données que l'application de gestion recevra des stations. Dans cette URL on a des variables qui vont contenir des valeurs. La variable 1 (_numRFID) contiendra le numéro RFID transmis par la station, puis la variable 2 (_heurePass) l'heure de passage du coureur et la variable 3 (_numSta) le numéro de station pour savoir quelle station a envoyé ses données. La station 1 pour le temps de départ, la station 2 pour le temps intermédiaire et la station 3 pour le temps d'arrivée.
Les instructions du programme sont englobées par des conditions, « **if** » et « **else** ». Toutes les instructions dans la condition « if » (ligne 12 à ligne 33) sont pour la réception des données transmises par les stations (numéro RFID, heure passage, numéro station), l'identification du coureur, et l'enregistrement dans la base de données de l'heure de passage du coureur. Ensuite toutes les instructions dans la condition « else » (ligne 36 à ligne 43) permet de transmettre l'heure courante et les numéros RFID aux stations. Pour faire simple les stations vont pouvoir recevoir l'heure courante et les numéros RFID 5 minutes avant le départ de la course en se connectant à l'URL « chrono.alwaysdata.net/RFID.php », et donc ce seront les instructions dans la condition « else » qui seront exécutées et pendant la course les stations transmettront des données à l'application de gestion en se connectant au même URL mais avec des variables et des valeurs, et là ce seront les instructions dans la condition « if » qui seront exécutées. Prenons un exemple d'un coureur A qui a comme numéro RFID 10 et qui passe à 9h30 devant la station 2 (temps intermédiaire). La station 2 transmettra dans ce cas des données dans l'URL comme celui-ci : chrono.alwaysdata.net/RFID.php?_numRFID=10&_heurePass=09:30:00&_numSta=2

Algorigramme de l'Application de Gestion :

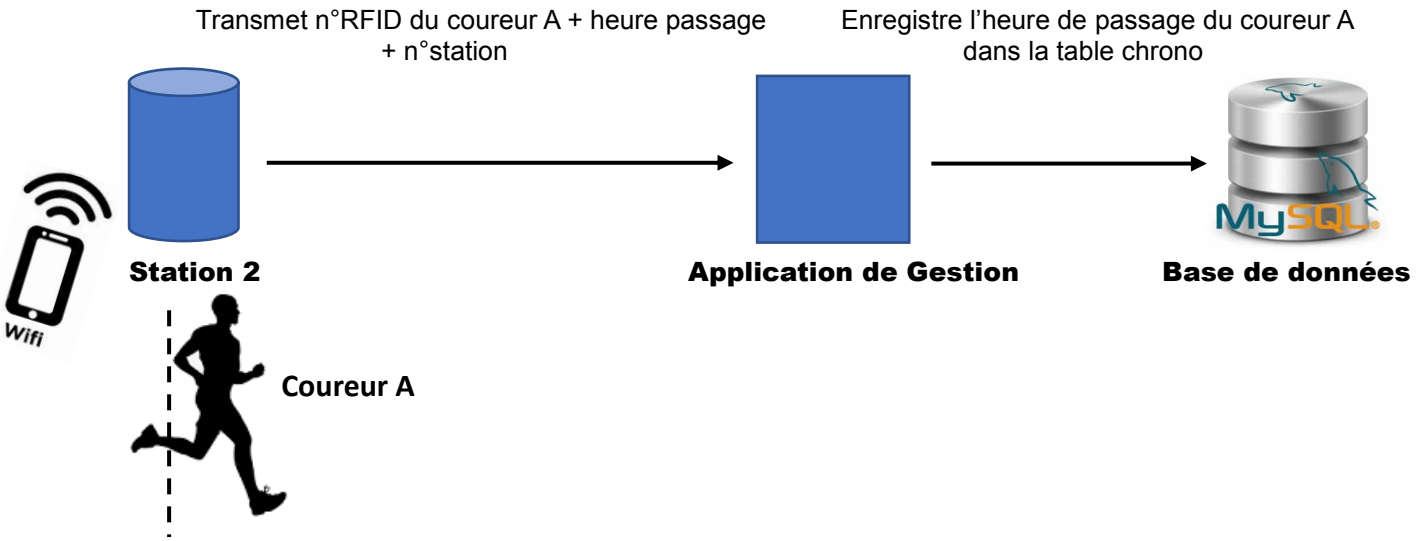


Envoie de l'heure courante et de la liste des numéros RFID à toutes les stations avant le départ :



Le schéma ci-dessus montre le déroulement de l'envoie de l'heure courante et des numéros RFID aux stations avant le départ de la course. L'Application de Gestion joue l'intermédiaire entre les stations et la base de données.
Les **Stations** font une demande avec l'URL « chrono.alwaysdata.net/RFID.php » de l'heure courante et de la liste des numéros RFID à l'**Application de Gestion**. L'**Application de Gestion** fait ensuite la demande des numéros RFID de la course d'aujourd'hui à la **Base de données** qui lui transmets en retour la liste. Et enfin l'**Application de Gestion** transmets en retour aux stations l'heure courante et la liste des numéros RFID.

Réception des données transmises par les stations et écriture dans la base de données



Le schéma ci-dessus montre le déroulement de l'enregistrement de l'heure de passage d'un coureur à une station dans la table chrono de la base de données. Reprenons l'exemple qu'on a pris précédemment du coureur A qui a comme numéro RFID 10 et qui passe à 9h30 devant la station 2 (temps intermédiaire). Au passage du **Coureur A** devant la **Station 2**, son numéro RFID (10) est récupéré par la station et ensuite transmis avec l'heure de passage (09:30:00) et le numéro de station (2) à l'**Application de Gestion**. L'**Application de Gestion** identifie le coureur et enregistre son heure de passage dans la table « chrono » dans la base de données.

CONCLUSION

Pour conclure le projet avance très bien, mais n'est pas complètement opérationnel, il nous manque maintenant de faire des tests entre les différentes parties du système pour vérifier au bon fonctionnement et à la correction des erreurs.

Pour ce qui est de ma part, ce projet m'a permis de mettre en pratique tout ce dont j'ai appris ces deux années en BTS, de pouvoir réaliser mon premier projet et de le travailler en équipe.

Je suis arrivé en BTS SN-IR avec un bagage de connaissances léger, j'en repars maintenant avec un bagage plus lourd et plus solide.

REMERCIEMENTS

Je tiens à remercier les professeurs de BTS SN du lycée Louis Couffignal pour m'avoir formé et accepté dans cette formation très formateur.