

# Projet LPSIL IDSE 2014-2015 - Ateliers RFID - La Ronde des Facs



Table des matières :

I) Présentation du projet

Phase d'inscription :

Phase de course :

Phase de résultats :

II) Architecture proposée

Phase d'inscription

2) Phase de course

3) Phase de résultats

III) Dossier de conception

1) Badger lors de la course.

2) Statistiques et KPIs

3) Pré-inscription en ligne

4) Modifier les données

5) Se connecter

6) Générer le classement

7) Associer carte étudiante à son inscription

8) Inscription sur place

9) Structure de la base de données

IV) Organisation

Planning prévisionnel.

2) Répartition des tâches.

V) Résultats

Fonctionnalités

2) Module BdD

3) Module Site

4) Module Web Services

5) Plannings

6) Difficultés rencontrés

7) Experience

8) Reste à faire

## I) Présentation du projet

La ronde des facs est un événement organisé par les associations étudiants et le personnel de l'Université Nice Sophia Antipolis (UNS). Cet événement solidaire sportif et convivial, soutenu par la ville de Nice, rassemble chaque année sur une demi-journée l'ensemble de la communauté universitaire : étudiant et personnels, universitaires ou assimilés.

Cet événement est représentatif de la force, la vitalité et l'unité de l'Université à travers la ville de Nice et le département des Alpes Maritimes ; il est possible grâce à la mobilisation de nombreux partenaires publics et privés.

En 2015, plus de 2500 participants étaient attendus sur le trottoir sud de la Promenade des Anglais pour marcher ou courir les 3.5 km qui séparent le théâtre de verdure de la plage de Carras. Un challenge intercampus permet à chaque campus de s'affronter et de défendre les couleurs de son équipe. La course a lieu un jeudi après-midi, demi\*journée durant laquelle les étudiants peuvent se consacrer à la pratique sportive et aux enseignements libre.

L'objectif de se projet est d'apporter une solution informatique à la logistique de cet événement découpé en trois phases :

### 1) Phase d'inscription :

Les participants auront la possibilité de s'inscrire en ligne avant le jour de la course ou sur place le jour même de la course avant le début de celle-ci (inscription de dernière minute). Leurs informations seront descendus d'une base de données LDAP de l'UNS sous soumission de leur identifiant UNS et un compte leur sera créée sur le site leur permettant ainsi de retrouver leurs résultats. Dans le cas où le participant voudra courir lors de la course, celui-ci devra fournir un certificat médical. Une fois sur place le jour de la course, le participant associera sa fiche d'inscription à sa carte RFID (identifiant *mifare*) permettant ainsi de regrouper les résultats de la course.

### 2) Phase de course :

Les participants partiront les uns après les autres (couloir de départ). Ils devront indiquer leur passage sur des bornes RFID à l'aide de leur carte UNS sur les deux relais présents sur la course séparés par 2 km non connectés à internet (un premier relais présent au départ et l'autre à mi-parcours). L'oubli de passage sur un des relais disqualifie le participant. Il est donc important d'indiquer à celui-ci qu'il a bien été

identifié lors de son passage à un relais. L'écran affichera donc un message de confirmation ainsi que les données relatives au participant concerné. Le nombre de boucle ainsi effectué permettra de déterminer les vainqueurs. Dans le cas où plusieurs participants se verraient attribués le même nombre de boucle, le temps qu'ils leur aura fallu pour les faire permettra de les départager.

### **3) Phase de résultats :**

Une fois la course terminée, il faudra réunir les données de passage des relais et fournir les résultats associés (KPI). Ces résultats seront fournis sur le site d'inscription. Le site fournira des résultats tel que :

- Classement général,
- Classement par sexe,
- Classement par groupe, établissement, département, etc.

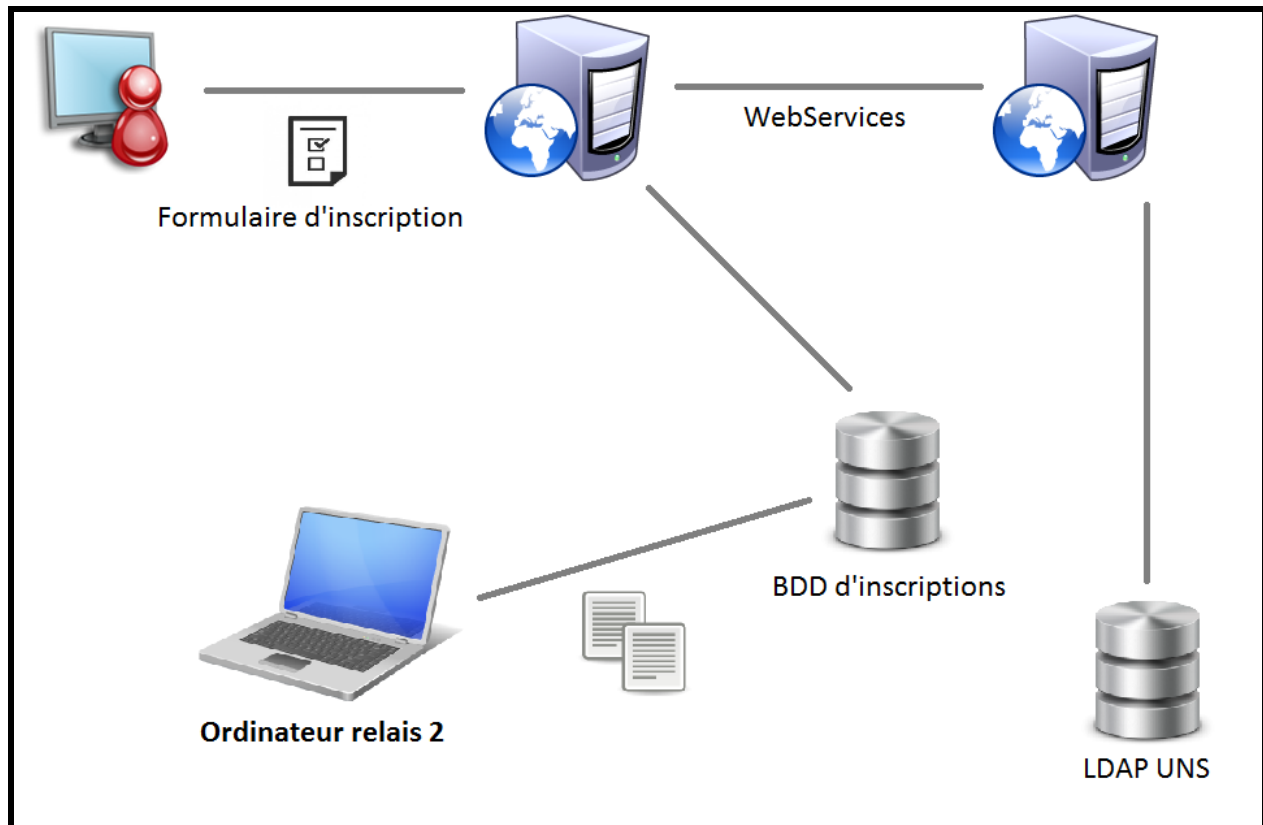
Et des statistiques tel que :

- Vitesse moyenne de course,
- Nombre de tours,
- Meilleur temps au tour,
- Distance parcourue, etc.

Si le participant se connecte sous son compte (créée lors de son inscription), ses résultats lui seront directement affichés.

## II) Architecture proposée

### 1) Phase d'inscription

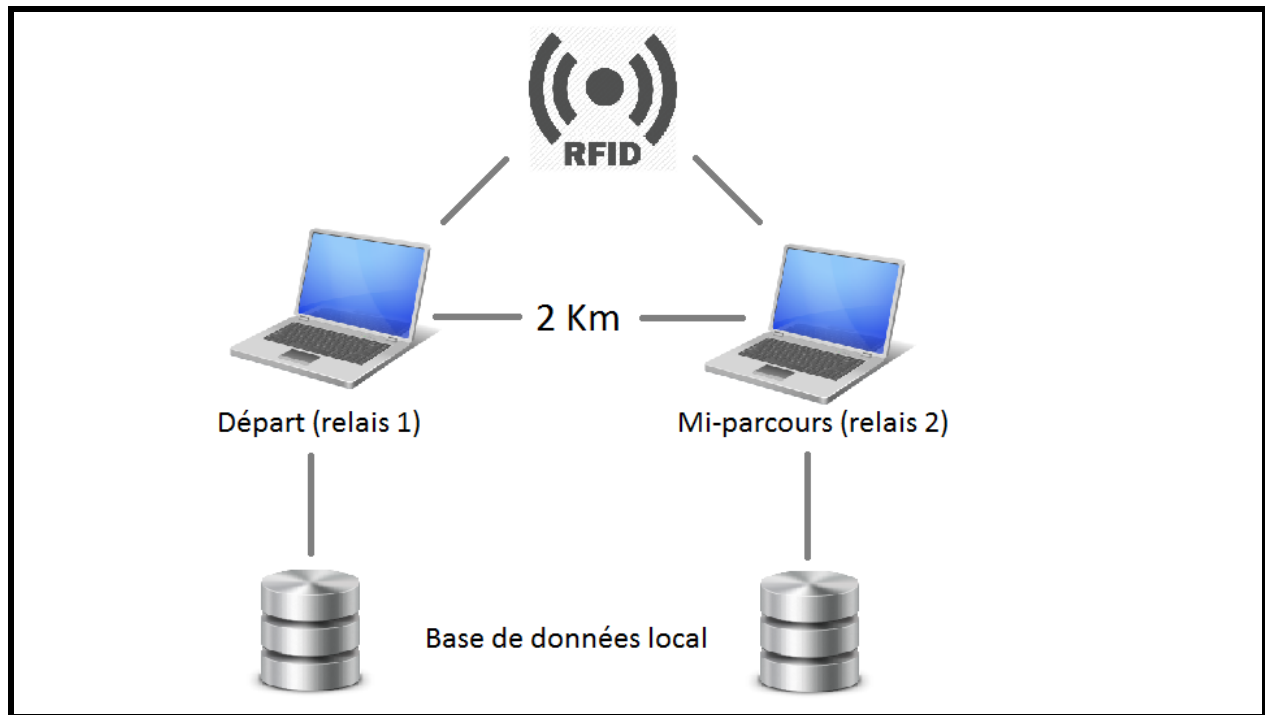


*Schéma d'architecture phase d'inscription*

Dans le cas de la phase d'inscription, le participant devra soumettre un formulaire d'inscription en ligne (**utilisation des langages du web HTML/CSS/PHP... sur un serveur web dédié**). Pour l'aider, un WebServices sera mis en place pour extraire les données de la base LDAP de l'UNS (**utilisation d'un serveur Tomcat et de WebServices en Java/JDBC**). Le résultat retourné permettra au serveur web de préremplir les champs du formulaire. Une fois le formulaire validé, la base de données liés aux inscriptions sera mise à jour (**base de données MySQL**). L'ordinateur qui devra être amené au relais 2 possède quant à lui un backup de cette base en local. Une fois les inscriptions clôturées, sa base de données changera d'état pour devenir autonome. Cette solution est applicable à la fois pour l'inscription précédant la journée de la course (le participant peut soumettre le formulaire en ligne directement) et les inscriptions de dernières minutes (les serveurs seront locaux et le navigateur pointera en localhost). Avant la

course, les participants devront badger une première fois pour valider leur inscription et associer leur identifiant mifare.

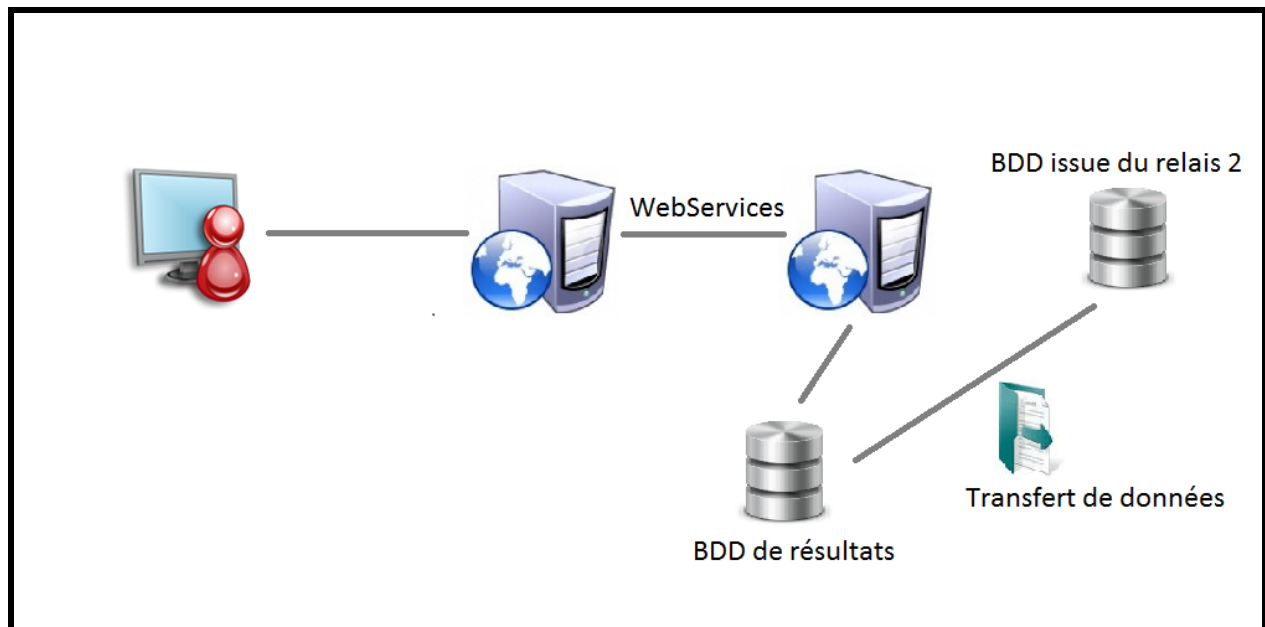
## 2) Phase de course



*Schéma d'architecture phase de course*

Lors de la course, les participants utiliseront leur carte UNS (dotée d'une balise RFID) sur les relais du parcours. Une application native (**utilisation des langages C/C++**) prendra en charge de réceptionner la trame RFID et de mettre à jour une table dédiée avec les données du passage. Les deux relais ne seront pas liés à internet et posséderont une base de données local. Lors de son passage, le participant verra ses données affichées sur l'écran de manière à confirmer son passage.

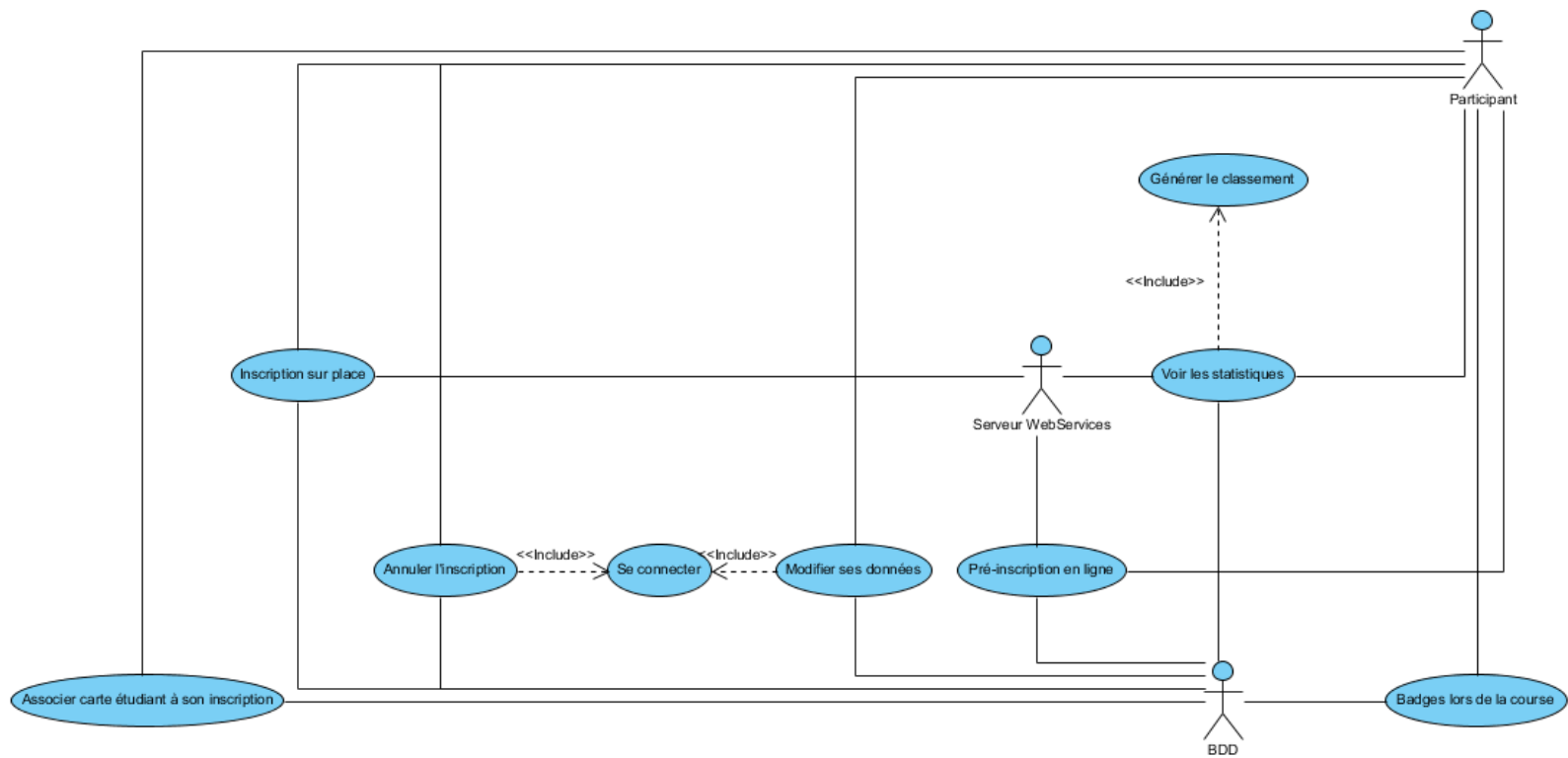
### 3) Phase de résultats



*Schéma d'architecture de la phase de résultats*

Une fois la course terminée, les données de la base issue du relais 2 seront transférées vers la base de données principale. Un algorithme de filtrage sera déployé pour éviter les données erronées (**utilisation de procédures stockées**). Les données ainsi validées pourront faire l'objet d'un traitement par procédures stockées dont les résultats seront transformés et remontés par WebServices. Dans le cas où le participant se sera identifié sur le compte créé lors de son inscription, les résultats seront centrés sur ses performances.

### III) Dossier de conception



*Diagramme de Use Case du projet*



### 1) Badger lors de la course.

Du fait de la lenteur de lecture des puces NFC et de leur faible portée, les étudiants participant auront l'obligation de ralentir afin de pouvoir "badger" ce qui aura pour effet d'enregistrer son heure de passage dans la copie local de la base de données. Le diagramme qui suit détaille cette étape importante :

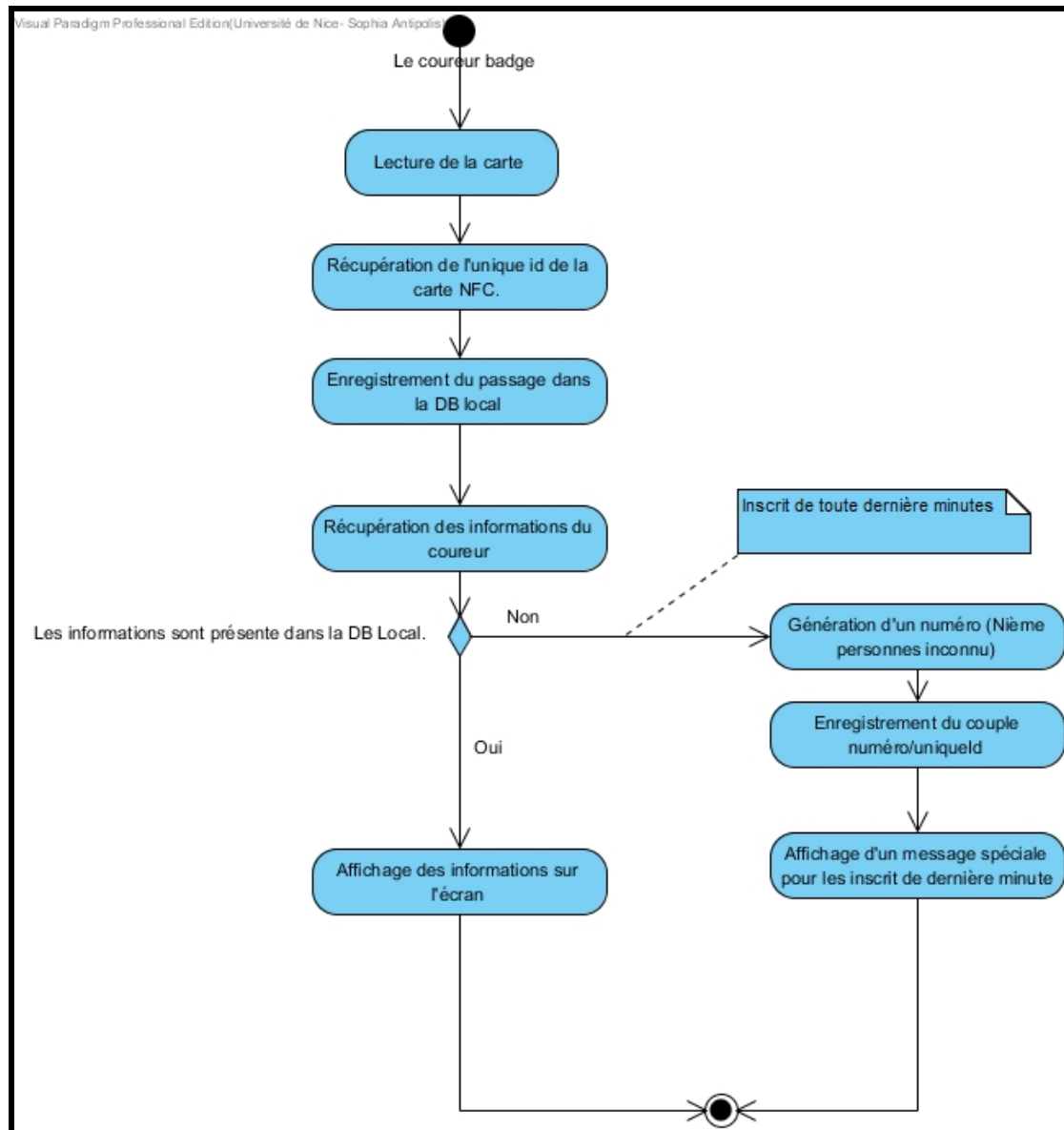


Diagramme de flux : Badger lors de la course

## 2) Statistiques et KPIs

Afin d'offrir une meilleur lisibilité des résultats et autres statistiques de la course, nous avons décidé de mettre en place un système de KPI avec la possibilité de filtrer les données à prendre en compte. Ce qui suit détaille le fonctionnement des différents outils liés aux statistiques :

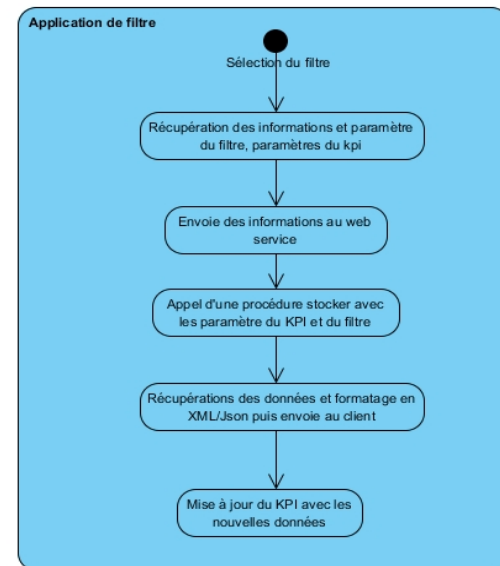
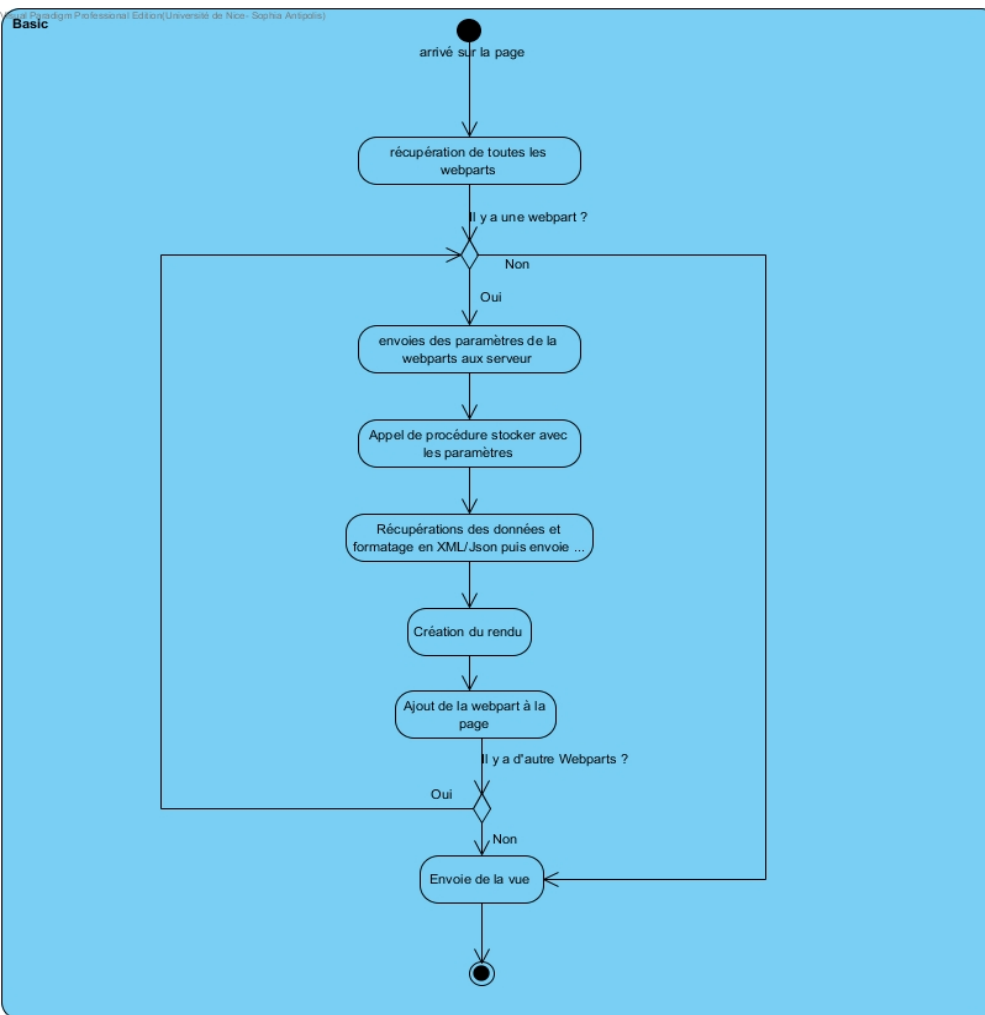


Diagramme de flux : Statistiques et KPIs

### 3) Pré-inscription en ligne

Pour une meilleur organisation de l'évènement, les participants auront la possibilité de s'inscrire au préalable sur le site internet.

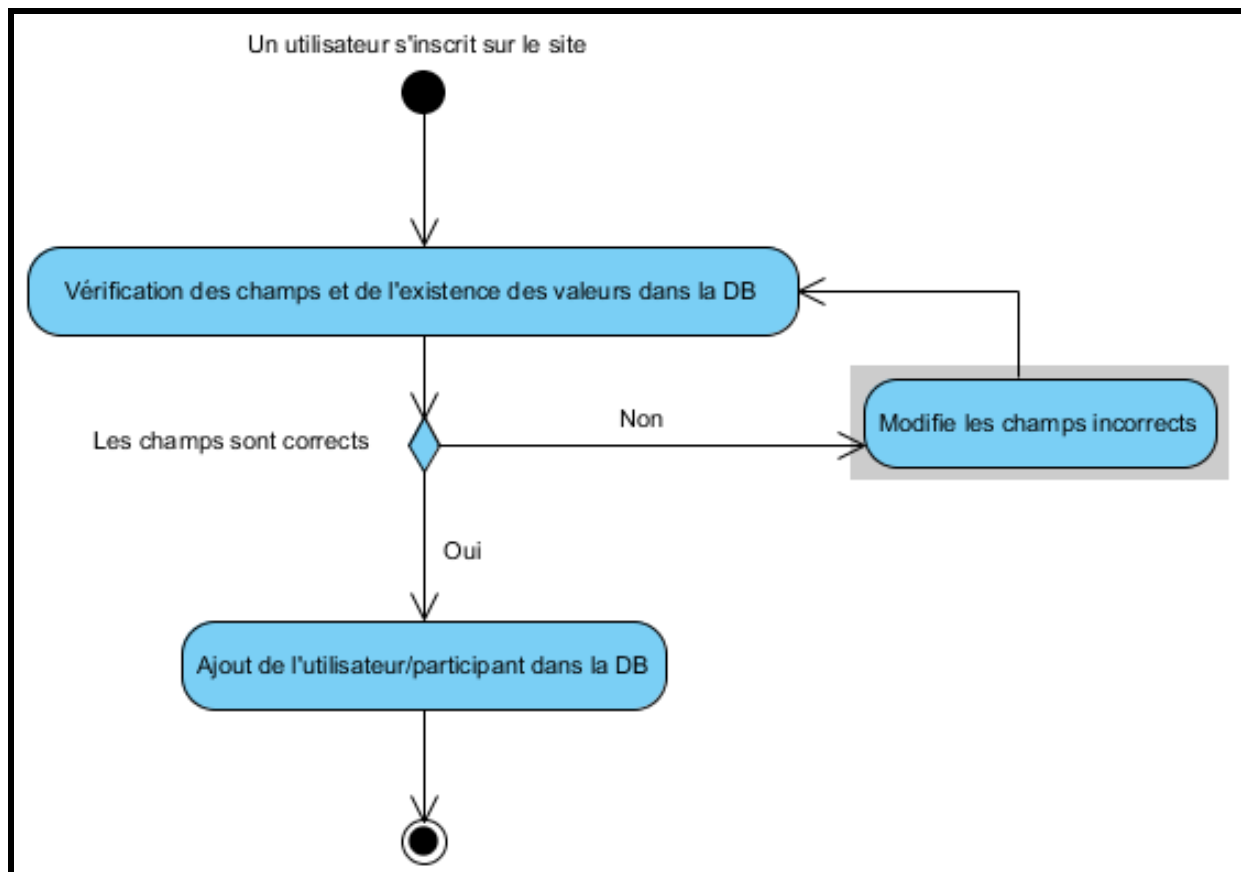
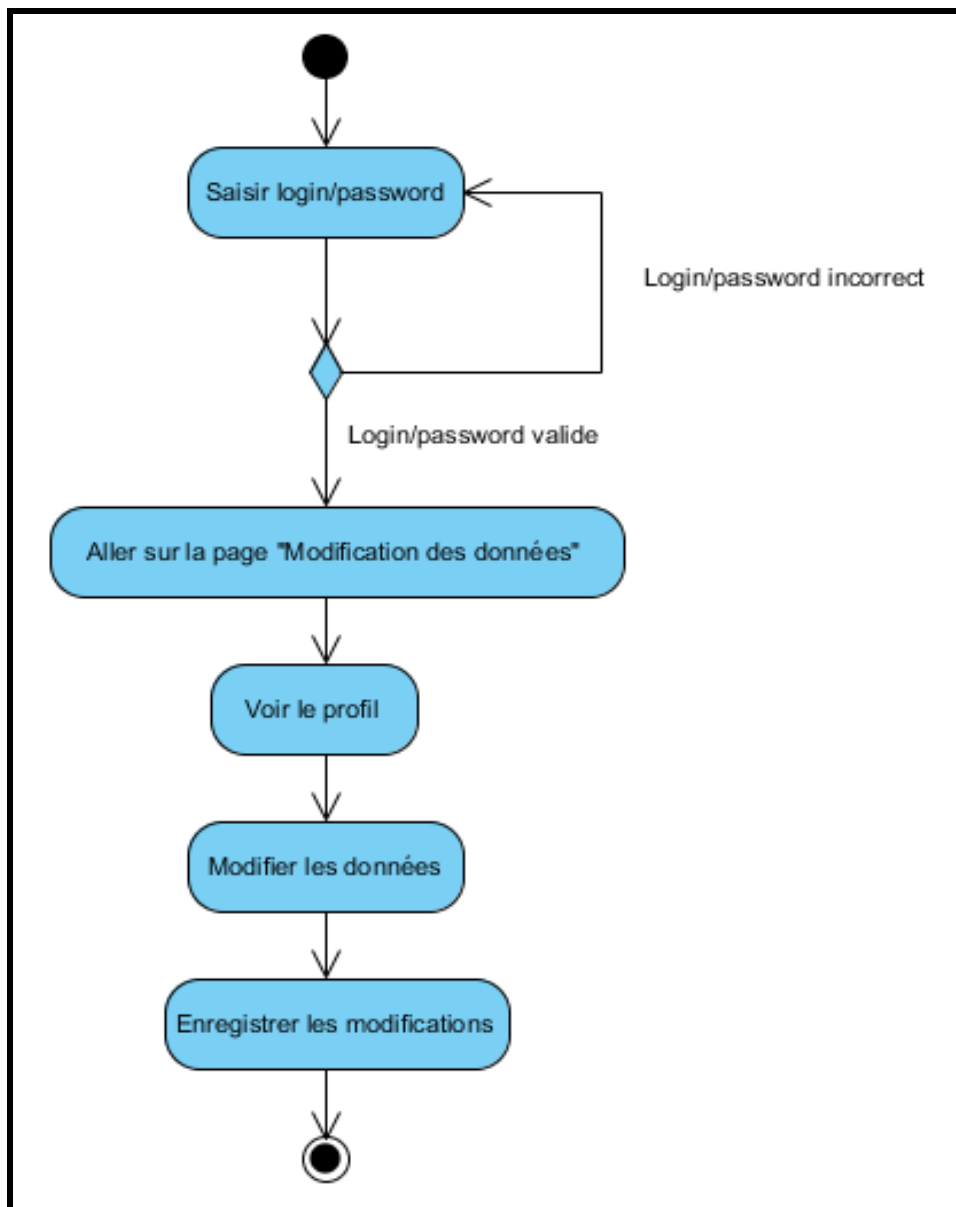


Diagramme de flux : Pré-inscription en ligne

#### 4) Modifier les données

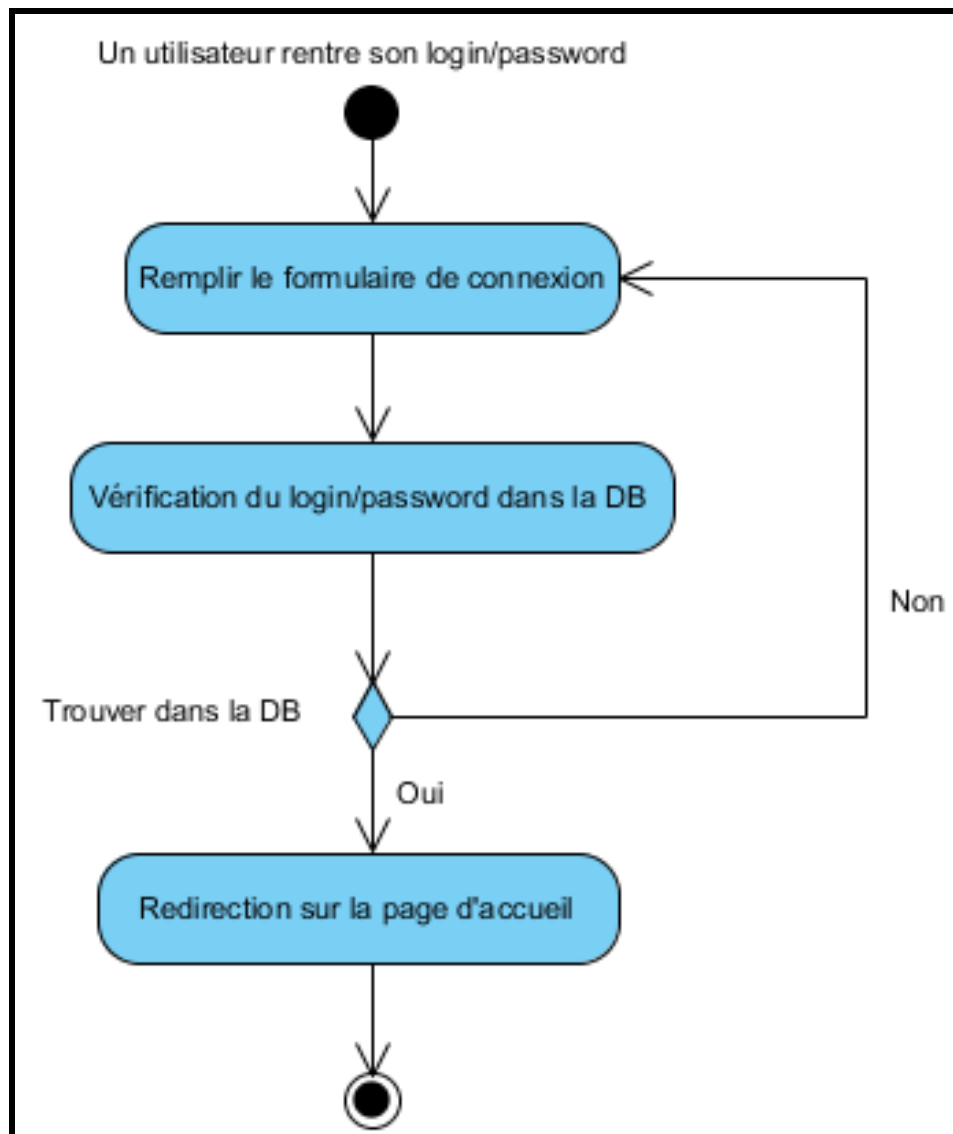
Si un participant souhaite modifier les données qu'il a insérées lors de son inscription, il a juste besoin de se rendre sur son profil, de modifier ses valeurs, et d'enregistrer.



*Diagramme de flux : Modifier les données*

## 5) Se connecter

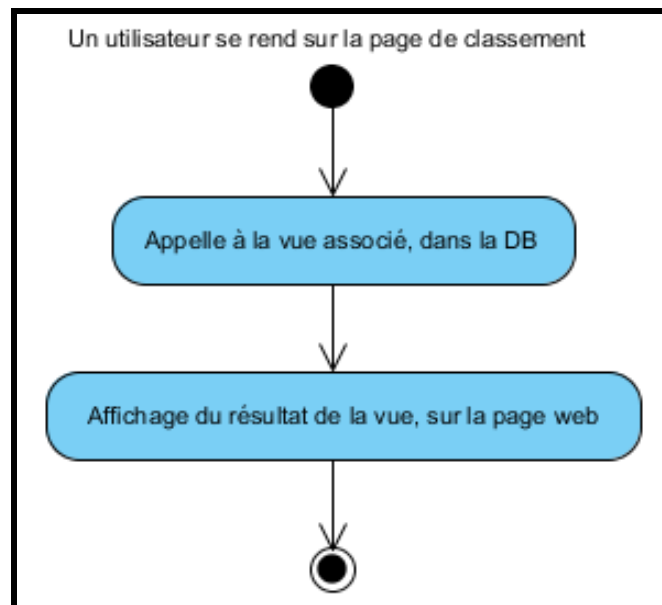
Un utilisateur peut se connecter sur le site pour accéder directement à son profil et sa place dans le classement.



*Diagramme de flux : Se connecter*

## 6) Générer le classement

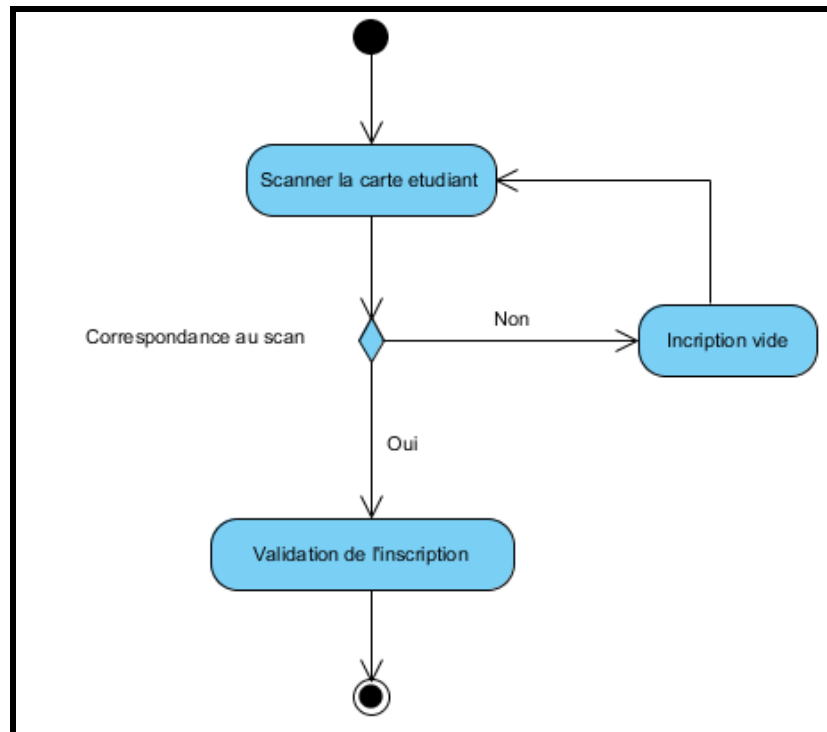
On peut se rendre sur la page de classement sur le site internet pour voir le classement des participant.



*Diagramme de flux : Générer le classement*

## 7) Associer carte étudiante à son inscription

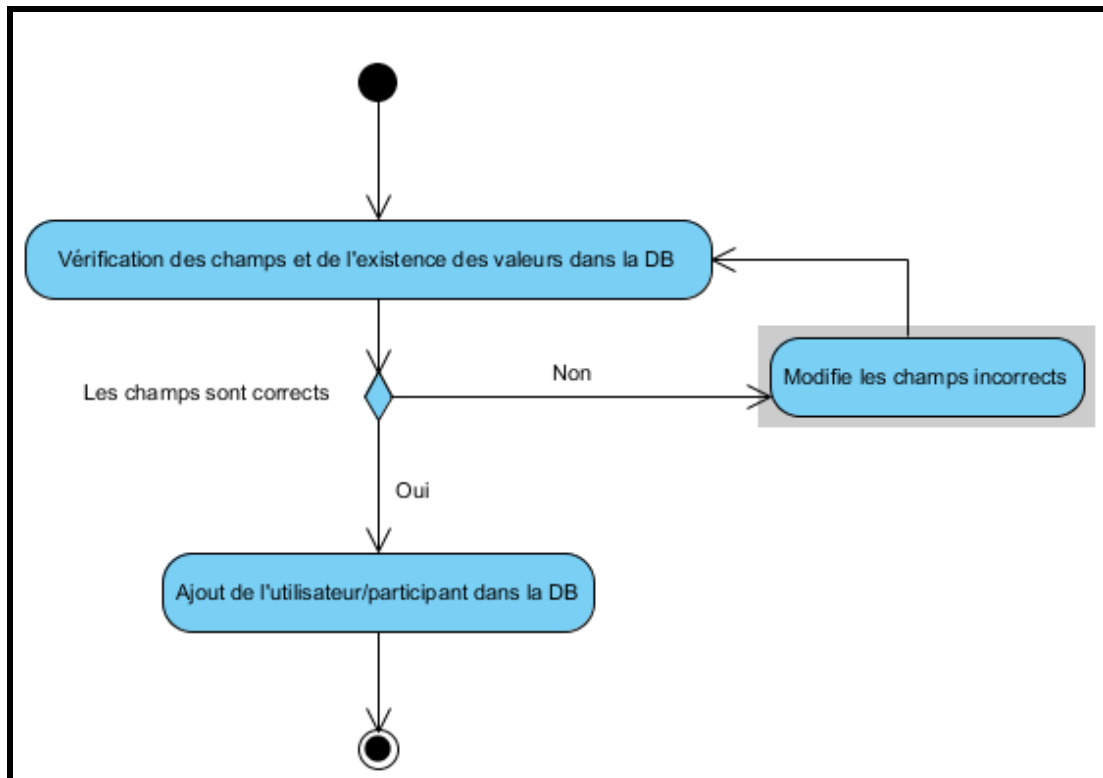
Avant la course le participant qui c'est inscrit auparavant doit scanner sa carte afin de confirmer son inscription pour participer à la course.



*Diagramme de flux : Associer sa carte étudiante à son inscription*

## 8) Inscription sur place

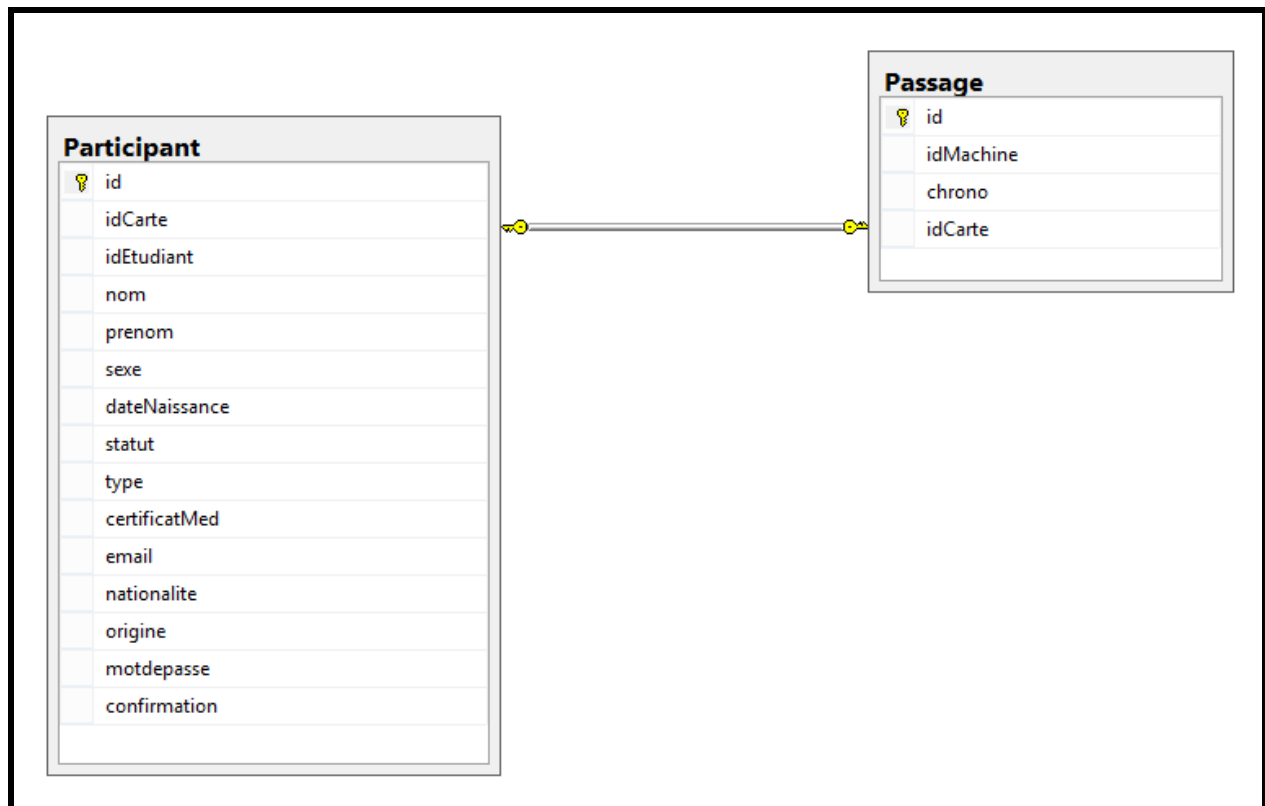
Si un participant souhaite s'inscrire au dernier moment il dispose du même formulaire d'inscription que sur le site internet sur le lieu de la course (en local).



*Diagramme de flux : Inscription sur place*



## 9) Structure de la base de données



*Structure de la base de données*

## IV) Organisation

### 1) Planning prévisionnel.

En concordance avec les méthodes agiles nous mettrons en place deux Sprints chacun avec un Cooldown de 3 jours. Chacun de ces deux Sprints aura pour objectifs de fournir une version fonctionnel du système. Nous nous concentrerons donc sur les fonctionnalités vitales du système lors du premier Sprint qui commencera le Lundi 23 Mars 2015 et prendra fin le Mercredi 15 Avril. Les éléments attendu en rendu du premier Sprint seront donc, un modèle de base de donnée complet avec toutes les procédures stocker définie, la lectures des puces RFID et l'enregistrement des informations sur le tour du coureur, la pré-inscription, l'inscription, une première version de la page de KPI, ainsi qu'un mécanisme merge des bases de donnée. Ces fonctionnalités nous assurent un système fonctionnel dès la fin du premier Sprint. Le deuxième Sprint sera axé sur l'ajout de fonctionnalité dite "User friendly" comme la connexion à une compte, la modification d'un compte, l'annulation d'inscription et tout le système de KPI, notamment du système de filtre. De cette manière, nous nous assurons donc d'un système près très vite et limitons donc les risques liés à l'intégration.

**2) Répartition des tâches.**

<b>Tâches</b>	<b>Personne en charge</b>	<b>Temps estimé</b>
<b>WebServices</b>	<b>Dorian</b>	<b>4</b>
- mise en place de l'environnement	/	1
- Service de statistiques + service de pré-remplissage + tests unitaires	/	2
- gestion des risques	/	1
<b>RFID &amp; KPI</b>	<b>Djoé</b>	<b>4</b>
- RFID	/	1
- KPI	/	2
- Croisement des données	/	1
<b>Site web</b>	<b>Benjamin</b>	<b>4</b>
- Page inscription	/	1
- Pages connexion, profil, classement	/	2
- Design, finition	/	1
<b>BDD &amp; procédures stockées</b>	<b>Antoine</b>	<b>4</b>
- Création + gestion de la BDD	/	1
- procédures stockées	/	1,5
- intégration avec les différents modules	/	1,5

## V) Résultats

### 1) Fonctionnalités

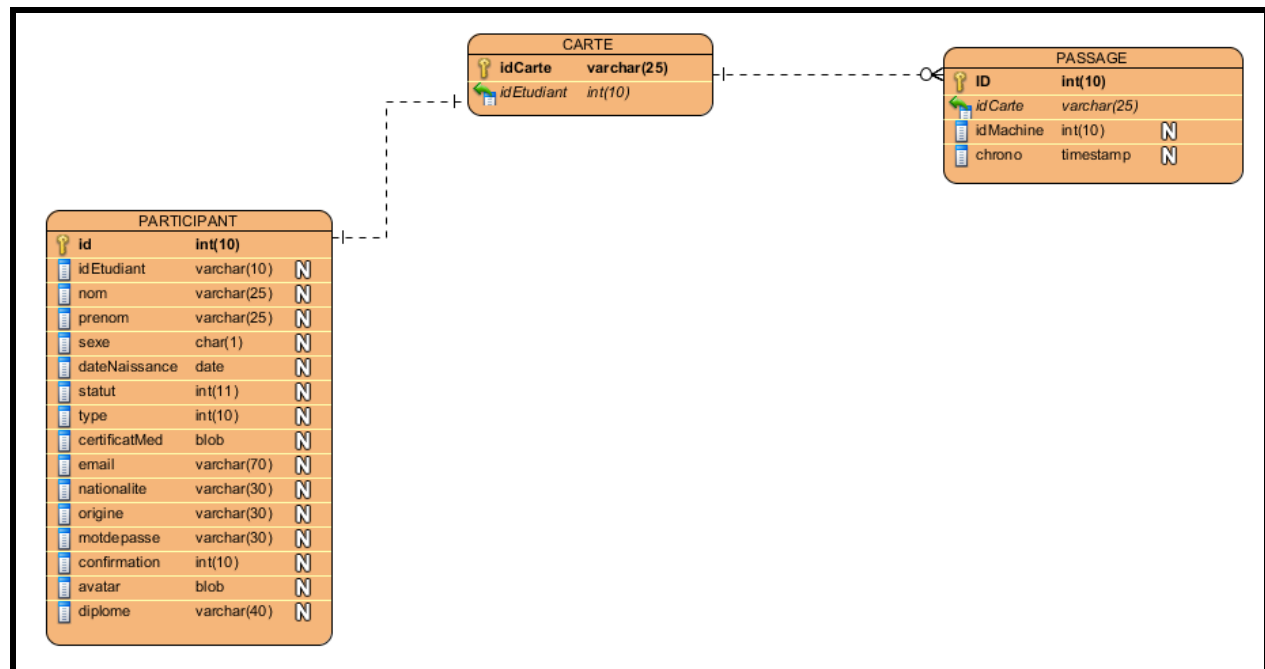
Complet

En cours

Non fait

	Fonctionnalité 1	Fonctionnalité 2	Fonctionnalité 3
BLAIN Antoine	Architecture de la BdD	Création des procedure stockées	-
DENNE Djoé	Gestion des KPI	Tag RFID	Croisement de données
LE MENACH Benjamin	Inscriptions sur le site	Connexion au site	Consultation des données utilisateur
LIZARRALDE Dorian	Web Services (LDAP)	Web Services (JDBC)	-

## 2) Module BdD



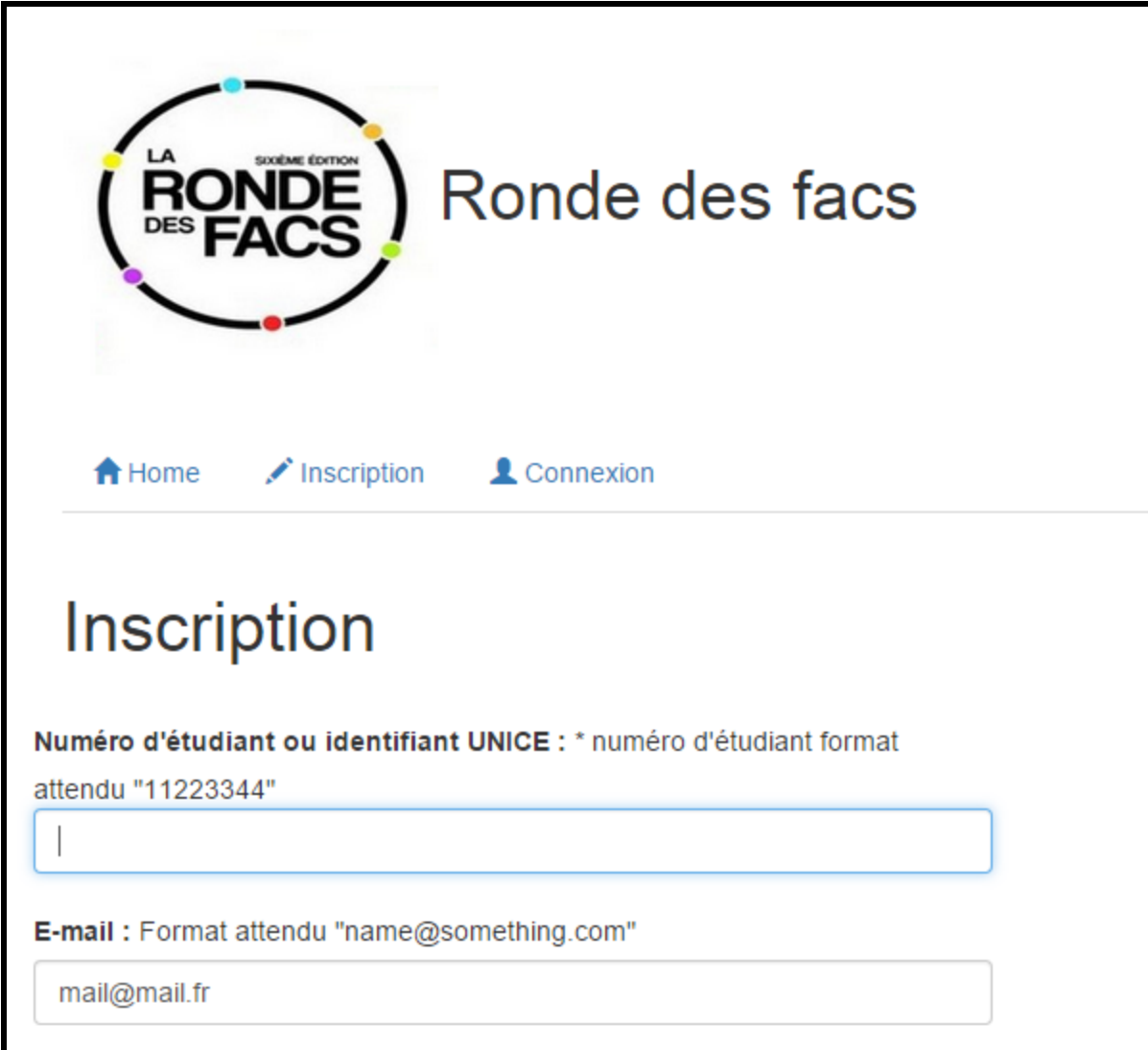
*Architecture finale de la BdD*

L'architecture de la BdD a légèrement évolué. En effet, celle-ci s'est vu dotée d'une table intermédiaire faisant le lien entre la table des participants et la table des passages via l'identifiant de la carte.

3 procédures stockées ont vu le jour :

- **topRunner** : Permet d'obtenir le classement (top 10) des participants en fonction d'un index donné (e.g. les 10 premiers du classement). Il était prévu d'obtenir aussi le classement en fonction de l'identifiant d'un participant mais la complexité de la recherche a bloqué son implémentation.
- **nbCoureurDepartement** : Permet d'obtenir le nombre de coureurs pour un département donné en fonction du nombre de tour effectué (e.g. 10 coureurs de l'IUT ont fini à 8 tours).
- **nbCoureurSexe** : Permet d'obtenir le nombre de coureurs pour un sexe donné en fonction du nombre de tour effectué (e.g. 10 femmes ont fini à 2 tours)

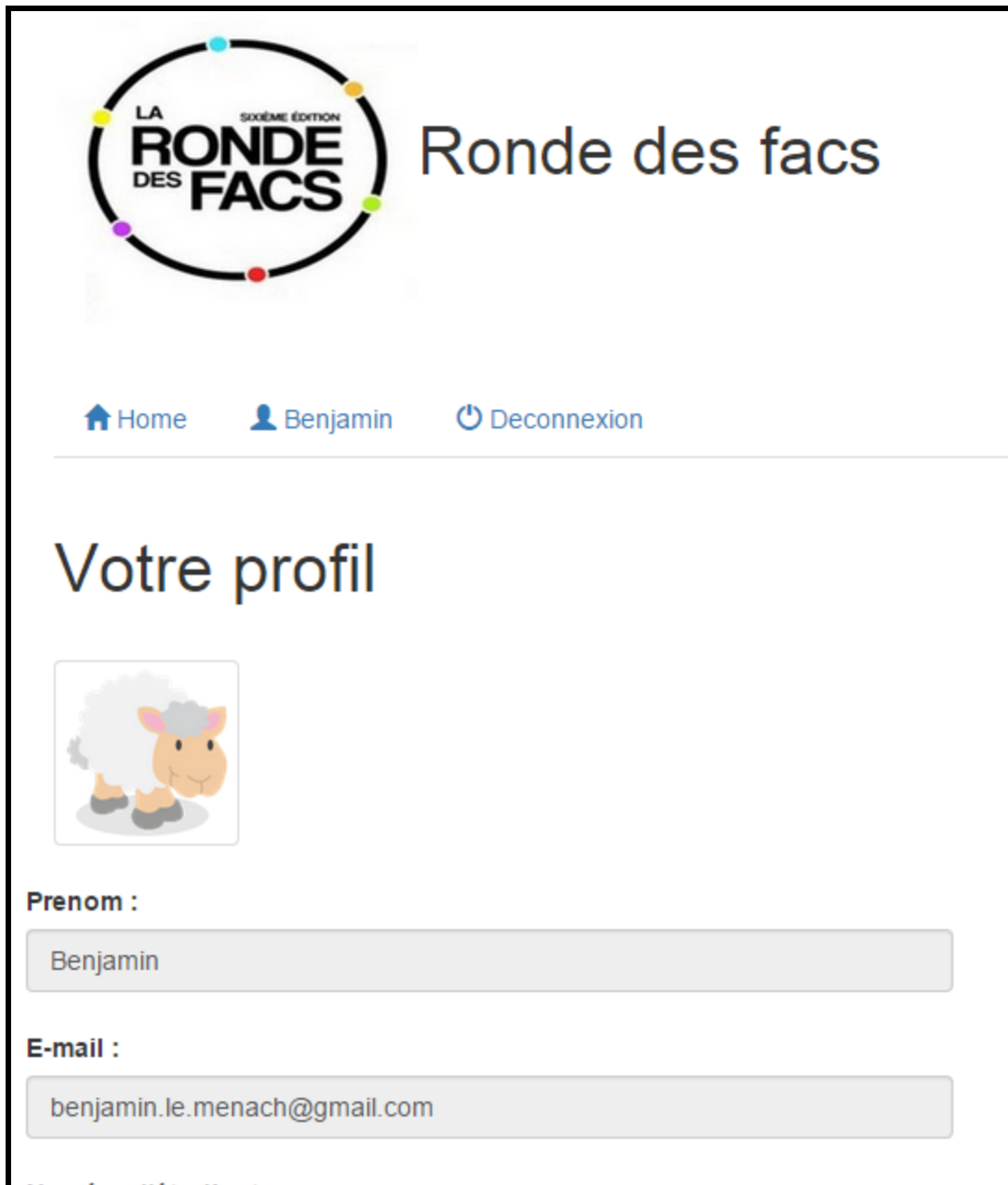
### 3) Module Site



The screenshot shows the 'Ronde des facs' website interface. At the top left is a circular logo with the text 'LA RONDE DES FACs' and 'SIXIÈME ÉDITION' around it. To the right of the logo is the title 'Ronde des facs'. Below the logo and title is a navigation bar with three links: 'Home' (with a house icon), 'Inscription' (with a pencil icon), and 'Connexion' (with a person icon). The main heading is 'Inscription'. Below this, there are two form fields. The first is for the 'Numéro d'étudiant ou identifiant UNICE', with a hint '\* numéro d'étudiant format attendu "11223344"'. The second is for the 'E-mail', with a hint 'Format attendu "name@something.com"'. The email field contains the text 'mail@mail.fr'.

*Page d'inscription au site*

Le site permet l'inscription d'un participant pouvant fournir en cas de besoin un scan de son certificat médical. La fonctionnalité LDAP n'est malheureusement pas fonctionnelle faute d'une restriction CROS (les domaines ne sont pas compatibles).



*Page de profil utilisateur*

Le profil aurait permis de consulter les informations personnelles de l'utilisateur. Par faute de temps, il ne permet pas de consulter ses statistiques (KPI). Le reste du profil reste cependant disponible.

#### 4) Module Web Services



*Jersey (API Web Services)*

Les web services sont disponibles dans leur globalité. Ils permettent l'obtention d'objets JSON liés au LDAP et aux procédures stockés. Dans un but de confidentialité, les web services ne sont disponible qu'en POST. Malheureusement l'utilisation de Jersey sur un serveur Tomcat à entraînée des erreurs CROS (le domaine du serveur Tomcat n'est pas compatible avec le domain du serveur Wamp). Cela bloque les retours d'objets JSON.

L'utilisation de jersey a permis de développer rapidement les web services. L'utilisation d'annotation jersey permettent en très peu de ligne de bloquer les accès et d'identifier les retours/prise de paramètres.

Le lien avec Tomcat a permis l'utilisation de l'API log4j utilisé pour logger le comportement des web services. Le niveau de trace est ainsi paramétrable dans le fichier log4j2.xml.

De même, il est possible de paramétrer les paramètres de connexion au LDAP et à la base de données à l'aide du fichier web.xml de l'application.

La mise en place se fait à l'aide du déploiement d'un war compilé via Maven sur un Tomcat compatible Java 7.



## **5) Plannings**

Le planning prévisionnel a largement été modifié avec le retard de l'utilisation des balises RFID. Les deux sprints n'en ont fait plus qu'un mais cela a retardé l'intégration des modules et donc la visibilité du problème CROS. L'intégration a échoué et nous ne disposons donc que de modules isolés.

## **6) Difficultés rencontrées**

Comme indiqué précédemment, une erreur de domaine a bloqué la réception des réponses JSON. Nous n'avons pas réussi à héberger la totalité du projet sur le Tomcat (l'utilisation de PHP n'était pas interpréter). Il faudrait pousser la résolution de ce problème au lieu d'essayer de le contourner. De plus, le langage MySql n'a pas aidé le développement des procédures stockées. La non représentation de ROWNUM (Oracle) nous a sérieusement déstabilisé sachant que les principales fonctions de ces procédures sont de placer les participants sur une échelle de réussite.

Enfin, les balises RFID n'ont pas été clémentes avec nous, retardant leur utilisation.

## **7) Experience**

Ce projet nous a permis de remettre en question la manière dont nous évaluons la durée des tâches. Nous tâcherons d'être beaucoup plus attentif sur cette partie à l'avenir.

Nous avons aussi réalisé l'importance de l'intégration sur un projet d'une telle ampleur. Une intégration plus rapide nous aurait certainement permis de surmonter le problème CROS.

Nous avons cependant obtenu beaucoup d'expérience en ce qui concerne les relations Bdd et l'utilisation d'API tel que Jersey/Log4j..

## **8) Reste à faire**

S'il nous était possible d'obtenir plus de temps sur le projet, il resterait donc à intégrer les différents modules entre eux et de surmonter le problème CROS. De plus, la liaison des Bdd des machines lors de la course reste à faire.